

Quest®



Toad® Data Modeler 8.0

# **User Guide**

# Copyright

## **Copyright 2025 Quest Software Inc. ALL RIGHTS RESERVED.**

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.  
Attn: LEGAL Dept  
4 Polaris Way  
Aliso Viejo, CA 92656

Refer to our Web site ([www.quest.com](http://www.quest.com)) for regional and international office information.

### **Patents**

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at [www.quest.com/legal](http://www.quest.com/legal).

### **Trademarks**

Quest, and the Quest logo are trademarks and registered trademarks of Quest Software Inc. in the U.S.A. and other countries. For a complete list of Quest Software trademarks, please visit our website at [www.quest.com/legal](http://www.quest.com/legal). All other trademarks, servicemarks, registered trademarks, and registered servicemarks are the property of their respective owners.

# Content

<b>Copyright</b> .....	<b>3</b>
<b>About Toad Data Modeler</b> .....	<b>22</b>
With Toad Data Modeler you can: .....	22
Benefits .....	22
Key Features .....	23
New in Toad Data Modeler 8.0 .....	24
Database Support .....	24
Operating Systems Support .....	24
Product Improvement Program .....	24
Visit Toad World Community .....	24
Submit Your Ideas and Suggestions .....	25
Supported Databases .....	25
Details of Database Support .....	26
Types of Connections by Databases .....	27
Toad Data Modeler Freeware .....	29
Freeware Version Limitations .....	29
Functionality Restrictions .....	29
About Us .....	30
Technical Support Resources .....	30
<b>User Interface</b> .....	<b>31</b>
Toad Data Modeler Application Layout .....	31
Simple and Minimalist Layout .....	31
Expanded Area .....	32
Application View .....	32
Application View Right-Click Options .....	33
Application Window .....	35
Designer and Workspace .....	36
Navigation on Workspace .....	38
Workspace Right-Click Options .....	38
Docking .....	39
Hot Keys .....	42
Inplace Editor .....	44
Message Explorer .....	45
Message Explorer Right-Click Options .....	46
Model Explorer .....	47
Object Navigator Dropdown Menu .....	49
Object Types and Properties - OTPs .....	50
Default (System) Selected OTPs versus User Selected OTPs .....	51

Define and Save User OTPs .....	51
Object Viewer .....	54
Status of Items in Grids .....	56
Menus .....	57
File .....	58
Edit .....	61
View .....	63
Objects .....	65
Layout .....	68
Model Menu .....	69
Tools Menu .....	72
Macros Menu .....	72
Expert Mode Menu .....	73
Settings Menu .....	75
Window Menu .....	75
Help Menu .....	76
Search Bar .....	77
Toolbars .....	77
Main Toolbar .....	78
Window Toolbar .....	80
Views Toolbar .....	80
Model Toolbar .....	81
Display Toolbar .....	81
Users Toolbar .....	82
Naming Conventions Toolbar .....	82
Grid Toolbar .....	82
Undo/Redo Toolbar .....	83
Model Objects Toolbar .....	83
Graphics Objects Toolbar .....	84
Zoom Toolbar .....	85
Colors and Alignment Toolbar .....	85
Layout Toolbar .....	85
Expert Mode Toolbar .....	86
Scripting Window .....	86
Alignment Toolbar .....	87
Help Search .....	88
Intelligence Central Toolbar .....	89
Style .....	89
<b>Models and Model Objects .....</b>	<b>90</b>
<b>Model Properties .....</b>	<b>91</b>
<b>Model Statistics .....</b>	<b>93</b>

About Physical Data Modeling .....	93
Benefits of Physical Data Model .....	93
Buttons: .....	98
Buttons: .....	105
Foreign Keys in the Attribute Properties Dialog .....	125
Foreign Keys in the Relationship Properties Dialog .....	125
Automatic FK Mapping .....	127
Domains .....	145
User Data Types .....	145
Dictionary Types .....	145
Cardinality .....	147
Cardinality .....	149
Identifying Relationship .....	150
Non-identifying Relationship .....	150
Non-identifying Self-relationship .....	151
M:N Relationship .....	151
Buttons: .....	156
Multiple Selection/Move of Relationships .....	163
Compare Procedures in Sync & Convert Wizard .....	173
Default box versus Default Rule combo-box .....	177
Method A - via the Users dialog .....	180
Method B - via the User Groups dialog .....	180
Select Target Database .....	181
Create Entities .....	182
Create Attributes .....	183
Create Keys .....	186
Create Relationships .....	188
Notation and Cardinality .....	190
Optional/Mandatory Parent/Child .....	195
Set up Referential Integrity Rules .....	196
Connect Parent and Child Entities .....	198
Understanding Foreign Keys .....	199
Parent Attributes (Rolenames) .....	202
About Advanced Options .....	204
Alternate Keys .....	204
Indexes .....	205
Check Constraints .....	207
Triggers .....	208
Views .....	210
Materialized Views .....	213
Procedures .....	214
Compare Procedures in Sync & Convert Wizard .....	217
Functions .....	217
Schemas .....	219
Users .....	219
User Groups .....	221

Permissions .....	222
About Universal Data Model .....	224
Specifics of Universal (Generic Relational) Data Model .....	224
About Logical Data Modeling .....	224
Benefits of Logical Data Model .....	225
Specifics of Logical Data Model .....	225
Benefits of Super and Sub Types .....	225
Model A - Utilizing Super and Sub Types .....	226
Model B - Lacking Super and Sub Types .....	227
Disadvantage .....	227
Objects in Logical Model .....	228
Format Logical Objects .....	228
Edit Entities .....	229
Entity Right-Click Options in Logical Model .....	231
Create Attributes .....	232
Buttons: .....	232
Edit Attributes .....	233
Create Unique Identifier .....	235
Edit Unique Identifiers .....	235
Select Linking Method .....	236
Edit Relationships .....	236
Create Inheritances .....	240
Edit Inheritances .....	240
Convert Logical Model to Physical Model .....	242
Data Type Conversion .....	247
M:N Relationships .....	248
Migration of Keys .....	249
Inheritance .....	254
Defaults .....	257
Rules .....	258
Add Objects .....	258
Edit Objects .....	259
Format Objects .....	260
Select Objects .....	262
Align Objects .....	264
Rename Objects .....	266
Copy Objects .....	266
Move Objects .....	267
Delete Objects .....	269
Find Objects .....	271
External Objects .....	272
About Shortcuts of Objects .....	274
Create Shortcuts .....	275
Edit Shortcuts .....	276
Remove and List Shortcuts .....	276
2-D Shapes .....	278

Note and Line .....	278
Image .....	279
Stamp .....	280
Caption of Categories .....	281
Application Variables .....	281
Accessing Application Variables .....	282
Syntax of Application Variables .....	283
Application Variables - Examples .....	284
Export/Import - Microsoft Excel .....	287
Export/Import - CSV .....	290
Export to Graphic File .....	291
Import from ER/Studio Data Architect 11 .....	292
DDL Script Generation Preparation .....	293
How to Generate DDL Script .....	296
Example of Generated Script .....	303
Autolayout .....	308
Arrange Objects in Layers .....	309
Arrange Relationship Lines .....	310
Move Lines, Anchor Points and Handle Points .....	311
Column to Column Alignment .....	312
Add Handle Points .....	312
Add Horizontal or Vertical Lines Only .....	313
Unhide Line .....	314
Move, Hide, Find Relationship Names .....	314
Format Relationship Lines .....	315
Categories .....	315
Change Notation .....	317
Data Warehouse Types .....	318
Display Modes .....	319
Display Level of Entities .....	322
Format Workspaces and Objects .....	323
Line Autolayout .....	323
Select Colors for Attributes and Keys .....	326
Show Grid and Grid Size .....	328
Shift and Zoom in Your Model .....	328
Navigation Tips on Workspace .....	328
Model Actions .....	329
Change Script .....	333
Change Script Generation - Temporary Tables .....	343
Simple Change Script .....	343
Merge Models .....	344
Simple Model Merge .....	351
Data Type Conversion Settings .....	352
PER - PER Conversion Information .....	354
PER - LER Conversion Information .....	357

Convert Model .....	359
Physical to Physical .....	360
Physical to Logical .....	364
Logical to Physical .....	367
Simple Model Conversion .....	372
Compare Models, Generate Change Report .....	372
Model Update .....	390
Synchronization .....	400
Database and Model Synchronization .....	400
Physical Model and Logical Model Synchronization .....	400
Limitations .....	401
Recommendations .....	401
Model Size Limitations .....	401
Print .....	402
Tips before You Print .....	402
Page Setup .....	402
Preview .....	403
PDF Printing .....	404
Create New Project .....	407
Add Existing Models to Project .....	409
Project Reports .....	413
XSL Transformation .....	414
Customize XSLT Templates .....	415
Predefined XSL Templates .....	417
Sample XSLT File .....	418
Edit Existing XSLT Templates .....	419
HTML Reports .....	419
RTF Reports .....	426
PDF Reports .....	431
Connections .....	436
Reverse Engineering Wizard .....	442
Object Explorer .....	444
Import Toad ERD .....	446
Universal DB/ANSI Model .....	446
About Scripting and Customization .....	447
Scripts/Macros .....	448
Packages .....	448
Metamodel .....	449
Form Customization .....	450
Other Notes .....	451
Scripting Window .....	451
Script Explorer .....	452
The Script Explorer Tree .....	453
Options for Folders .....	454
Options for Scripts .....	454

Script Properties .....	455
Script Editor .....	456
Script Editor Right-Click Options .....	457
How to Call Toad Data Modeler from Other Applications .....	457
Package Explorer .....	464
The Package Explorer Tree .....	465
Right-click Package options .....	466
Package Explorer Toolbar .....	466
Icons of Packages in Package Explorer .....	467
Scripting in Script Editor .....	468
Internal Scripting .....	469
Key Words .....	469
Creating Custom Properties .....	471
Custom Property Example 1 .....	474
Sample Scripts and Scripting Tips .....	483
Create Package .....	491
Add New Properties in Metamodel .....	493
Creating New Objects .....	496
Modify Form .....	498
Set Default Values .....	501
Add Events .....	502
Dialogs .....	507
Access Property Values via Scripting Window .....	509
Code: .....	510
File System Scripts .....	514
Create Script .....	515
Code .....	516
Getting Settings Information .....	517
Iterate Entity And Attributes .....	517
Call Existing Script from Model Properties Form .....	519
Code .....	519
Modify HTML Reports .....	519
Code: .....	524
Editable Forms and Frames .....	526
Editable Forms Right-Click Options .....	529
Component Inspector .....	529
Component Palette .....	530
Form Explorer .....	531
Macros .....	532
Productivity Pack .....	539
Rename Objects Pack .....	547
Macros and User Forms .....	550
EVENTS .....	552
CONTROL .....	552
Button .....	553
Checkbox .....	553
Combobox .....	553

Edit .....	553
Memo .....	553
Radio Button .....	553
Macros and User Forms - Use Case .....	553
About Metamodel in Toad Data Modeler .....	558
Open Metamodels .....	559
About Templates .....	567
Template Editor .....	568
Manage Templates .....	571
Toad for Oracle Templates .....	573
Toad for Oracle Auto Replacement Words .....	574
Rename .....	574
Tips and Hints for Large Models .....	575
Command Line Parameters .....	580
Other Quick Tips .....	581
Objects on the Workspace and Keyboard Arrows .....	581
Navigation on Workspace .....	582
Make a Copy of Multiple Objects on the Workspace (CTRL+A, CTRL+C, CTRL+V) .....	582
Print Models .....	582
HTML Report Layout .....	582
About Integration Options .....	582
Toad for Oracle - Basic Information on Product .....	583
Default Settings for Toad for Oracle Integration .....	583
Toad for Oracle® Connections .....	584
Import Toad for Oracle® ER Diagrams .....	585
Open Toad for Oracle® Projects .....	587
Toad for Oracle® as Default Editor .....	587
Toad® for Oracle® Icons .....	588
Basic Actions .....	589
Publishing Models/Reports .....	594
Notifications .....	595
About Naming Conventions .....	596
Verification and Synchronization .....	598
Valid Characters .....	603
Character Replacement .....	604
Word Replacement .....	605
Valid Characters + Replacement .....	605
To Do List .....	609
Quick Search .....	611
Standard Search .....	611
Wildcards .....	612
Word Recognition .....	612
Camel Case .....	613
Dot Notation .....	614
Gallery .....	614

Modifying Items .....	618
Gallery Explorer .....	619
Model Verification .....	621
Model Verification Form .....	621
Verification Log .....	622
On Form Verification .....	623
Syntax Validity .....	625
Settings .....	625
Data Generation for SQL Server .....	626
Refactoring Utility .....	627
Renamed Objects .....	628
Objects to Modify .....	628
Current and Modified Code Previews .....	629
Schema/Owner Assignment .....	629
Infer Relationships .....	630
Infer Relationships and Reverse Engineering .....	633
Git Version Control .....	634
Git application view .....	635
Getting Started - Subversion .....	635
Application Settings - Version Control System .....	641
Project Settings .....	644
Subversion Actions .....	648
About Version Manager .....	651
What Is a Project? .....	652
Projects and Files in Version Manager .....	652
Version Manager Toolbar and Options .....	654
List of Projects, Files, Versions .....	657
<b>Options .....</b>	<b>659</b>
Application .....	660
Model .....	667
Import and Export of Settings .....	672
Default Values .....	674
Default Values Dialog .....	676
Dictionaries .....	677
Localized HTML, RTF and PDF Reports .....	678
File Extensions .....	680
*.TXP Files .....	681
*.TXL Files .....	681
*.TXM Files .....	681
*.TXG and * TBG Files .....	681
*.TXD Files .....	681
*.TXC Files .....	681
*.TXE File .....	682
*.TXS Files .....	682
*.XSLT Files .....	682

*.XSD Files .....	682
*.TXA Files .....	682
*.TXV Files .....	682
*.TXN Files .....	682
*.TXI Files .....	683
*.TXO Files .....	683
*.CSV Files .....	683
*.TXN Files .....	683
Enabled/Disabled Databases .....	684
<b>Supported Databases .....</b>	<b>687</b>
Details of Database Support .....	688
Specifics - Amazon Redshift 1.0 .....	689
Reverse Engineering - Amazon Redshift 1.0 .....	689
Script Generation - Amazon Redshift 1.0 .....	690
Specifics - DB2 9.5 (LUW) .....	692
Attribute .....	693
Index .....	695
Dictionary Type .....	696
Extra Objects - DB2 9.5 (LUW) .....	696
Reverse Engineering - IBM DB2 LUW .....	697
Specifics - DB2 9.7 (LUW) .....	700
Dictionary Type .....	701
User Data Type .....	702
Stored Procedure .....	703
Synonym .....	704
Reverse Engineering - IBM DB2 LUW .....	705
Specifics - DB2 10.1 (LUW) .....	708
Entity .....	709
Attribute .....	710
Function .....	711
Index .....	712
Trigger .....	712
Reverse Engineering - IBM DB2 LUW .....	713
Script Generation - DB2 v.10.1 (LUW) .....	716
Specifics - DB2 10.5 (LUW) .....	717
Entity .....	718
Index .....	722
Type .....	724
Reverse Engineering - IBM DB2 LUW .....	725
Script Generation - DB2 v.10.5 (LUW) .....	728
Specifics - DB2 11.1 (LUW) .....	729
Functions and Procedures .....	729

Datatypes .....	729
Reverse Engineering - IBM DB2 LUW .....	729
Script Generation - DB2 v.11.1 (LUW) .....	732
Specifics - DB2 z/OS v. 11 .....	733
Index .....	733
Exclude Null Keys .....	733
Business Time Period .....	733
Dictionary Type .....	734
Array Type .....	734
Inline Length parameter .....	735
Attribute .....	736
Key .....	737
Trigger .....	738
Function .....	739
Reverse Engineering - DB2 z/OS v. 11 .....	740
Script Generation - DB2 z/OS v. 11 .....	742
Specifics - Greenplum 4.1 .....	743
Entity .....	743
General tab .....	743
Storage Parameters tab .....	744
Table Partitions tab .....	744
Attribute .....	745
Dictionary Type/Domain .....	746
Function .....	747
Foreign Key (Referential Integrity) .....	748
Trigger .....	748
Aggregate .....	749
Sequence .....	750
Rewrite Rules .....	751
External Tables .....	751
Reverse Engineering - Greenplum 4.1 .....	753
Specifics - Greenplum 4.2 .....	755
Entity .....	755
Attribute .....	756
External Table .....	757
User Data Type .....	757
Function .....	758
Aggregate Function .....	759
Reverse Engineering - Greenplum 4.2 .....	759
Script Generation - Greenplum 4.2 .....	760
Specifics - Ingres 9.3 .....	761
Entity .....	761
Index .....	762
Synonyms .....	763

Reverse Engineering - Ingres 9.3 .....	764
Script Generation - Ingres 9.3 .....	766
Specifics - Ingres 10.0 .....	767
Reverse Engineering - Ingres 10.0 .....	767
Specifics - EDB Postgres Advanced Server 10 .....	768
Script Generation - EDB Advanced Server 10 .....	769
Reverse Engineering - EDB Advanced Server .....	769
Specifics - Microsoft Access 2010 .....	772
Entity .....	772
Attribute .....	773
User Data Types in the Model menu .....	774
Specifics - Microsoft Access 2010 .....	775
Entity .....	775
Attribute .....	776
Reverse Engineering - Microsoft Access 2007/2010 .....	777
Script Generation - Microsoft Access 2007/2010 .....	777
Specifics - Microsoft Azure SQL Database V12 .....	779
Reverse Engineering - Microsoft Azure SQL Database V12 .....	780
Script Generation - Microsoft Azure SQL Database V12 .....	783
Specifics - Microsoft SQL Server 2012 .....	784
Entity .....	784
Fulltext Index tab .....	784
Search Property List .....	785
Data Compression Tab .....	787
Description Column .....	788
Index .....	789
Columnstore option .....	789
Spatial Index Parameters Tab .....	790
The FilestreamFilegroup box .....	791
Data Compression Tab .....	792
Secondary XML Index Parameters Tab .....	793
Spatial Index Parameters Tab .....	794
Key .....	795
Data Compression Tab .....	795
Index Options Tab .....	796
Data Space .....	797
View .....	798
User Data Type .....	799
Function .....	800
Extended Properties .....	801
Attribute .....	802
Data Types .....	802
Trigger .....	804
CLR Trigger .....	804
Removal of the With Append option .....	805

Index Options Tab .....	806
Dictionary Type .....	807
User Data Type .....	808
Users .....	809
Domain .....	810
Schema .....	811
View .....	812
Procedure .....	814
Functions .....	815
Defaults .....	816
Check Constraint Rules .....	816
Synonym .....	816
Extended Properties .....	818
File Table .....	819
Sequence .....	821
Extended Properties .....	822
Reverse Engineering - Microsoft SQL Server 2012 .....	822
Script Generation - Microsoft SQL Server 2012 .....	823
Specifics - Microsoft SQL Server 2014 .....	824
Entity .....	824
Index .....	825
Key .....	828
Procedure .....	830
User Data Type .....	831
Reverse Engineering - Microsoft SQL Server 2014 .....	831
Script Generation - Microsoft SQL Server 2014 .....	832
Specifics - Microsoft SQL Server 2016 .....	833
Entity .....	833
Attribute .....	834
Function .....	835
Index .....	836
User .....	836
Misc. ....	836
Reverse Engineering - Microsoft SQL Server 2016 .....	837
Script Generation - Microsoft SQL Server 2016 .....	838
Specifics - Microsoft SQL Server 2017 .....	839
Entity .....	839
Details .....	840
Reverse Engineering - Microsoft SQL Server 2017 .....	840
Script Generation - Microsoft SQL Server .....	841
Specifics - Microsoft SQL Server 2019 .....	842
Indexes .....	842
Functions .....	842

Reverse Engineering - Microsoft SQL Server 2019 .....	842
Script Generation - Microsoft SQL Server .....	843
Specifics - Microsoft SQL Server 2022 .....	844
Keys .....	844
Reverse Engineering - Microsoft SQL Server 2022 .....	845
Script Generation - Microsoft SQL Server .....	848
Specifics - MySQL 5.6 .....	849
Entity .....	849
Row Format .....	849
Partition tab .....	851
Attribute .....	853
Model Conversion from MySQL to SQL Server and Oracle .....	853
Relationship .....	854
Index .....	854
Trigger .....	855
User Data Types, Dictionary Types .....	856
Databases .....	856
Data Types .....	856
Index .....	856
Entities .....	856
Reverse Engineering - MySQL 5.6 .....	857
Script Generation - MySQL 5.6 .....	857
Specifics - MySQL 5.7 .....	858
Attributes .....	858
Data Types .....	858
Tables .....	859
Triggers .....	859
Change Script Generation .....	859
Reverse Engineering - MySQL 5.7 .....	859
Script Generation - MySQL 5.7 .....	859
Specifics - MySQL 8.0 .....	860
Reverse Engineering - MySQL 8.0 .....	860
Script Generation - MySQL 8.0 .....	862
Specifics - Oracle 11g Release 1 .....	864
Entity .....	864
External tables .....	864
Attribute .....	866
Data Types .....	866
Virtual Column .....	868
Index .....	869
Expression Index .....	869
XML Index .....	871
Trigger .....	872
Trigger (Entity) .....	874
Trigger (View) .....	875

User Data Type .....	876
Materialized View .....	877
Procedure .....	878
Function .....	880
Directory .....	882
Java .....	883
Sequence .....	884
Synonym .....	885
Tablespaces .....	886
Reverse Engineering - Oracle 11g Release 1 .....	886
Specifics - Oracle 11g Release 2 .....	887
View .....	887
Edition .....	889
Trigger (Entity) .....	890
Trigger (View) .....	891
Change Script .....	892
Script Generation - Oracle 11g Release 2 .....	894
Reverse Engineering - Oracle 11g Release 2 .....	896
Specifics - Oracle 12c Release 1 .....	897
Entity .....	897
Attribute .....	898
Index .....	900
Keys .....	901
Materialized View .....	902
Function, Procedure, Package, User Data Type, Synonym, View, Trigger .....	903
View .....	903
Sequence .....	904
Reverse Engineering - Oracle 12c Release 1 .....	904
Script Generation - Oracle 12c Release 1 .....	905
Specifics - Oracle 12c Release 2 .....	906
Reverse Engineering - Oracle 12c Release 2 .....	910
Script Generation - Oracle 12c Release 2 .....	911
Specifics - Oracle 18c .....	912
Reverse Engineering - Oracle 18c .....	912
Script Generation - Oracle 18c .....	912
Specifics - Oracle 19c .....	913
Reverse Engineering - Oracle 19c .....	913
Script Generation Oracle 19c .....	913
Specifics - Oracle 21c .....	914
Attribute Properties .....	914
JSON Data Type .....	914
Reverse Engineering - Oracle 21c .....	915
Script Generation - Oracle 21c .....	918

Specifics - PostgreSQL 9.5 .....	919
Entity .....	919
Table Type checkbox .....	919
Unlogged checkbox .....	920
Table, Attribute - Check Constraint .....	921
Foreign Table .....	922
Materialized Views .....	923
Views .....	923
Tables .....	924
Aggregates .....	924
Schemas .....	925
Foreign Server .....	927
Attribute .....	928
Collation combobox .....	928
Attributes - Foreign Table .....	929
Collation .....	930
Index .....	934
Set Collation .....	934
Index Access Method .....	935
Relationship .....	936
Key Properties .....	937
Trigger .....	938
View - Trigger .....	939
User Data Type .....	940
Tables .....	941
Indexes .....	942
Materialized Views .....	943
User Groups .....	944
Change Script Generation .....	944
Functions .....	945
Reverse Engineering - PostgreSQL 9.5 .....	945
Script Generating - PostgreSQL 9.5 .....	945
Specifics - PostgreSQL 10 .....	946
Reverse Engineering - PostgreSQL 10 .....	946
Script Generating - PostgreSQL 10 .....	946
Specifics - PostgreSQL 11 .....	946
Reverse Engineering - PostgreSQL 11 .....	946
Script Generating - PostgreSQL 11 .....	946
Specifics - PostgreSQL 12 .....	947
Reverse Engineering - PostgreSQL 12 .....	948
Script Generation - PostgreSQL 12 .....	950
Specifics - SQLite 3.7 .....	951
Entity .....	951
Attribute .....	952
Collation .....	952

Relationship .....	953
Key .....	954
Index .....	956
Trigger - Entity .....	957
Trigger - View .....	958
View .....	959
Virtual Table .....	960
Database .....	961
Reverse Engineering - SQLite 3.7 .....	961
Script Generation - SQLite 3.7 .....	963
Specifics - Sybase ASE 15.5 .....	965
Entity .....	965
Attribute .....	966
Encryption Keys .....	967
Trigger .....	968
Web Services .....	969
Reverse Engineering - Sybase ASE 15.5 .....	969
Specifics - Sybase ASE 15.7 .....	973
Entity .....	973
Attribute .....	975
Reverse Engineering - Sybase ASE 15.7 .....	976
Specifics - SAP ASE 16.0 .....	979
Entities .....	979
Triggers .....	980
Indexes .....	981
Procedures .....	982
Reverse Engineering - SAP ASE 16.0 .....	983
Script Generation - SAP ASE 16.0 .....	986
Specifics - Sybase IQ 15.2 .....	987
Attribute .....	987
Relationship .....	988
Index .....	989
Text Configuration Object .....	991
Database Spaces (DBSPACE (51001)) .....	992
User-Defined Messages (MESSAGE (53201)) .....	993
Reverse Engineering - Sybase IQ 15.2 .....	993
Specifics - Sybase SQL Anywhere 11 .....	996
Attribute .....	996
Relationship .....	997
Index .....	998
Reverse Engineering - Sybase SQL Anywhere 11 .....	999
Specifics - SAP SQL Anywhere 17 .....	1002
Reverse Engineering - SAP SQL Anywhere 17 .....	1002

Script Generation - Sybase SQL Anywhere 11 .....	1004
Specifics - Teradata 13 .....	1005
Entity .....	1005
Attribute .....	1006
Index .....	1006
Procedure .....	1009
Images .....	1010
Reverse Engineering - Teradata 13 .....	1010
Specifics - Vertica Database 8.0 .....	1012
Reverse Engineering - Vertica 8.0 .....	1012
Script Generation - Vertica Database 8.0 .....	1013
<b>Legal Notices .....</b>	<b>1014</b>
Third-party components .....	1014
Acknowledgments .....	1014
<b>Glossary .....</b>	<b>1015</b>
<b>Index .....</b>	<b>1016</b>

# About Toad Data Modeler

Toad Data Modeler helps organizations create, maintain and document their database systems with an easy-to-use graphical interface.

## With Toad Data Modeler you can:

- Create database structures visually (Logical, Universal and Physical Entity Relationship Diagrams - ERD)
- Create ERD for various target database systems (Oracle Database, SQL Server, DB2, MySQL, PostgreSQL and more, see [Supported Databases](#))
- Reverse engineer already existing database structures and see them in a form of a diagram
- Add logical data to your diagrams and describe existing database structures better
- Verify model, get a list of Errors, Warnings and Hints and use Quick Fixes to correct issues
- Automatically generate SQL code for selected target database
- Generate detailed documentation in HTML, RTF or PDF format
- Export database structures to Microsoft Excel and edit comments, notes and other model item properties
- Synchronize models with already existing databases (using Change Script Generation and Update Model features) and much more

## Benefits

Toad Data Modeler allows you to:

- Reduce errors in development
- Significantly improve your productivity
- Visualize your database structures and get better understanding of your existing databases
- Verify your work automatically
- Generate very detailed HTML, RTF or PDF reports
- Work with existing database structures
- Keep existing database structures up-to-date

# Key Features

- Physical Model, Universal Model (Generic Relational Model) and Logical Model
- Support for Various Databases
- Reverse Engineering
- Version Control System (support for Apache Subversion and Git) and Internal Version Manager
- Projects for storing models and additional documentation
- SQL/DDDL Script Generation
- HTML/RTF Reports (including Alter reports)
- Model Merge, Model Compare
- Model Verification
- Autolayout
- Refactoring Utility
- Automation
- Customization and Scripting
- Templates, Gallery, Default Values, Application Variables, Macros
- Undo/Redo
- To-Do List
- Zoom, Loupe, Model Overview features and many more...

**Toad Data Modeler delivers increased level of efficiency, quality and comfort to all database professionals.**

# New in Toad Data Modeler 8.0

## Database Support

The following databases are supported in the current release:

- Oracle 21c
- Microsoft SQL Server 2022

## Operating Systems Support

The following operating systems are supported in the current release:

- Windows 11\*
- Windows Server 2019
- Windows Server 2022

**i** | \*Note: Windows 11 24H2 patch and higher are not supported.

## Product Improvement Program

To prioritize enhancements in future releases, Toad Data Modeler collects data about the use of its different features, and periodically, this data is communicated back to us. Initially, this usage data includes an IP address. Upon its receipt at a temporary server in the U.S.A., the IP address is removed, and then the anonymous data is aggregated before it is sent to our servers in California. Our product team analyses the aggregated data to understand our user community's preferences and common practices. This analysis influences our future releases.

No personal information is collected and you can stop participating at any time.

Freeware users are required to participate in the product improvement program and cannot reject to participate. When you insert your license key the freeware version becomes full and you might modify product improvement program settings.

To view or change your participation in the product improvement program:

1. Go to **Help Menu | Product Improvement Program**
2. Select **Yes, I want to participate** or **No, thank you**

You can review the privacy policy at <https://www.quest.com/legal/privacy.aspx>.

## Visit Toad World Community

Watch instructional flash movies that will guide you through great features of Toad Data Modeler, learn about tips and tricks, read how-to's, tutorials, download sample scripts and user packages, dictionary files and templates for reports, join our discussion forum and much more at: <https://www.toadworld.com>.

# Submit Your Ideas and Suggestions

If you are already the community member, feel free to post any **ideas and suggestions for new features and improvements** for Toad Data Modeler.

## Supported Databases

Toad Data Modeler provides full support to the databases listed below:

- Amazon® Aurora MySQL 5.6
- Amazon® Aurora PostgreSQL 9.5
- Amazon® Redshift 1.0
- IBM® DB2® z/OS® 11
- IBM® DB2® LUW 9.7, 10.1, 10.5, 11.1, 11.5
- Greenplum Database® 4.2
- Ingres 9.3, 10.0
- EDB Postgres Advanced Server 10
- Microsoft® Access® 2010-2019, incl. Office 365
- Microsoft® Azure® SQL Database V12
- Microsoft® SQL Server® 2012, 2014, 2016, 2017, 2019, 2022
- MySQL 5.6, 5.7, 8.0
- Oracle® 11g R1, 11g R2, 12c R1, 12c R2, 18c, 19c, 21c
- PostgreSQL 9.5, 10, 11, 12
- SQLite 3.7, 3.32
- SAP® SQL Anywhere 17
- SAP® ASE 16.0
- Sybase® ASE 15.7
- Sybase® IQ 15.2
- Teradata 13, 16.2
- Vertica Database 8.0
- Other (Universal Model)

**i** **IMPORTANT:** You can also create and work with models from other versions of databases that have been deprecated by their production companies. Toad Data Modeler cannot provide fixes and provide support for more features of these databases. Uncheck **Show Supported Databases Only** to display all databases that you can create and open models for.

# Details of Database Support

Supported Database System	Reverse Engineering		Change Script Generation	SQL/DDDL Code Generation
	From a Database	From a SQL File		
Amazon Aurora MySQL	●	●	●	●
Amazon Aurora PostgreSQL	●	●	●	●
Amazon Redshift	●	●	●	●
IBM DB2 z/OS	●	●	●	●
IBM DB2 LUW	●	●	●	●
Greenplum	●	●	●	●
Ingres	●	●	●	●
EDB Postgres Advanced Server	●	●	●	●
Microsoft Access	●	●	●	●
Microsoft Azure SQL Database	●	●	●	●
Microsoft SQL Server	●	●	●	●
MySQL	●	●	●	●
Oracle	●	●	●	●
PostgreSQL 9.5 and newer	●	●	●	●
SQLite	●	●	●	●
SAP ASE	●	●	●	●
Sybase ASE	●	●	●	●
Sybase IQ	●	●	●	●
SAP SQL Anywhere	●	●	●	●
Teradata	●	●	●	●
Vertica Database	●	●	●	●
Other databases (Universal Model)	●	●	●	●

**i** Note: Toad Data Modeler includes also support of Universal DB/ANSI Models. See [Universal DB/ANSI Model](#) for more information.

# Types of Connections by Databases

**Native** Native connection via client. The provider does not require installation of any additional software on the client. Some databases do not require installation of a client because it is embedded in Toad Data Modeler (Amazon Aurora PostgreSQL, Greenplum, EDB Postgres, Greenplum, SQLite and PostgreSQL).

**ODBC** Connection via ODBC Database Connection via ODBC (Open Database Connectivity) Driver. ODBC Driver is not part of Toad Data Modeler. In most cases, it is distributed directly with database.

**ADO** Connection via ADO Database Connection via ADO (ActiveX Data Objects) with OLE DB Drivers. OLE DB Drivers are not part of Toad Data Modeler. In most cases, they are distributed directly with database.

**TCP/IP** Connection via TCP/IP This connection type allows you to connect to your database without client. To make the connection successfully, you need to know the TCP/IP server name and port.

Supported Database System	Native	ODBC	ADO	Other
Amazon Aurora MySQL 5.6		●		TCP/IP
Amazon Aurora PostgreSQL 9.5	●	●		
Amazon Redshift 1.0		●		
DB2 z/OS v. 11	●	●		
DB2 v. 11.1 (LUW)	●	●		
DB2 v. 10.5 (LUW)	●	●	●	
DB2 v. 10.1 (LUW)	●	●	●	
DB2 v. 9.7 (LUW)	●	●	●	
DB2 v. 9.5 (LUW)	●	●	●	
Greenplum 4.2	●			
Greenplum 4.1	●			
Ingres 10.0		●		
Ingres 9.3		●		
EDB Postgres Advanced Server 10	●	●		
Microsoft Access 2010				ADO and DAO
Microsoft Azure SQL Database V12	●		●	
Microsoft SQL Server 2022	●		●	
Microsoft SQL Server 2019	●		●	
Microsoft SQL Server 2017	●		●	
Microsoft SQL Server 2016	●		●	
Microsoft SQL Server 2014	●		●	

Supported Database System	Native	ODBC	ADO	Other
Microsoft SQL Server 2012	●		●	
MySQL 8.0		●		TCP/IP
MySQL 5.7		●		TCP/IP
MySQL 5.6		●		TCP/IP
Oracle 21c	●		●	TCP/IP
Oracle 19c	●		●	TCP/IP
Oracle 18c	●		●	TCP/IP
Oracle 12c R2	●		●	TCP/IP
Oracle 12c R1	●		●	TCP/IP
Oracle 11g R2	●		●	TCP/IP
Oracle 11g R1	●		●	TCP/IP
PostgreSQL 12	●	●		
PostgreSQL 11	●	●		
PostgreSQL 10	●	●		
PostgreSQL 9.5	●	●		
SQLite 3.7	●			
SAP ASE 16.0	●	●	●	
Sybase ASE 15.7	●	●	●	
Sybase ASE 15.5	●	●	●	
Sybase IQ 15.2		●	●	
SAP SQL Anywhere 17		●	●	
Sybase SQL Anywhere 11		●	●	
Teradata 13				
Vertica Database 8.0		●		
Other databases (Universal Model)				Universal ADO

# Toad Data Modeler Freeware

**i** **Important:** Toad Data Modeler Freeware:

- expires on February 11, 2026 or 380 days after first run.
- has some limitations. For more information see the list of limitations in **Installation Guide**.

## Freeware Version Limitations

Freeware version of Toad® Data Modeler does not require a license.

## Functionality Restrictions

Freeware version of Toad Data Modeler has the following restrictions:

- **Save Model** – You can save a model with max 25 entities.
- **Preview** – You can preview a model with max 25 entities.
- **Print** – You can print a model with max 25 entities.
- **Export to Graphic File** – You can export a model with max 25 entities to graphic file.
- **Alter Report** – You can generate alter report for a model with max 25 entities.
- **Report** – PDF reports with watermark are limited to a maximum of 25 entities. Number of reported other objects is not limited. HTML and RTF reports cannot be generated.
- **XSL Transformation Report** – You can generate the XSL Transformation report only for first 25 entities of the model. Other objects (procedures, functions etc.) don't have any limitation.
- **SQL/DDL Script Generation** – You can generate SQL/DDL script only for first 25 entities of the model. Other objects (procedures, functions etc.) don't have any limitation.
- **Change Script Generation** – This feature is not available at all.
- **Reverse Engineering** – Reversing database schema is limited to 25 entities. Other objects have no limitation.
- **Undo** – Only one step back is available in Freeware version.

# About Us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit [www.quest.com](http://www.quest.com).

## Technical Support Resources

For sales or other inquiries, visit [www.quest.com/company/contact-us.aspx](http://www.quest.com/company/contact-us.aspx) or call +1 949 754-8000.

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at <https://support.quest.com>.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product

# User Interface

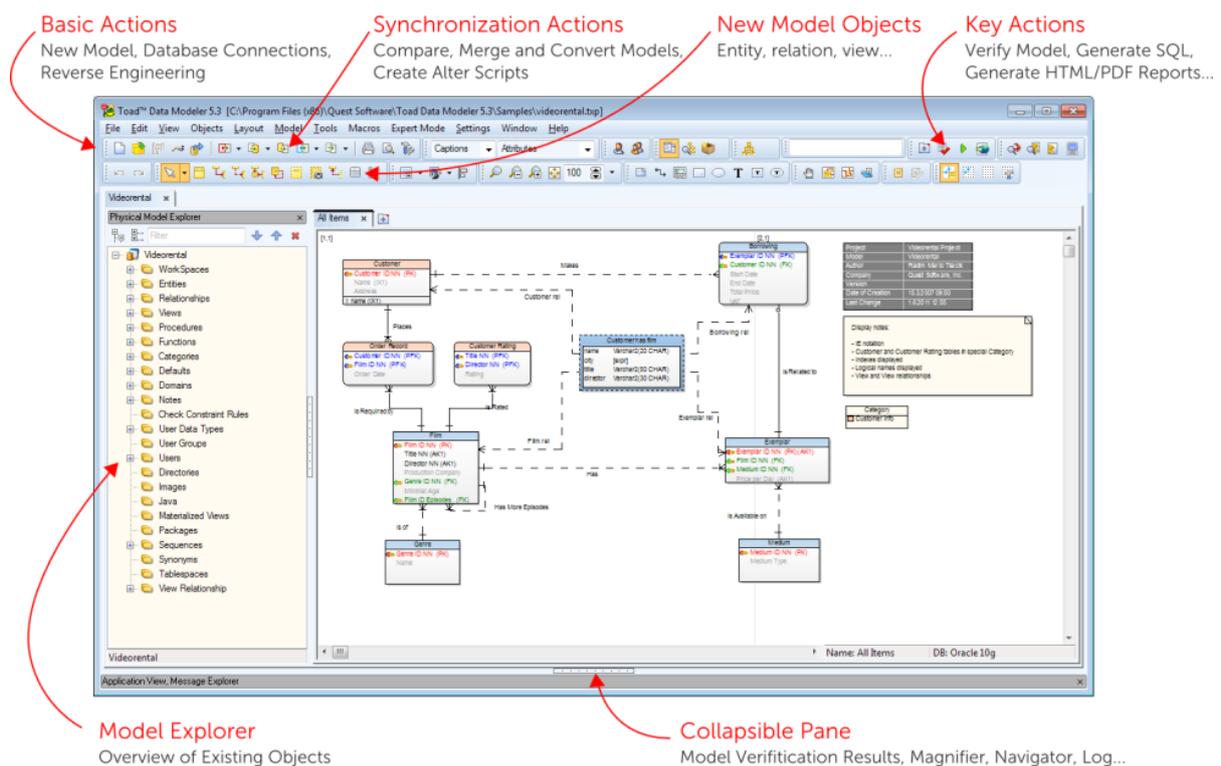
Toad Data Modeler offers these basic ways how to work in it and control all its features and functions:

- Menus
- Toolbars
- Toad Data Modeler Application Layout

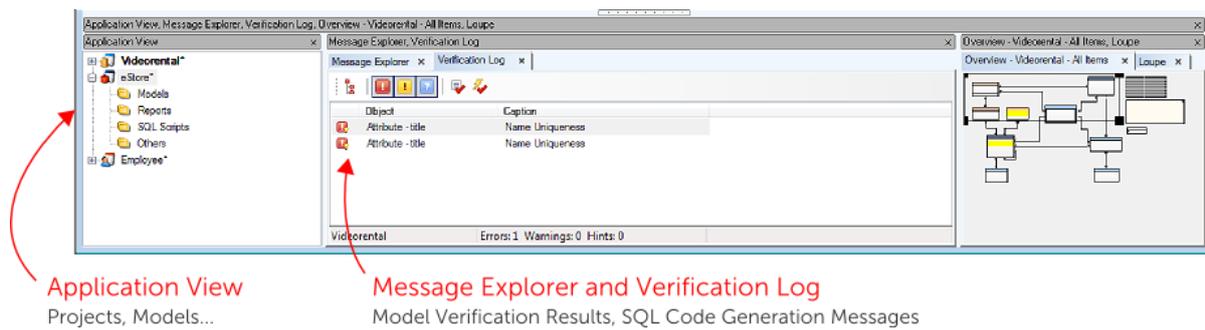
## Toad Data Modeler Application Layout

### Simple and Minimalist Layout

Layout of Toad Data Modeler can be both simple and complex. Various panes can be docked on the left side, the right side or at the bottom of the application and some of the areas can be collapsed using tiny buttons in the middle of window splitters. The default layout for basic operations and tasks can be similar to this:



# Expanded Area



## Application View

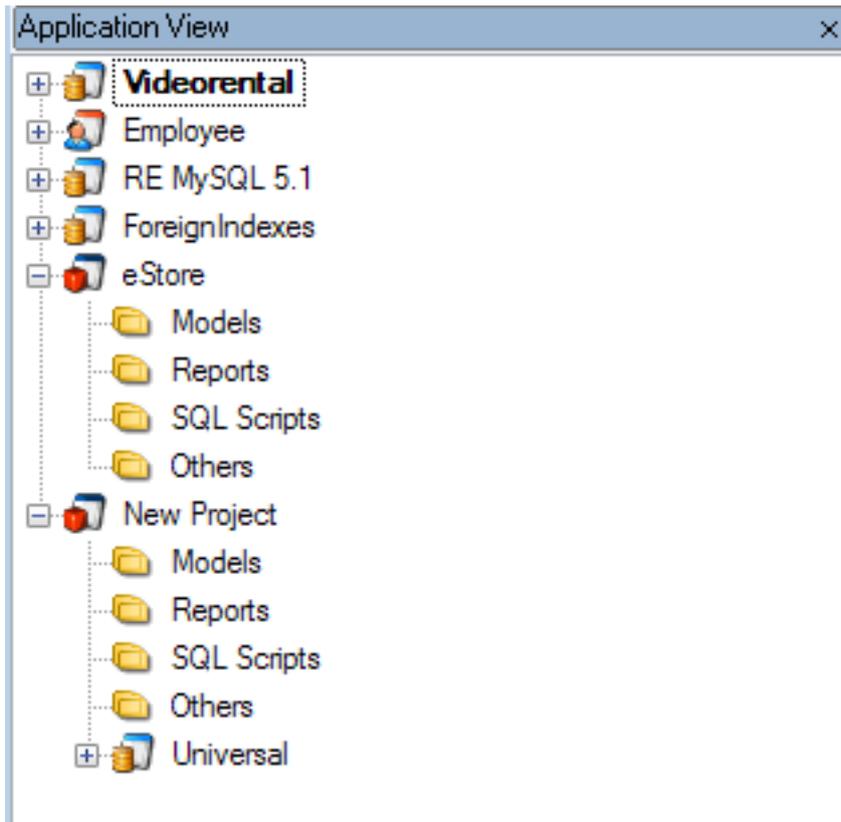
All currently opened projects, models and their workspaces are listed in the Application View (AV). There, you can:

- Manage your models (activate, save, close, etc.)
- Edit model names (**F2** or **Rename**)
- Work with Workspaces and Designers

Application View is by default docked on the bottom left side, next to the **Message Explorer**. You can close it at any time and open it later.

### To open the Application View

Click  on the **Window Toolbar** (or go to **Window Menu | Application View**).



TIP:

- You can dock Application View to a custom position - see [Docking](#) for more information.
- When you need to find a specific model/workspace in Application View, expand the Model/Workspaces items and start typing the name of the item you're looking for. It will be found and highlighted. This function does not look for items in collapsed notes, that's why you need to expand them first.
- Application View also displays versions and revisions of items checked out from **Version Manager**.

## Application View Right-Click Options

**Right-click a Model to see the following options:**

Option	Description
Activate Model	Highlights the selected model and enables menu and toolbar options for the model.
Add Workspace	Adds a new workspace (WS) to the selected model. You can define the workspace name, description and objects in the automatically opened <b>Workspace Properties</b> dialog.

Option	Description
	<p><b>i</b> TIP: To disable this function, go to <b>Settings Menu   Options   General</b> and uncheck the <b>Open Workspace Properties Dialog after Add Workspace</b> checkbox.</p>
Model Properties	Opens the <b>Model Properties</b> dialog.
Save	Saves the model.
Save as	Allows you to save the selected model in another file format or to another location.
Object Viewer	Displays a complete list of model objects, e.g. attributes, triggers, check constraints, views, etc. See <b>Object Viewer</b> for more information.
Add to Project	Adds the model to a chosen Project.
Rename	Renames the model.
Close Model	Closes the selected model. If any changes have been made in the model, a confirmation dialog is displayed.
Create Indexes to All Foreign Keys	Creates Indexes for all foreign keys that don't already have one. <b>i</b> Note: This option is only available for Universal and Physical models.
Infer Indexes of Foreign Keys	Binds suitable Indexes of foreign keys to an entity relationship. <b>i</b> Note: This option is only available with Universal and Physical models. Indexes can only be bound to Non-Identifying Relationship.
In <b>Expert Mode</b> , additional options are available:	
Add to Version Manager	Allows you to add the model to <b>Version Manager</b> .
Test Model	Runs internal test of model consistency.
Repair Model	Tries to fix errors found during Test Model. Fixes roughly 60 % of problems. <b>i</b> Note: Verify Model versus Test Model: <b>Verify Model</b> checks your model from the modeling point of view. If your model is verified and you still encounter problems, you can run <b>Test Model</b> . Test messages will be displayed in the <b>Message Explorer</b> where they can be saved to a text file.

**Right-click the Workspaces node to see the following options:**

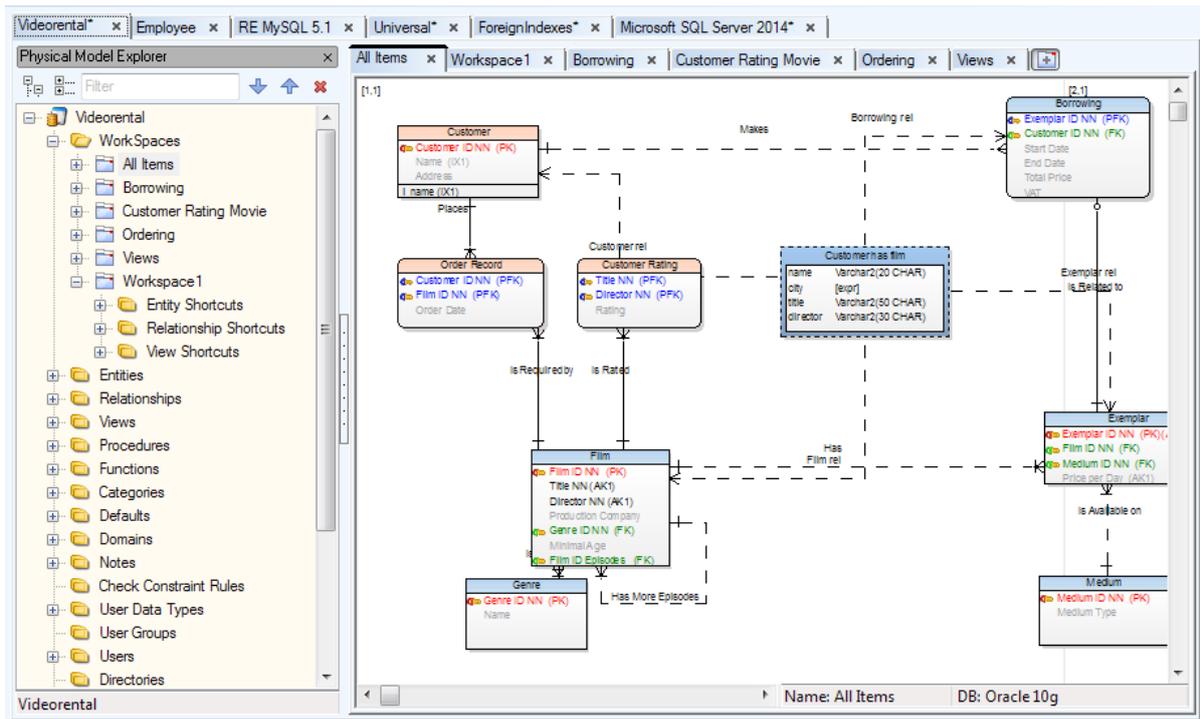
Option	Description
Add Workspace	Creates a new workspace (WS) in the model.
Open All Designers	Opens Designers for all existing Workspaces in the model.

**Right-click a specific Workspace to see the following options:**

Option	Description
Activate Designer	Activates already opened Designer of the selected WS, or opens Designer for this WS in the Application Window.
Open Designer	Opens another Designer for the selected WS.
Edit	Opens the <b>Workspace Properties</b> dialog where you can define the name of the WS and description.
Model Properties	Opens the <b>Model Properties</b> dialog.
Workspace Format	Opens the <b>Workspace Format</b> dialog.
Copy Workspace Layout to	Copies this layout to another <b>Workspace</b>
Add All Model Objects to Workspace	Adds shortcuts of Model objects to the selected workspace.
Fill Relationships to Workspace	Adds shortcuts of relationships to the selected WS. Note that this function only adds shortcuts of existing relationships between objects in case they are not present on the selected workspace.
Optimal Style for All Lines	Seeks an optimal relationship line arrangement and adjusts them accordingly.
Straighten All Lines	Straightens all relationship lines wherever it is possible.
Copy Workspace	Creates a copy of the selected WS.
Rename	Renames the Workspace name.
Delete Workspace	Deletes the selected WS from model.

## Application Window

Application Window (AW) represents a work area where you design your models, work with scripts etc. Toad Data Modeler allows you to work with several models of the same or different databases simultaneously. The opened models, their workspaces and even Script Editor or Scripting Window are organized using the system of tabs.



The system of tabs used by the Application Window can be divided into two levels:

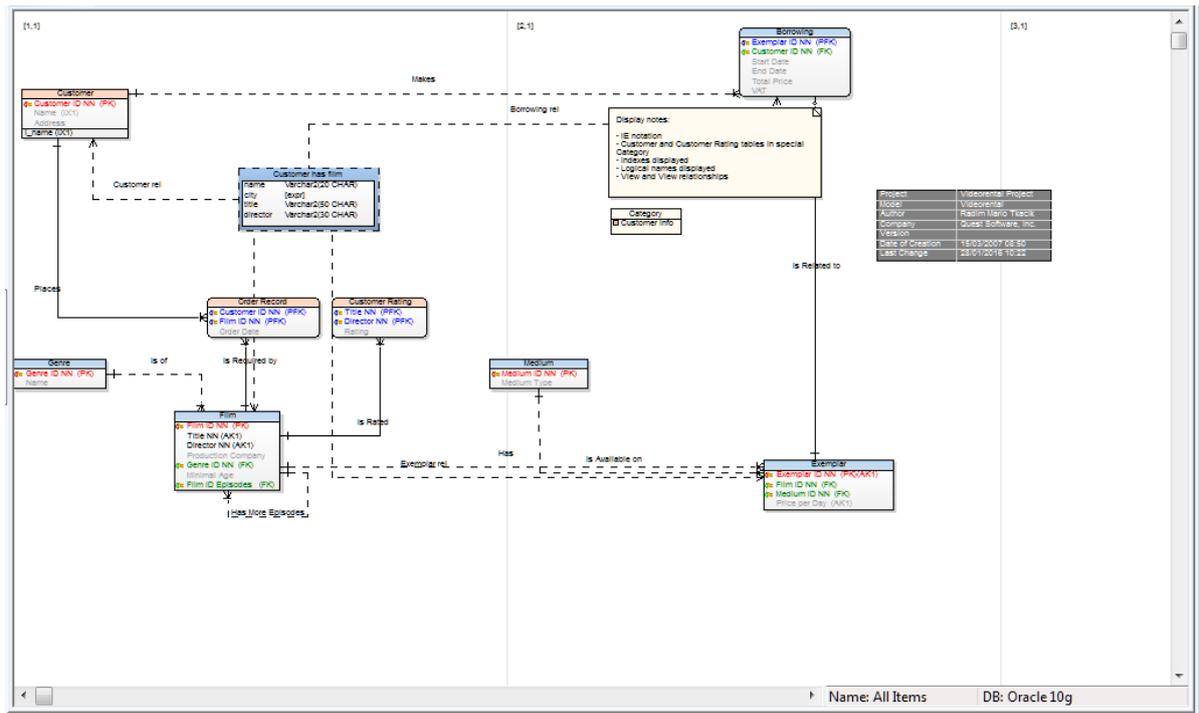
- **Top level tabs** - Models, Script Editor, Scripting
- **Sub level tabs** - Designers (for Models), Scripts (for Script Editor, Scripting)

Closing a top level tab will close all of its sub level tabs (e.g. closing a model tab will close all its designer tabs)

## Designer and Workspace

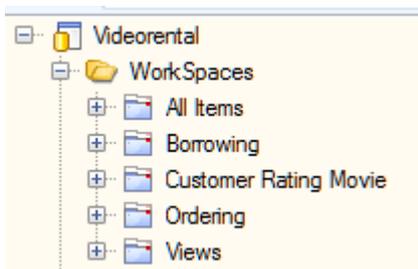
### Designer

- Designer is the area where a workspace is displayed.
- One workspace can be displayed in multiple designers (e.g. each designer can display a part of an especially large workspace)



## Workspace

- Workspaces can be seen as a sub-models. They may contain all or just some of the model objects.
- A default workspace is automatically created for all models - *All Items* - and it contains all of the model objects.
- A model can have several workspaces. In case of large models, it is often convenient to divide a model into multiple workspaces for easier management.



**i** TIP: It is possible to undock designers, for example if you need to work with designers of multiple models at once (you can also simply run **multiple instances** of Toad Data Modeler).

### To create a new workspace in your model, you can:

- Click  on the tabs toolbar
- Go to **Model Menu | Workspace** and select **New Workspace**

- **Right-click** the model in **Model Explorer** or **Application View** and select **Add Workspace**

The **Workspace Properties** dialog opens automatically. In this dialog, you can define the name of the workspace, its description and which objects it should contain.

**i** TIP: To not display the **Properties** dialog every time you create a new Workspace, go to **Settings Menu | Options | General** and uncheck the **Open Workspace Properties Dialog after Add Workspace** checkbox.

### **To open another designer for a workspace**

Right-click the selected workspace in **Model Explorer** (or **Application View**) and select **Open Designer**.

## Navigation on Workspace

- CTRL + scroll mouse to zoom in/zoom out
- CTRL +, CTRL + Page Up to zoom in
- CTRL-, CTRL+ Page Down to zoom out
- Scroll up/down to move up/down
- SHIFT + scroll up/down to move to the right/left
- Hold down the middle mouse button and move the mouse to move the entire workspace
- Page Down or CTRL + down to move to next page
- Page Up, CTRL + up to move to the previous page
- CTRL + left to move to the left page
- CTRL + right to move to the right page
- Click  on **Zoom Toolbar** to fit your entire ERD to screen.
- Press F11 to display the application in full screen mode.

**i** TIP:

Other useful shortcuts are:

- F2 to rename the selected object name/description on workspace.
- Enter to open **Properties** dialog of the selected object(s) on workspace.
- CTRL+F4 to close the currently active Workspace.
- Keyboard arrows to move selected objects on workspace.
- SHIFT + keyboard arrows to resize objects on workspace.

## Workspace Right-Click Options

Option	Description
Edit	Opens the <b>Workspace Properties</b> dialog.
Model Properties	Opens the <b>Model Properties</b> dialog.

Option	Description
Workspace Format	Opens the <b>Workspace Format</b> dialog. See <a href="#">Format Objects</a> for more information.
Copy Workspace Layout to	Copies this layout to another <b>Workspace</b>
Add All Model Objects to Workspace	Adds shortcuts of all objects of the model to the selected WS.
Fill Relationships to Workspace	Adds shortcuts of relationships to the WS.
Optimal Style for All Lines	Changes all relationship lines into letter Z or L shapes while not moving the endpoints.
Straighten All Lines	Straightens all relationship lines where possible while moving the endpoints.
Copy Workspace	Creates a copy of the WS.
Select Objects	Selects objects by category or schema/owner on Workspace. Define a category or owner or category and owner and click <b>Select</b> .
Macros	Offers you available macros to ease your work on the WS.

## Graphic Options on the Workspace

Select **Settings** | **Options** | **Graphics** to access the Settings.  
See [Graphics Options](#) for more information.

## Docking

Docking means moving an UI element (window, toolbar) to a custom position. This feature allows you to customize the Toad Data Modeler interface to meet your needs and requirements.

You can dock the following UI elements:

- **Big windows** (Designer, Script Editor, etc.)
- **Small windows** (Message Explorer, Application View, etc.)
  - **Model Specific** - exists for one model only (Model Explorer)
  - **Non-Model Specific** - exists across multiple models (Application View)
- **Toolbars**

## Docking rules

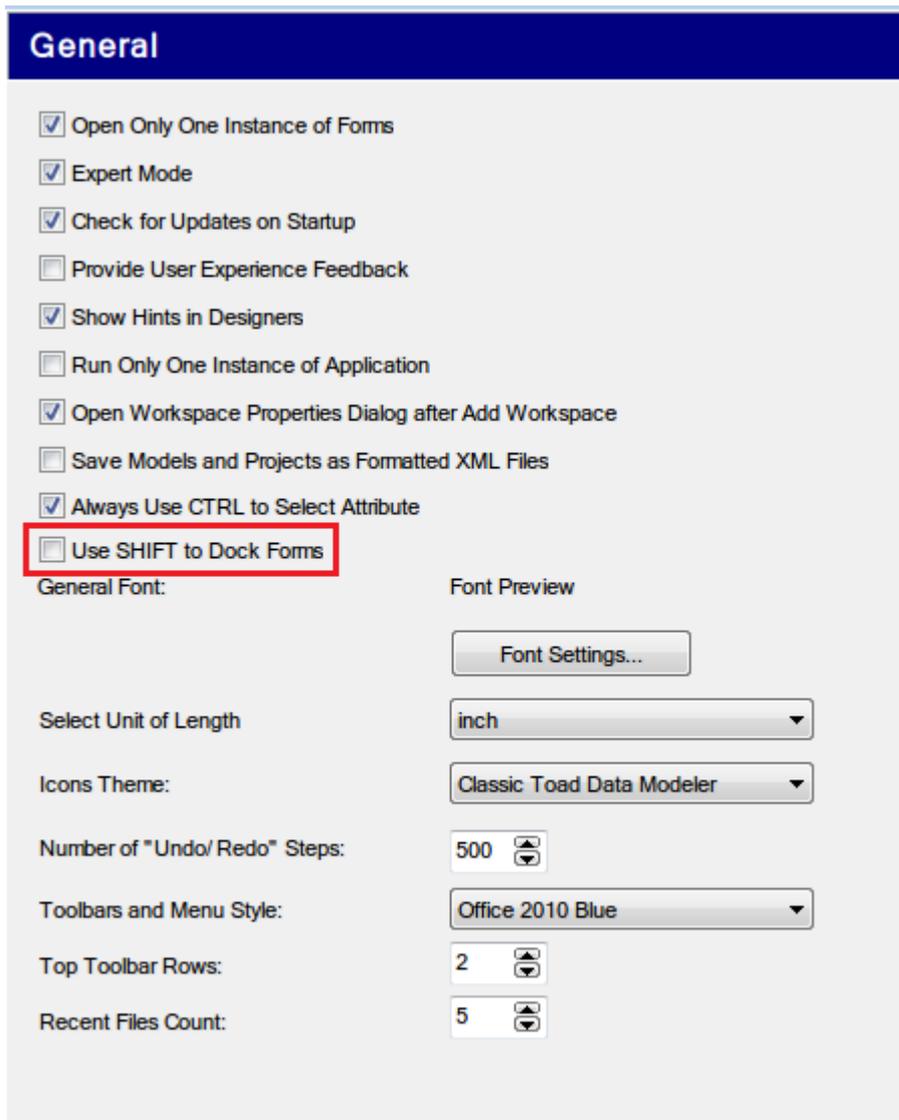
UI Element	Rules
Big windows	<ul style="list-style-type: none"><li>• Can be undocked and moved out of the application</li></ul>
Small windows - Model Specific	<ul style="list-style-type: none"><li>• Can be undocked and moved out of the application</li><li>• Can be docked to <b>right side</b> and <b>left side</b> of a <b>Big Window</b> (Model explorer -&gt; right side of Designer)</li><li>• Can be docked to <b>any side</b> of another <b>Model Specific Window</b> (Object Viewer -&gt; above Model Explorer)</li></ul>
Small windows - Non-Model Specific	<ul style="list-style-type: none"><li>• Can be undocked and moved out of the application</li><li>• Can be docked to <b>any side</b> of a <b>Big Window</b> (Message explorer -&gt; above Scripting Window]</li><li>• Can be docked to any side of another <b>Non-Model Specific Window</b> (Application View -&gt; below Verification Log)</li></ul>
Toolbars	<ul style="list-style-type: none"><li>• Can be undocked and moved out of the application</li><li>• Can be docked to any side of a <b>Big Window</b> (Alignment Toolbar -&gt; right side of Designer)</li></ul>

**i** Note: **Model Specific Windows** cannot be docked to **Non-Model Specific Windows**. The opposite also applies.

### To dock windows using **SHIFT** key

When you are dragging a window, Toad Data Modeler by default shows you available docking positions. If you don't like this behavior, you can enable it only when you are dragging a window and holding down **SHIFT** key simultaneously.

1. Go to **Settings Menu | Options**.
2. In section **General**, check **Use SHIFT to Dock Forms** checkbox.



### ***To dock a window***

Drag a window to the desired position and drop.

### ***To dock a form/pane with SHIFT***

1. Hold down SHIFT key.
2. Drag a window to the desired position.
3. Release the mouse button and then SHIFT to dock the window.

The changed layout will be automatically saved after you close the application (.txe file).

### ***To undock a form/pane***

Press SHIFT and double-click the top of the form (pane)

# Hot Keys

Shortcut	Description
CTRL+N	Opens the <b>New Model</b> dialog.
CTRL+O	Opens already existing model.
CTRL+S	Saves a model.
CTRL+W	Creates a new Workspace (WS).
CTRL+F9	Opens the <b>Model Verification</b> dialog.
CTRL+E	Creates a new entity.
CTRL+R	Creates an identifying relationship.
CTRL+C	Copies selected object(s).
CTRL+X	Cuts selected objects(s).
CTRL+V	Pastes copied or cut object(s).
CTRL+Z	Undo step.
SHIFT+CTRL+Z	Redo step.
CTRL+A	Selects all.
CTRL+F	Opens the <b>Find</b> dialog.
CTRL+ALT+F	Opens the <b>Find in Scripts</b> dialog.
CTRL+M	Minimizes all undocked forms.
Del	Removes selected object(s) from Workspace.
SHIFT+Del	Deletes selected object(s) from model.
ALT+O	The <b>Delete Confirmation</b> dialog - <b>OK</b> button.
ALT+C	The <b>Delete Confirmation</b> dialog - <b>Cancel</b> button.
CTRL+M	Minimizes Forms.
CTRL+ALT+S	Synchronizes metamodel.
CTRL+I	Creates Inheritance (Logical Model).
F1	Opens the Help file.
F2	Renames model/object.
F9	Opens the <b>DDL Script Generation</b> dialog.

<b>Shortcut</b>	<b>Description</b>
F11	Displays Toad Data Modeler in full screen mode.
CTRL+Up	Moves selected object(s) up. (Order of Generated Objects)
CTRL+Down	Moves selected object(s) down. (Order of Generated Objects)
Keyboard arrows	Moves shapes in Workspace (select a shape first).
SHIFT+keyboard arrows	Changes size of entity box.
CTRL+scroll mouse	Zooms in/out.
CTRL+, CTRL+Page Up	Zooms in.
CTRL-, CTRL+Page Down	Zooms out.
SHIFT+scroll mouse	Moves to the right/left on the Workspace.
Scroll mouse	Moves up/down on the Workspace.
Holding down the middle mouse button	Moves in entire page/Workspace.
Page Down, CTRL + down	Moves to next page.
Page Up, CTRL + up	Moves to previous page.
CTRL + left	Moves to the left page.
CTRL + right	Moves to the right page.
SHIFT + drag&drop the heading of a form/pane	Docks the selected form/pane on a particular place in the Application Window.
SHIFT + double-click the heading of a form/pane	Undocks the selected form/pane.
SHIFT	Multiple selection of objects (on the WS, dialogs, forms, Model Explorer etc.)
Drag&Drop techniques+CTRL+SHIFT	Creates a shortcut of object on the WS.
Drag&Drop techniques from Model Explorer	Creates a shortcut of object on the WS.
Drag&drop techniques+CTRL	Copies objects on the WS/Object dialog (e.g. Entities dialog)/Model Explorer.
CTRL+Drag&Drop techniques	Copies attributes between entities in WS.
CTRL+click an attribute	Selects an attribute to move it to another entity box on the WS. (Click the attribute and release CTRL key to

Shortcut	Description
	move the selected attribute.)
CTRL+click attributes	Multiple selection of attributes in entity box on the WS.
CTRL+double-click an object name	Renames the object name on the WS (including attributes).
CTRL+click a relationship line	Adds a handle point.
CTRL+click a handle point	Deletes the selected handle point.
ALT+click a relationship line	Selects a relationship line segment.

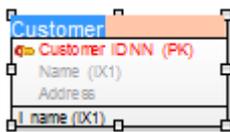
## Inplace Editor

Toad Data Modeler allows you to edit items directly in appropriate place (Workspace, frame). E.g. you can change names of entities, attributes, indexes, relationships, views, categories, stamp items etc. directly on the WS.

See the following examples.

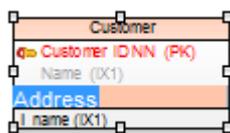
### To change an entity name on the Workspace

1. Click an entity on the Workspace.
2. Press F2.



### To change an attribute name on the Workspace

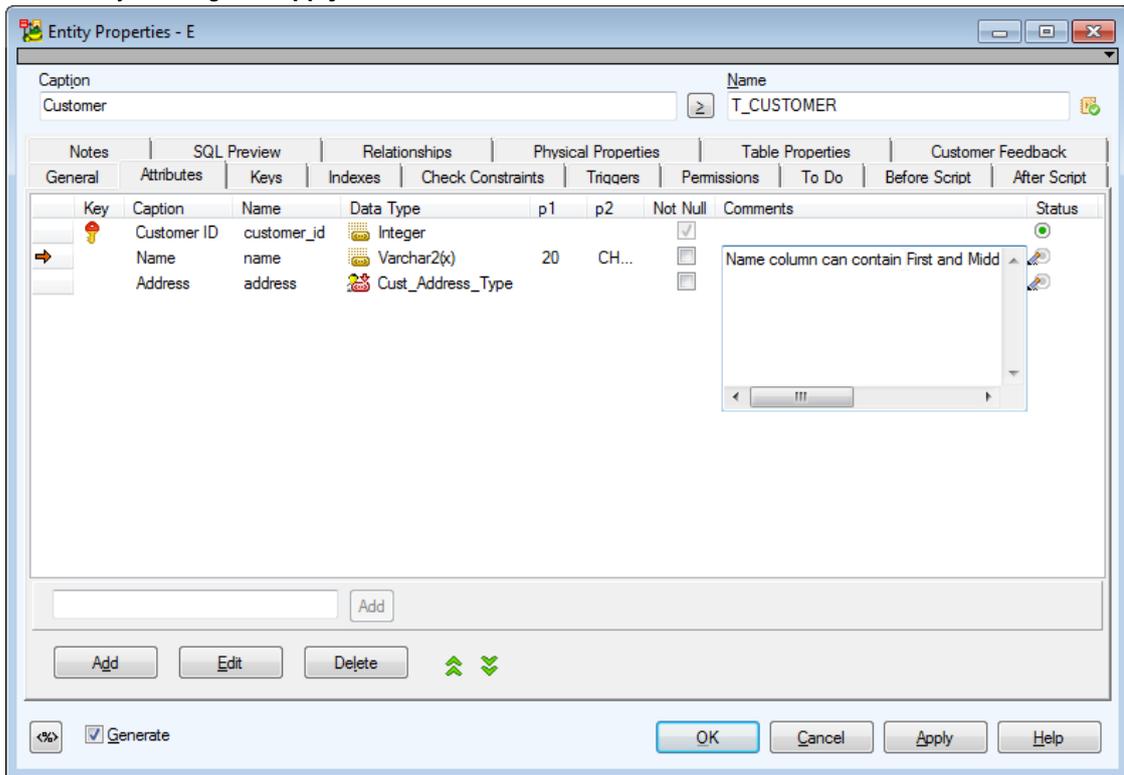
1. Click an entity on the Workspace.
2. Press CTRL and click the attribute you want to rename.
3. Press F2 or hold the mouse left button for a while without moving (as in Windows).



### To edit attribute properties directly in an entity frame:

1. Open the **Entity Properties** form.
2. Select the attribute property that you want to edit and press **F2** (or click the item that you want to change, wait a second and click it again). Note that not every property is editable from this form.

3. Change the value and click somewhere else in the grid.
4. Confirm by clicking on **Apply**.



## Message Explorer

In Message Explorer you can see hints, errors, warnings and other messages that appear during your work with Toad Data Modeler.

Message Explorer is by default docked at the bottom of the application window. However, you can hide or close it at any time.

### To open Message Explorer again

Select **Window | Message Explorer Log**.

Id	Date	Time	Message
1	7.8.2007	11:33:33	Connecting to database
2	7.8.2007	11:33:33	Connecting to Oracle version: 10.2.0.1.0
3	7.8.2007	11:33:33	Fetching tables from schema SYSTEM...
4	7.8.2007	11:33:34	Disconnected from database
5	7.8.2007	11:33:56	Connecting to database.
6	7.8.2007	11:33:57	Reversing table details...
7	7.8.2007	11:33:58	Reversing table comments...
8	7.8.2007	11:33:58	Reversing columns...
9	7.8.2007	11:34:00	Reversing primary keys...

You can sort messages by IDs, date, time and message type, simply click on the appropriate column.

## Message Explorer versus Log Area

Log area in appropriate forms/dialogs/wizard displays information only on the operation taking place in the particular form/dialog/wizard. (Show/Hide Log options are available.)

Message Explorer displays information about all ongoing operations.

## Message Explorer Right-Click Options

Right-click the selected message to see the following options:

Id	Date	Time	Message	
19	7.8.2007	11:34:05	Reversing synonyms...	
7	7.8.2007	11:33:58	Reversing table comments...	
6	7.8.2007	11:33:57	Reversing table details...	
15	7.8.2007	11:34:05	Reversing triggers...	Details
20	7.8.2007	11:34:06	Reversing users...	View Options
14	7.8.2007	11:34:03	Reversing views...	Clear Messages
26	7.8.2007	11:35:31	Verification Videorental	Save Messages
31	7.8.2007	11:36:11	Verification Videorental	Save Selected Messages

Option	Description
Details	<p>Displays details on the selected message.</p> <p><b>i</b> TIP: Double-click the selected message opens the details too. Use <b>Next</b> and <b>Previous</b> buttons for quick navigation among messages.</p>
View Options	<p>Opens the <b>Message Explorer Properties</b> dialog where you can define:</p> <ul style="list-style-type: none"> <li>• What information you want to see in the Message Explorer (Date, Time, Format, Type),</li> <li>• Type of messages (Errors, Warnings, Information, Hints etc.),</li> <li>• Path to save the Log.txt file on tab <b>Message Saving</b>.</li> </ul> <p><b>Save Messages to File</b> - If this checkbox is checked, the messages displayed in Message Explorer will be saved to the Log.txt file continuously.</p> <p><b>Overwrite File</b> - If this checkbox is checked, new messages overwrite old messages. If this checkbox is unchecked, new messages are listed after the older messages in the Log.txt file.</p> <p>By default, the following path is set up in the <b>Settings</b> menu   <b>Options</b>   <b>Paths</b>   <b>Message Explorer Log Path</b>:            C:\Documents and Settings\%UserName%\My Documents\Toad Data Modeler\Log</p>
Clear Messages	<p>Clears the Message Explorer.</p> <p><b>i</b> Note: Undo/Redo cannot be applied here.</p>

Option	Description
Save Messages	Opens the <b>Save As</b> dialog via which you can save all messages to LocalLog.txt file.
Save Selected Messages	Opens the <b>Save As</b> dialog via which you can save the selected messages to LocalLog.txt file.

## Model Explorer

In Model Explorer, you can see all objects that exist in your model.

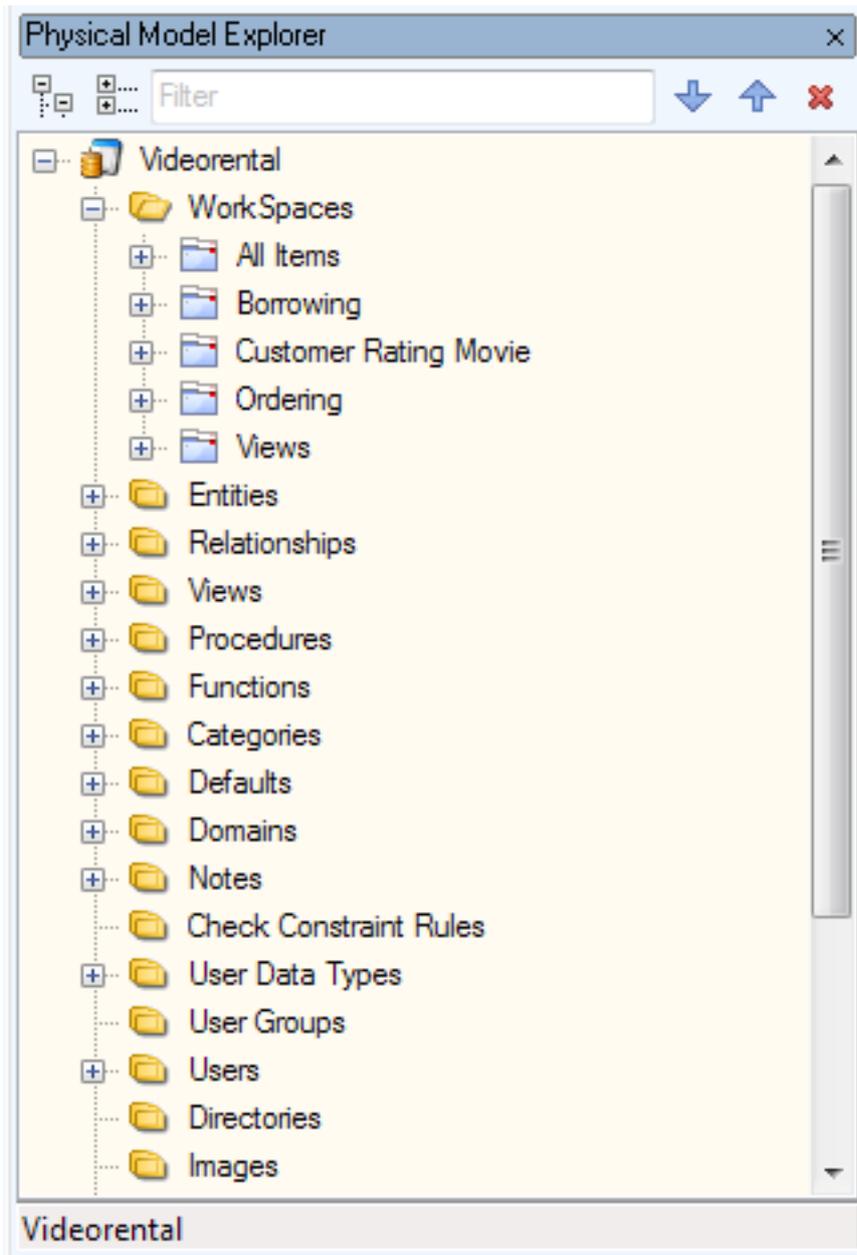
Model Explorer is by default docked on the left side of the **Application Window**. You can always undock it and move it somewhere else or close it.

### *To display Model Explorer*

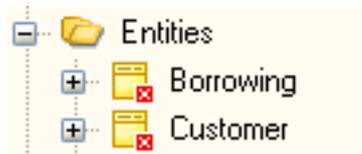
Click  on the toolbar or go to **Window Menu | Model Explorer**.

The contents of Model Explorer depend on the type of your model (Physical/Universal/Logical) and the used database platform and version.

Example of Physical Model Explorer of Oragel 10g model:



**i** Note: Objects of your model that have the **Generate** checkbox disabled in their **Properties** dialogs are displayed in Model Explorer this way:

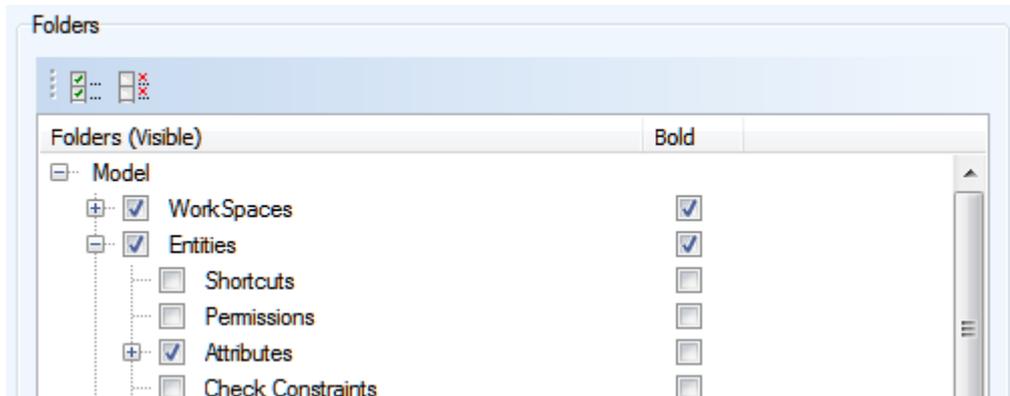


## Model Explorer settings

You can access the settings by **right-clicking** in **Model Explorer** and selecting **Settings**.

In settings, you can change the background color of the Explorer or enable the **Use Colors of Category to Draw** option to color items in the Explorer according to their category.

You can also **hide/show** specific item categories or make them **bold** to make the important objects stand out.



## Managing Items in Model Explorer

In Model Explorer, you can:

- View objects **Properties** dialog by double-clicking them
- Place object shortcuts on workspace (or even workspace of another model) using drag and drop
- Create a copy of the object in the Explorer or on workspace using CTRL + drag and drop
- Locate shortcuts of an object on workspace easily (using **Find on Workspace** context menu option)

## Object Navigator Dropdown Menu

Object Navigator Dropdown Menu, together with the modeless dialogs, can ease and speed up your work in Toad Data Modeler significantly.

Object Navigator Dropdown Menu:

- Is available in Object **Properties** dialogs and forms (entity, attribute, check constraint, view, function, users, user groups etc.).
- Allows you to select items for edit from **one (the same) place**.

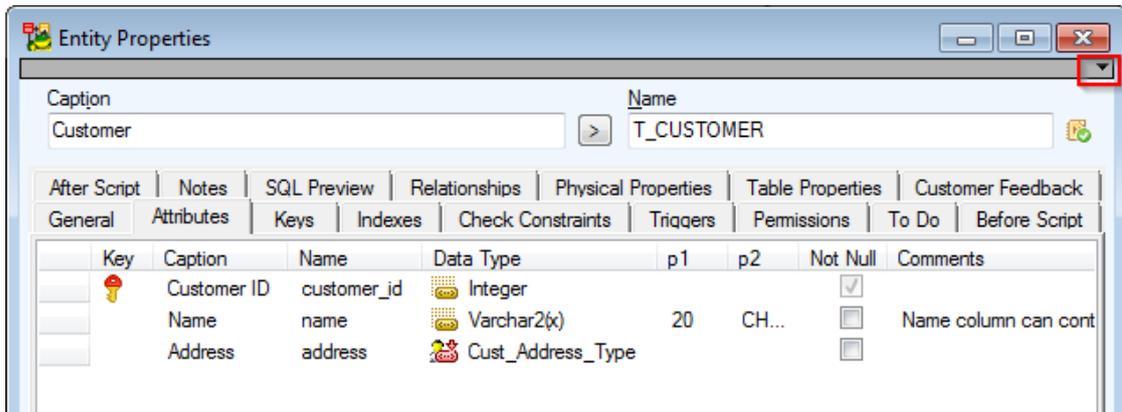
---

### Scenario

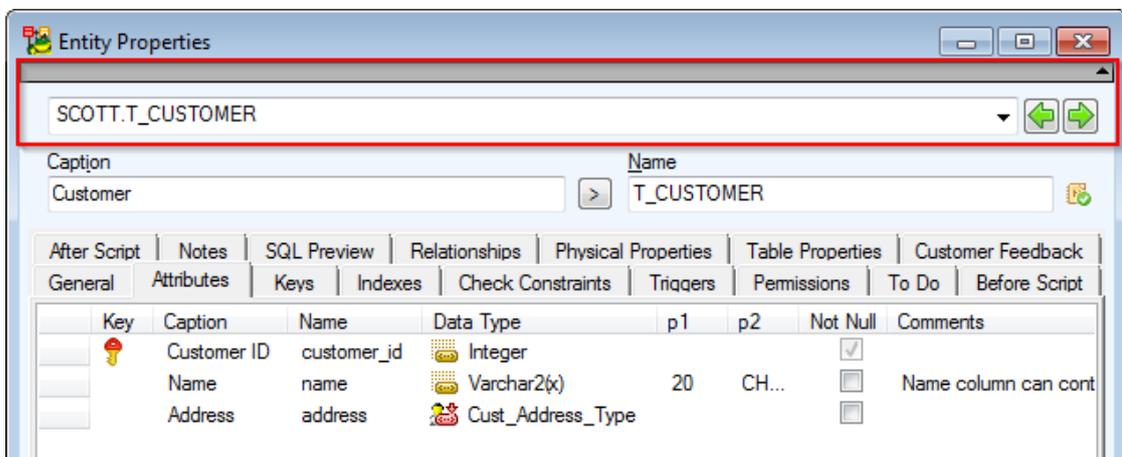
You need to edit several entities of your model.

---

1. Double-click ANY entity on the Workspace to edit it.
2. Click the small black arrow on top right-hand corner.



3. The form navigator area appears on the form.



4. From the Object Navigator Dropdown, select the entity you need to edit (*Borrowing*).
5. Change its properties and confirm **Apply**. -> The changes will be saved and the **Entity Properties** form remains opened.
6. Again, from the Object Navigator Dropdown, select another entity you need to edit (*Customer*). Confirm **Apply** to save the changes.
7. Take the same steps until you edit all entities you need.
8. For the last entity, you can confirm the changes **OK** to close the **Entity Properties** form.

Feel free to take advantage of this box when editing relationships, procedures, views, functions, users etc.

## Object Types and Properties - OTPs

OTPs (Object Types and Properties) enable you to select objects and properties on various forms, define default OTPs selection and store and load them.

**Example:** See the **Sync & Convert Wizard**, the **Select Object Types** page, **Detailed Settings** button where you can define object types and properties for model compare/merge/conversion. If a particular object is selected (e.g. entities, relationships, functions etc.), it means you want to compare the object between the Left model and Right model. (All objects are selected by default). uncheck the checkboxes at particular objects if you do not want to compare them and therefore ignore the possible changes.

By default, only basic selection of objects and properties is available. Saving, loading and creating default OTPs are options for Experts.

Object Types and Properties are available in:

- Reverse Engineering Wizard and Model Update Wizard - **What to Reverse** page
- DDL Script Generation dialog - **What to Generate** tab
- HTML/RTF Report Wizard - **What to Report** page
- Sync & Convert Wizard - **Select Object Types** page
- Model Verification - **What to Verify** tab

## Default (System) Selected OTPs versus User Selected OTPs

In Toad Data Modeler there are:

- **Default (System) Selected OTPs**

They are stored by default at:

C:\Program Files\Quest Software\Toad Data Modeler\Selected OTPs

They cannot be modified/overwritten.

- **User Selected OTPs**

They are stored by default at:

C:\Documents and Settings\user name\My Documents\Toad Data Modeler\Installation name  
\Selected OTPs

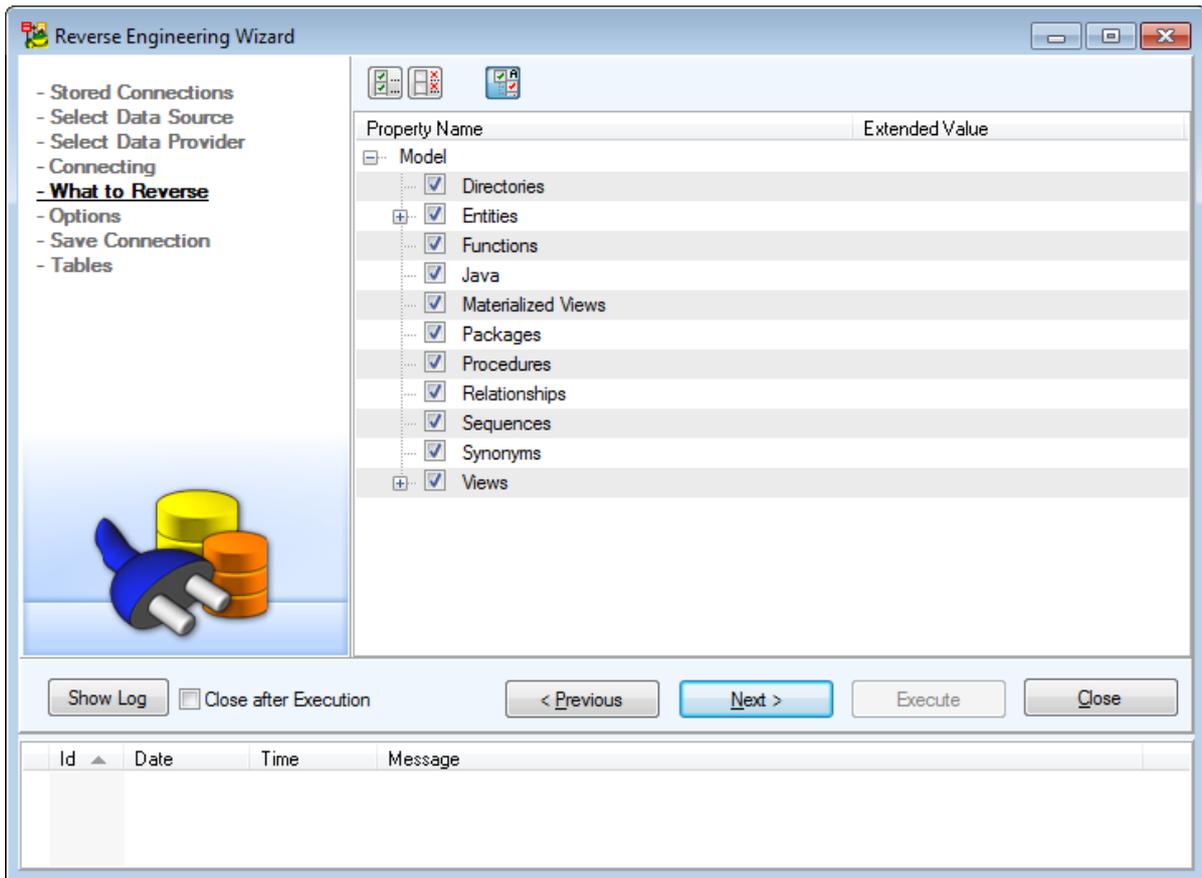
They are preserved during update of Toad Data Modeler.

To set/change the path for user OTPs, select **Settings | Options | Paths | Advanced** tab | **Selected OTPs**.

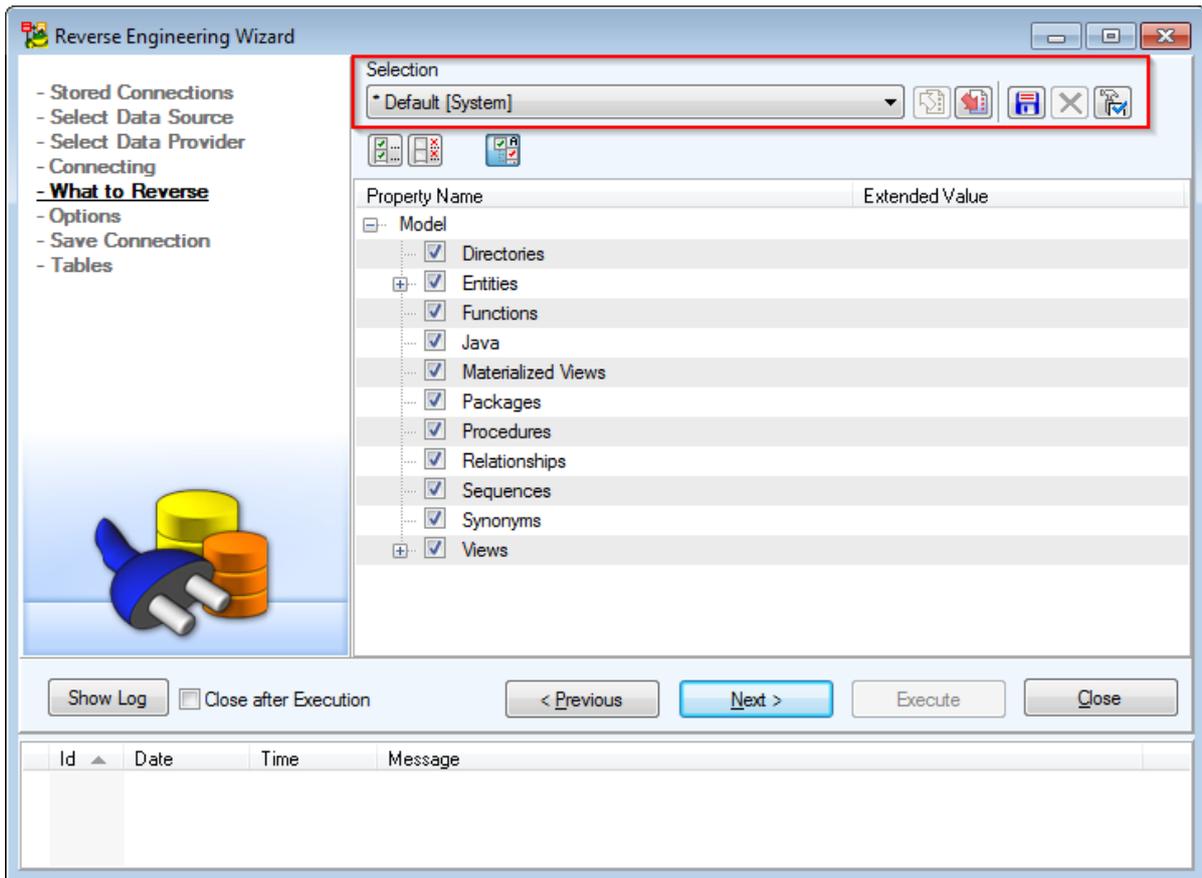
## Define and Save User OTPs

See the differences between the following two dialogs (Expert Mode off versus Expert Mode on).

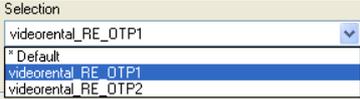
**RE Wizard** (Oracle 10g db), **What to Reverse** page, default OTPs (Expert Mode disabled):



**What to Reverse** page, OTPs options (Expert Mode enabled):



In this dialog, you can see options for managing OTPs:

Option	Description
	Select an OTP.
	Saves the changed OTP settings to the OTP selected in the <b>Selection</b> box (either resaves the current OTP or overwrites another OTP).
	Loads OTP settings of the OTP selected in the <b>Selection</b> box.
	Saves the selection to a new OTP (similar to Save as). Define a name of the new OTP and press <b>OK</b> . After you click <b>OK</b> , the OTP will be saved to .txo file to particular database (Oracle 10g.txo in our example).
	Deletes the selected OTP.

Option	Description
	Click <b>Yes</b> to delete the OTP from the .txo file of particular database (Oracle 10g.txo in our example).



Sets the selected OTP as default.

**i** Note: The OTP options are the same also in other dialogs and wizards - e.g. **DDL Script Generation** dialog or **Report Wizard** etc.

## Save User OTPs

Each database supported in Toad Data Modeler has its .txo file (Oracle 10g.txo, Microsoft SQL Server 2005.txo etc.).

Default (System) Selected OTPs are stored by default at: C:\Program Files\Quest Software\Toad Data Modeler\Selected OTPs.

As soon as you install Toad Data Modeler, the default .txo files will copy to the user section at: C:\Documents and Settings\user name\My Documents\Toad Data Modeler\Installation name\Selected OTPs. Here, you can store your user OTPs that will be preserved during update of Toad Data Modeler.

New OTPs, which you create, will be saved in .txo file of the particular database.

### Example:

You create a new OTP in the **DDL Script Generation** dialog for your Oracle 10g model and save it. This OTP will be saved to Oracle 10g.txo. This OTP will be available in the **Selection** box in the **DDL Script Generation** dialog for all Oracle 10g models for which you need to generate SQL script. (OTPs are saved separately for every dialog, which means that OTPs defined in DDL Script Generation dialog will not be available in Model Verification dialog etc.)

If you want to save this OTP to a particular model, you have to click **Save Settings**. The OTP, including other settings defined in the **DDL Script Generation** dialog, will be saved within a model. Next time, when you open the **DDL Script Generation** for the model, the OTP will be selected automatically. If you didn't click **Save Settings**, you would have to select the OTP and load it again.

## Available OTPs Dialog

### *To see a list of all OTPs available in your model*

Select **Expert Mode | Expert Mode Settings** menu | **Available OTPs**.

This option is available for real experts interested in writing their own support for new database systems or for people who need to enhance existing support for the selected database system.

## Object Viewer

Toad Data Modeler allows you to display a complete list of selected objects of your model, e.g. attributes, triggers, check constraints, views, etc. with a possibility to open the **Properties** dialog of the selected object(s), delete them or find on the Workspace.

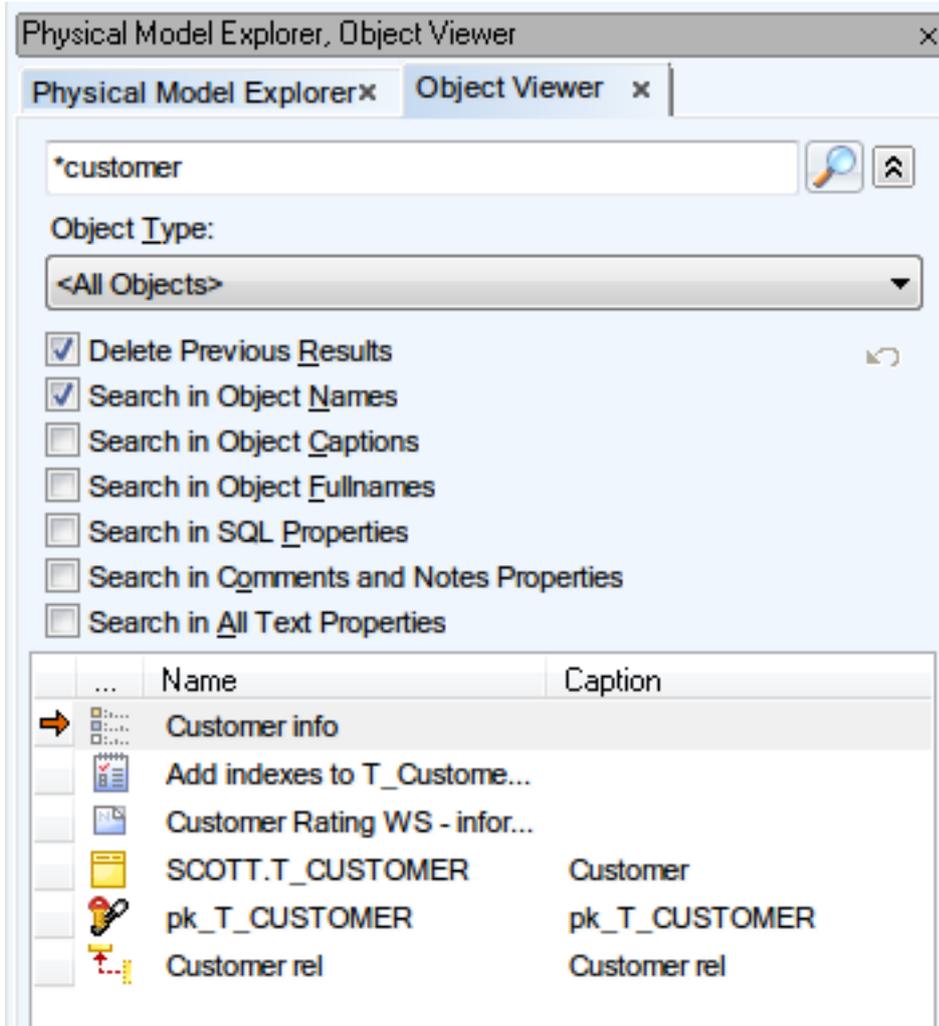
### To open Object Viewer

Click  on the toolbar.

or

Select **Window | Object Viewer...**

Object Viewer is docked by default next to the Model Explorer.



Option	Description
Object Name	Type an object name (case sensitive).
Object Type	Select a type of object. Select options for the search process in the checkboxes below.
	Click this icon to view the objects.

In the list, you can see:

- Icon of an object type
- Physical object name
- Logical object name
- Information about parent object
- Object type (entity, attribute, key, trigger etc.)

Right-click the selected item in the list to display other options (edit, delete, find on Workspace).

## Status of Items in Grids

Various types of status are shown in the following examples of attributes and entities.

### 1. Normal state

	Name	Caption	Data Type	p1	p2	Not Null	Status
➔	customer_id	Customer ID	Integer			<input checked="" type="checkbox"/>	●
	address	Address	Cust_Address_...			<input type="checkbox"/>	●
	name	Name	Varchar2(20)	20		<input type="checkbox"/>	●

### 2. Attribute *Customer ID* is being edited. - The item is in grey.

	Name	Caption	Data Type	p1	p2	Not Null	Status
➔	customer_id	Customer ID	Integer			<input checked="" type="checkbox"/>	⊙
	name	Name	Varchar2(20)	20		<input type="checkbox"/>	●
	address	Address	Cust_Address_...			<input type="checkbox"/>	●

### 3. A new attribute has just been added, the change has not been confirmed yet. (After the change is confirmed, the status will change and set to Normal. Until creation of the new item is confirmed, it's not possible to edit it.)

	Name	Caption	Data Type	p1	p2	Not Null	Status
	customer_id	Customer ID	Integer			<input checked="" type="checkbox"/>	●
	name	Name	Varchar2(20)	20		<input type="checkbox"/>	●
	address	Address	Cust_Address_...			<input type="checkbox"/>	●
➔	Attribute1	Attribute1	Char(20)	20		<input type="checkbox"/>	+

### 4. The attribute has been modified in the grid directly (see the changed name). The change has not been confirmed yet.

	Name	Caption	Data Type	p1	p2	Not Null	Status
	customer_id	Customer ID	Integer			<input checked="" type="checkbox"/>	●
	name	Name	Varchar2(20)	20		<input type="checkbox"/>	●
	address	Address	Cust_Address_...			<input type="checkbox"/>	●
➔	New Attribute	New Attribute	Char(20)	20		<input type="checkbox"/>	✎

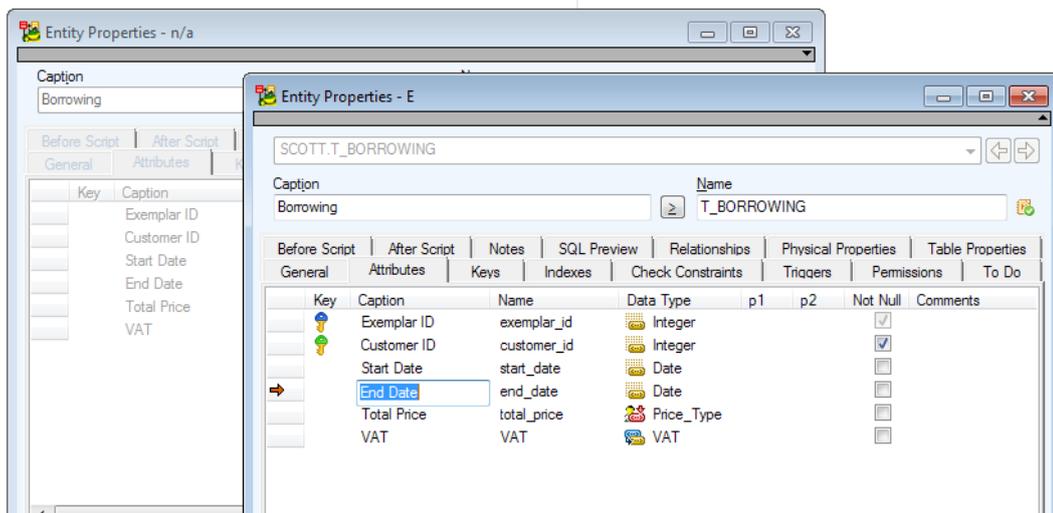
5. Item *NewAttribute* has just been deleted, the change has not been confirmed yet.

	Name	Caption	Data Type	p1	p2	Not Null	Status
	customer_id	Customer ID	Integer			<input checked="" type="checkbox"/>	
	name	Name	Varchar2(20)	20		<input type="checkbox"/>	
	address	Address	Cust_Address_...			<input type="checkbox"/>	
➔	New Attribute	New Attribute	Char(20)	20		<input type="checkbox"/>	

6. Deletion of item *NewAttribute* has been confirmed by clicking the **Apply** button. The item is marked as deleted and will not be displayed when you open the **Entity Properties** form/**Attributes** tab next time.

	Name	Caption	Data Type	p1	p2	Not Null	Status
	customer_id	Customer ID	Integer			<input checked="" type="checkbox"/>	
	name	Name	Varchar2(20)	20		<input type="checkbox"/>	
	address	Address	Cust_Address_...			<input type="checkbox"/>	
➔	New Attribute	New Attribute	Char(20)	20		<input type="checkbox"/>	

7. You've opened two instances of the same **Entity Properties** form and started editing properties in one of them. That one will be marked by **E** and will be editable, while the other form will be marked by **n/a** and won't be editable.



**i** Note: By default, you can open only one instance of a form. To open more instances at the same time, go to **Settings | Options | General** and uncheck the **Allow One Instance of Form** checkbox.

## Menus

The following menus are available in Toad Data Modeler:

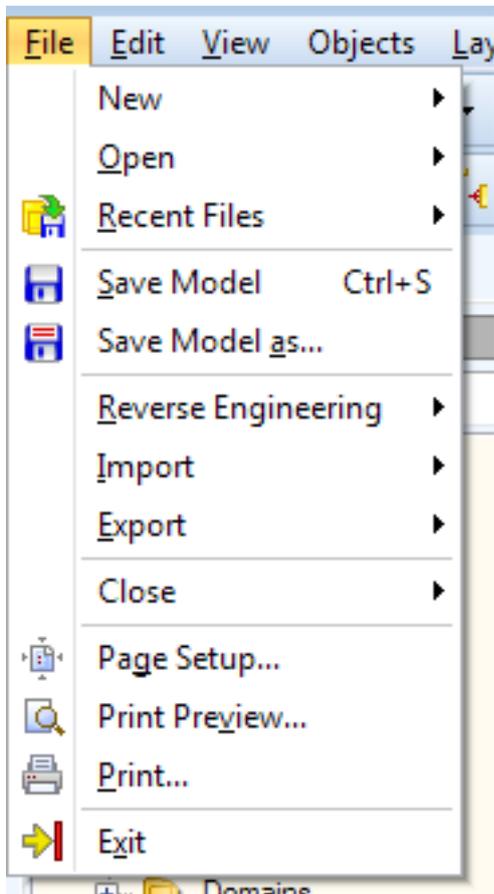
- [File](#)
- [Edit](#)
- [View](#)

- Objects
- Layout
- Model Menu
- Tools Menu
- Macros Menu
- Expert Mode Menu
- Settings Menu
- Window Menu
- Help Menu

**i** | **TIP:** Find any menu option easily by typing its first letters into **Jump To...**

## File

The following options are displayed when there is at least one Designer opened inside the Application Window.



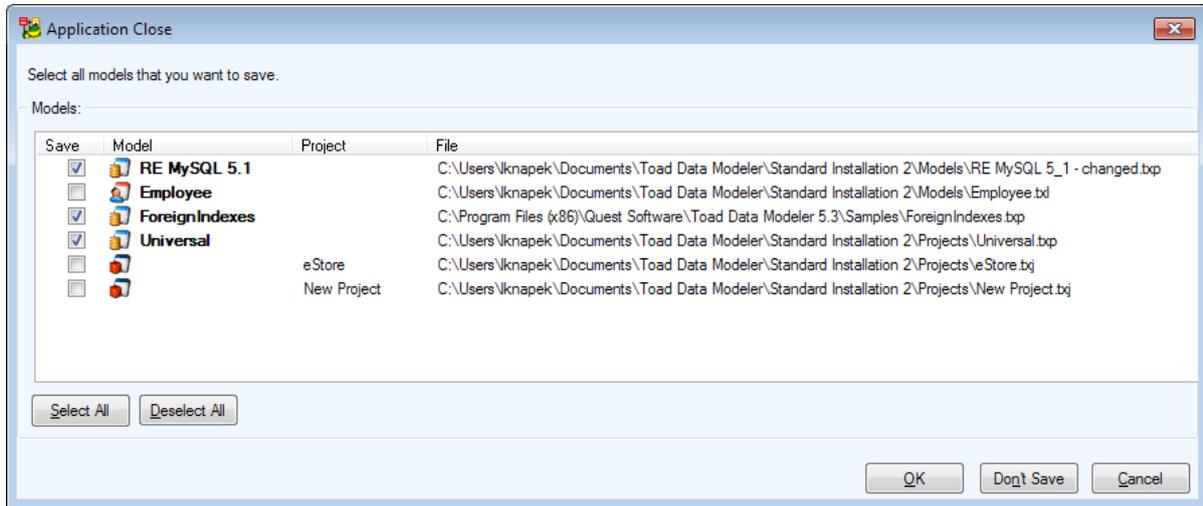
Option	Description
New   Project	Creates a new project, displays the <b>New Project</b> settings dialog.
New   Model	<p>Opens the <b>New Model</b> dialog.</p> <p>Supported models:</p> <ul style="list-style-type: none"> <li>Physical Data Model</li> <li>Universal Data Model</li> <li>Logical Data Model</li> </ul> <p><b>i</b> TIP: Right-click the dialog to select the display of the options (Large Icons, Small Icons, List).</p> <p><b>Model Name</b> - Define a name of your model. (Also, you can change the name later in the Application View or Model Explorer (press F2).)</p> <p><b>Notes:</b></p> <p><b>i</b> Note:</p> <ul style="list-style-type: none"> <li><b>Database Name</b> - A database for which the model has been created (e.g. Oracle 10g). The database name information can be found at the bottom of the Application Window, or it is displayed in the pop-up hint when you point your mouse cursor at the Model Name in the Application View or Model Explorer.</li> <li><b>Model Name</b> - In Toad Data Modeler, Model Name should be understood as a title of a document that is saved.</li> <li><b>File Name</b> - A name of file under which the model is saved. File Name is defined after you select <b>Save Model</b> or <b>Save Model as</b>.</li> </ul>
New   Gallery	<p>Opens the <b>Gallery Edit</b> dialog.</p> <p>You can create Gallery to store frequently used parts of your models such as entities, attributes, stored procedures and other objects. You can then access these parts from any project.</p>
Open   Project	Displays the <b>Open</b> dialog in your default Projects folder.
Open   Model	Displays the <b>Open</b> dialog in your default Models folder.
Open   Samples	Displays the <b>Open</b> dialog in your default Samples folder.

Option	Description
Open   Gallery	Displays the <b>Open</b> dialog in your default Gallery folder.
Open   Sample Gallery	Displays the <b>Open</b> dialog in your default Sample Gallery folder.
Recent Files	Contains a list of recently opened files.
Save Model	Saves opened model or opens the <b>Save</b> file dialog.
Save Model as	Opens the <b>Save</b> file dialog and allows you to save your model to another folder or in another file format.
Reverse Engineering   Connections	Opens the <b>Connections</b> dialog.
Reverse Engineering   Reverse Engineering Wizard	Opens the <b>Reverse Engineering Wizard</b> .
Import   Toad for Oracle Project	Displays the <b>Open</b> dialog in you default Toad Data Modeler folder.
Import   Toad for Oracle ERD	Displays the <b>Open</b> dialog in your default Toad Data Modeler folder and also opens <b>Connections</b> dialog.
Import   Case Studio 2 Model	Displays the <b>Open</b> dialog in your default Models folder. This option is used for importing model files from Case Studio 2.
Import   Import from Excel	Opens the <b>Import from Excel</b> dialog.
Import   Import from CSV	Opens the <b>Import from CSV</b> dialog.
Export   Export to Excel	Opens the <b>Export to Excel</b> dialog.
Export   Export to CSV	Opens the <b>Export to CSV</b> dialog.
Export   Export to Image	Opens the <b>Export to Image</b> dialog.
Close   Model	Closes currently selected model.
Close   All Models	Closes all opened models.
Page Setup	Opens the <b>Page Setup</b> dialog.
Print Preview	Displays a preview of your ER diagram as it will look when printed.
Print	Opens the <b>Print</b> dialog to configure printer settings.
Exit	<p>Finishes your work in Toad Data Modeler. Before closing the application your are asked to save changes you made in your models.</p> <p>This is done in the <b>Application Close</b> dialog (shown on the screenshot below), where all of your opened models and projects are listed. Select all the items that you</p>

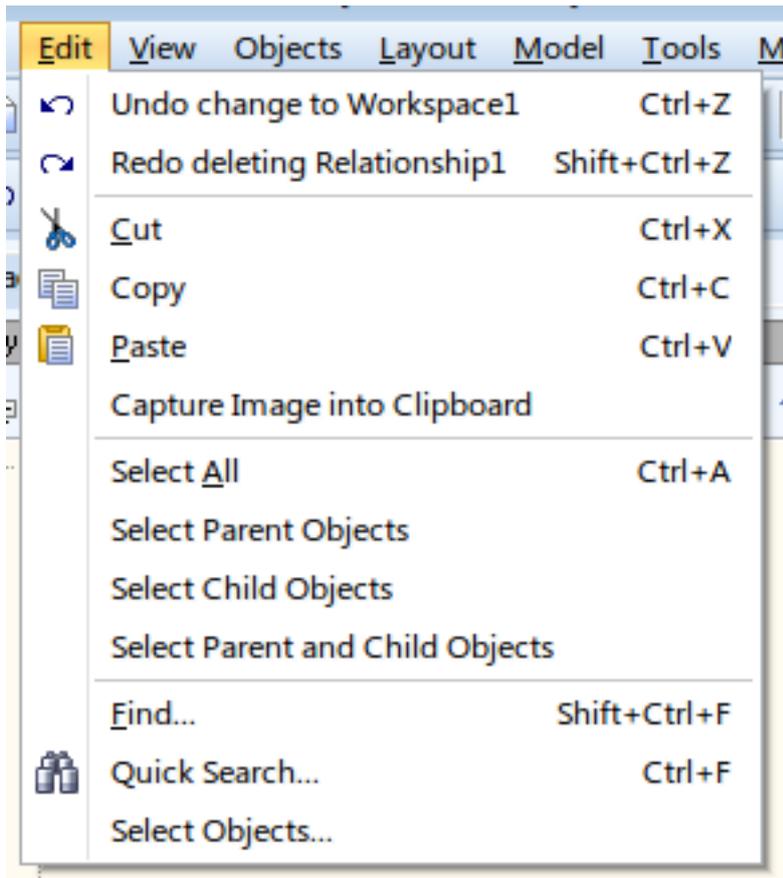
## Option

## Description

want to save, all items that have been modified are selected to be saved by default.



## Edit

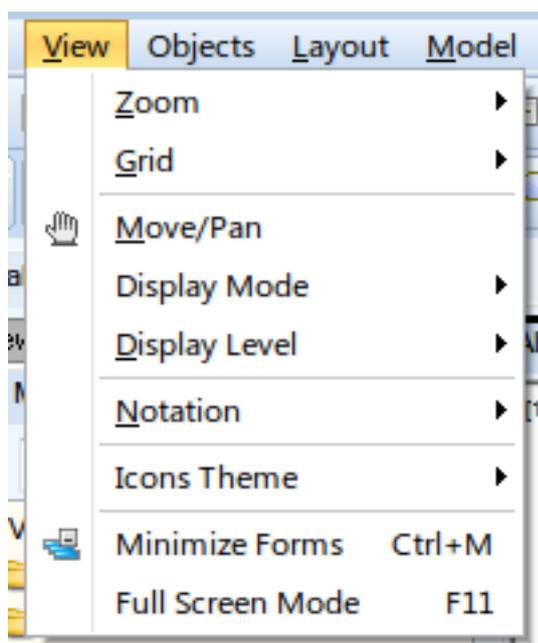


The Edit menu mostly contains basic actions such as Copy, Cut, Paste, Undo and Redo and also several options to select objects on Workspace.

Option	Description
Undo	Undoes the last action.
Redo	Redoes the last undid action.
Cut	Cuts selected object to clipboard.
Copy	Copes selected object to clipboard.
Paste	Pastes the copied/cut object from clipboard.
Capture Image into Clipboard	Choosing this option allows you to select an area on your Workspace. This area will be captured as an image and stored in your clipboard (can be pasted to somewhere else).
Select All	Selects all objects on current Workspace.
Select Parent Objects	Selects all Parent objects of the currently selected object.
Select Child Objects	Selects all Child objects of the currently selected object.

Option	Description
Select Parent and Child objects	Selects both Parent and Child objects of the currently selected object.
Find	Opens the <b>Find</b> dialog. <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p><b>i</b> TIP: Use wildcards - example: "*user" or "?ser".</p> <p>* - replaces unlimited number of characters</p> <p>? - replaces any single character</p> </div>
Quick Search	Opens <b>Quick Search</b> dialog which allows you to search through all objects in a model.
Select Objects	Allows you to select objects based on their <b>Owner</b> and <b>Category</b> .

## View



Display modes, display levels, used notation and icons are all configured in this menu.

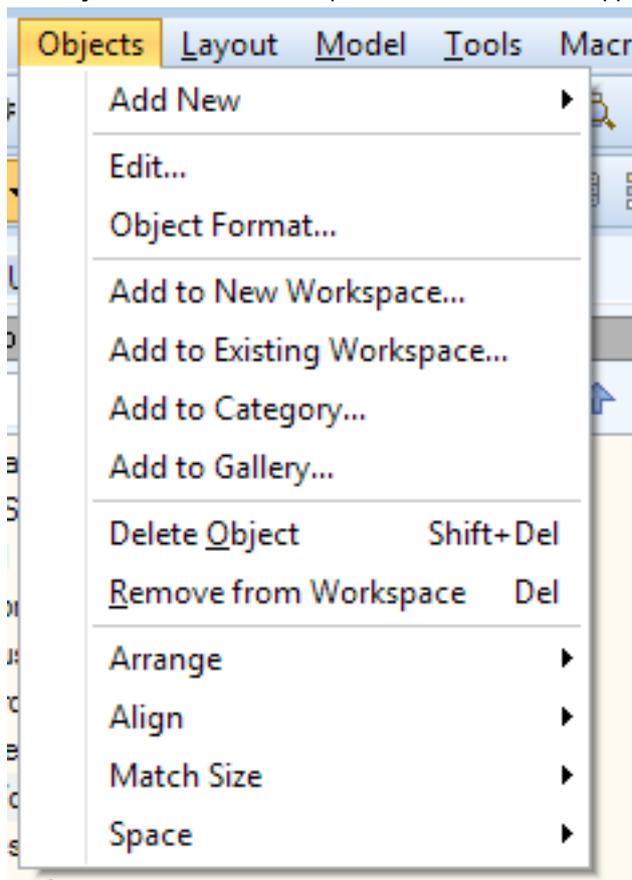
Option	Description
Zoom	Zoom options: <b>Zoom In</b> Turns your cursor into a magnifier glass that allows you to zoom in. <b>Zoom Out</b> Turns your cursor into a magnifier glass

Option	Description
	<p>that allows you to zoom out.</p> <p><b>i</b>   Note: To deactivate zoom tools, press <b>Esc</b>, or right-click the work area.</p> <p><b>Zoom Box</b> - Turns your cursor into a magnifier glass that allows you to draw a selection box. The area will be then zoomed in.</p> <p><b>Fit to Screen</b> - Sets the display level so the whole diagram can be seen completely.</p> <p>See <a href="#">Designer and Workspace</a> for more tips on navigation on Workspace.</p>
Grid	<p><b>Show Grid</b> - Shows/hides the Grid.</p> <p><b>Snap to Grid</b> - Toggles Snap to Grid function on/off. When on, it helps you to align objects while moving by snapping them to the Grid.</p> <p><b>Align Objects to Grid</b> - Aligns already existing objects to the Grid.</p> <p><b>Grid Settings</b> - Allows you to set properties of Grid.</p> <p><b>Minimal Size of Displayed Grid</b> - Determines how big the Grid must be to be displayed at all.</p> <p><b>Grid Size</b> - Sets the horizontal and vertical distance between points of Grid.</p>
Move/Pan	<p>Activated the Move/Pan tool. Drag your mouse to move around workspace. Deactivate by right-clicking or pressing ESC.</p>
Display Mode	<p>Determines which property is shown in object headers - captions/names/full names (eg. Customer/T_CUSTOMER/SCOTT.T_CUSTOMER)</p>
Display Level	<p>In Physical model sets the level of details displayed:</p> <ul style="list-style-type: none"> <li>• Entities</li> <li>• Primary Keys</li> <li>• PK and FK Keys</li> <li>• All Keys</li> <li>• Attributes</li> </ul> <p>In Logical model determines the objects you want to display:</p> <ul style="list-style-type: none"> <li>• Entities</li> <li>• Primary Identifiers</li> </ul>

Option	Description
	<ul style="list-style-type: none"> <li>• Unique Identifiers</li> <li>• Attributes</li> <li>• Descriptions</li> </ul>
Notation	<p>Allows you to switch between two types of notations used in ER Diagram:</p> <ul style="list-style-type: none"> <li>• IE</li> <li>• IDEF1X</li> </ul>
Icons Theme	Allows you to use Toad Data Modeler icons or <b>Toad for Oracle</b> icons.
Minimize Forms	Minimizes all currently opened forms and dialogs.
Full Screen Mode	Displays Toad Data Modeler in full screen mode (F11).

## Objects

The **Objects** menu is model dependent. Different items appear in different Models.

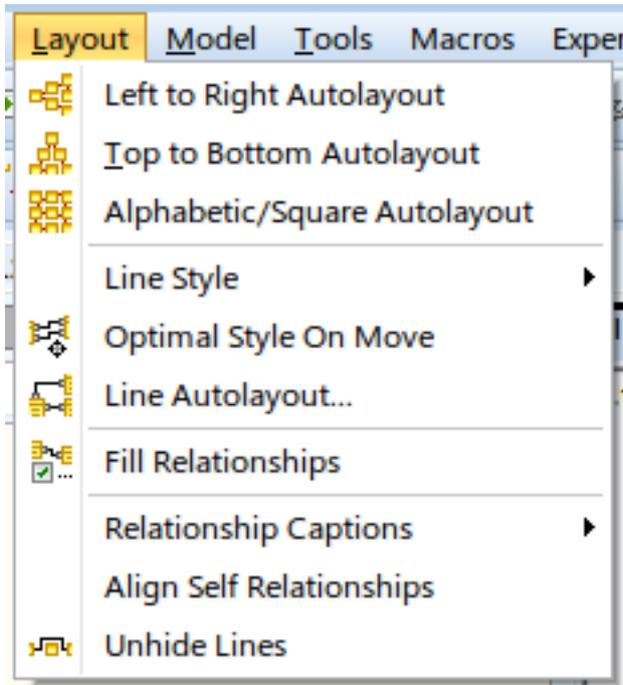


Option	Description
Add New	<p>Adds a new object to the currently selected Workspace.</p> <p><b>Database objects:</b></p> <ul style="list-style-type: none"> <li>• Entity</li> <li>• Relationship</li> <li>• Non-identifying Relationship</li> <li>• Self Relationship</li> <li>• Inheritance (logical)</li> <li>• M:N Relationship (universal, physical)</li> <li>• View (universal, physical)</li> <li>• View Relationship (universal, physical)</li> <li>• Materialized View (physical)</li> </ul> <p><b>Diagram objects:</b></p> <ul style="list-style-type: none"> <li>• Note</li> <li>• Line</li> <li>• Stamp</li> <li>• Categories</li> <li>• Image</li> <li>• Rectangle</li> <li>• Ellipse</li> <li>• Text (universal, physical)</li> <li>• Label</li> <li>• Label Quadrangle</li> <li>• Label Ellipse</li> </ul> <p><b>i</b> Note: If not followed by parentheses containing Model type, the object is available in all Models.</p>
Edit	Edits currently selected object.
Object Format	Opens <b>Object Format</b> dialog of the currently selected object.
Add to New Workspace	Creates new Workspace and adds the selected object to it.
Add to Existing Workspace	Adds object to an already existing Workspace.
Add to Category	Adds object to new or existing Category.
Add to Gallery	Adds object to new or existing Gallery.

Option	Description
Delete Object	Deletes selected object from Model.
Remove Object from Workspace	Removes selected object from Workspace. <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p><b>i</b>   Note: The object still remains in the Model, only its graphical representation is removed.</p> </div>
Arrange	Contains various options for arranging objects into layers: <ul style="list-style-type: none"> <li>• <b>Bring to front</b> - brings the object to the top layer.</li> <li>• <b>Bring forward</b> - brings the object one layer up.</li> <li>• <b>Send backward</b> - sends the object one layer down.</li> <li>• <b>Send to back</b> - sends the object to the bottom layer.</li> <li>• <b>Setting</b> - opens the <b>Object Format</b> dialog where you can specify <b>Z-Order</b> - number which determines the layer arrangement of objects on workspace (objects with higher Z-Order are shown on top of objects with lower Z-Order)</li> </ul>
Align	Aligns multiple selected objects: <ul style="list-style-type: none"> <li>• Top</li> <li>• Left</li> <li>• Right</li> <li>• Bottom</li> <li>• Horizontal Center</li> <li>• Vertical Center</li> </ul>
Match size	Matches sizes of multiple selected objects: <ul style="list-style-type: none"> <li>• Width</li> <li>• Height</li> <li>• Width and Height</li> </ul>
Space	Offsets selected objects by the same amount: <ul style="list-style-type: none"> <li>• Vertical Equally</li> <li>• Horizontal Equally</li> <li>• Anchor Points - Offsets anchor points of an object by the same amount)</li> </ul>

# Layout

Layout Menu contains various tools to organize your objects on Workspace.

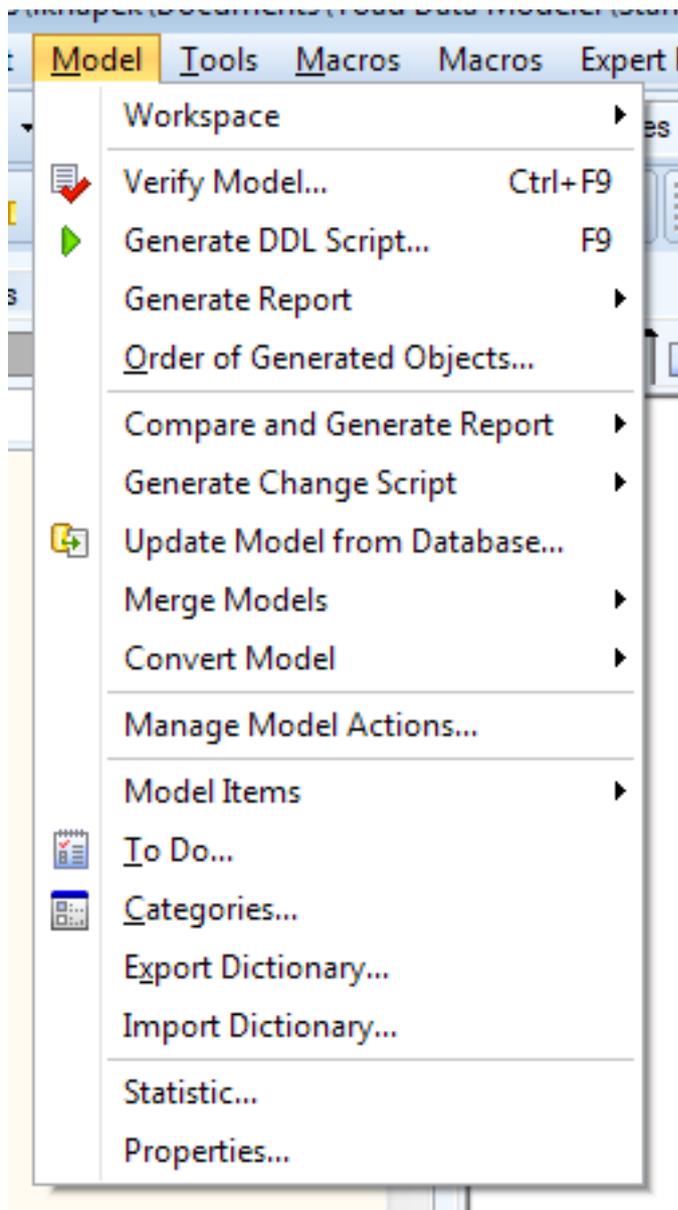


Option	Description
Left to Right Autolayout	Automatically organizes all objects from left to right hierarchically.
Top to Bottom Autolayout	Automatically organizes all objects from top to bottom hierarchically.
Alphabetic/Square Autolayout	Automatically organizes all objects to square shape and sorts them alphabetically.
<p><b>i</b> TIP: Not selecting any objects will organize all objects on Workspace. You can select multiple objects before choosing Autolayout style. In that case a prompt will show up and you will be asked to choose the area where the selected objects should be organized. Draw a rectangle and the objects will move into the drawn area. The other way is to simply click, which defines the upper left corner of autolayout area.</p>	
Line Style	<p>Changes relationship and note lines into shapes of letters:</p> <ul style="list-style-type: none"> <li>• Optimal Style</li> <li>• U Style</li> <li>• A Style</li> <li>• C Style</li> <li>• D Style</li> </ul>

Option	Description
	<ul style="list-style-type: none"> <li>• Vertical Style</li> <li>• Horizontal Style</li> <li>• Z Style</li> <li>• L Style</li> </ul>
Optimal Style On Move	As you move the object shapes on Workspace, all lines will automatically change their style to the most optimal one. This function can change the number of break points on lines.
Line Autolayout	Automatically tries to reorganize lines on Workspace to the most optimal variant.
Fill Relationships	Adds missing relationships between entities in Workspace. The relationships already have to be a part of the Model, this function does not create new ones.
Relationship Captions	Moves or hides the relationship captions: <ul style="list-style-type: none"> <li>• Move to Parent</li> <li>• Move to Child</li> <li>• Move to Center</li> <li>• Hide</li> </ul>
Align Self-Relationships	Resets Self-Relationship lines into their default position.
Unhide Lines	Displays lines hidden behind object shapes.

## Model Menu

The options in the **Model** menu are model dependent. From here, key features such as Verify model, generate SQL script, generate report, synchronize model etc. can be accessed.



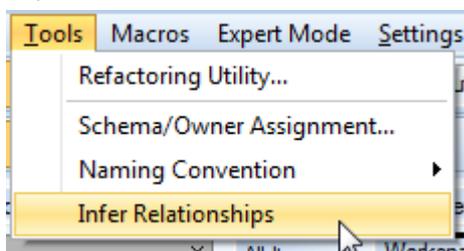
Option	Description
Workspace   New Workspace	Creates a new Workspace, prompting you to name it and select objects it should contain.
Workspace   Edit Workspace	Allows you to edit active Workspace properties such as name and description. You can also select objects that will be shown or hidden in the Workspace.
Workspace   Delete Workspace	Deletes Workspace selected from list of currently opened Workspaces.
Workspace   Workspace Format	Contains mostly graphical settings that can be changed to your liking.
Verify Model	Displays <b>Model Verification</b> dialog which allows you

Option	Description
	to quickly check for errors in your model and fix them.
Generate DDL Script	Allows you to generate SQL code which can be run to create a database identical to your Toad Data Modeler model.
Generate Report   Report Wizard	Opens <b>Report Wizard</b> , allowing you to generate customizable HTML, RTF or PDF reports
Generate Report   Reports / XSL Transformation	Allows you to generate reports using customized XSL templates.
Order of Generated Objects	The generation order of objects during DDL script generation can be changed here.
Compare and Generate Report	Opens <b>Model Compare</b> Wizard, showing you the differences between two selected models. You can generate a report by clicking <b>Report</b> button in <b>Compare Tree</b> dialog.
Generate Change Script	Opens <b>Generate Change Script</b> Wizard which compares two different models (or a model and a database) and generates a SQL script reflecting all the changes you made in the first model to your target.
Update Model From Database	Compares local model with target database and changes the model to match the database.
Merge Models   Run	Opens <b>Merge Model</b> Wizard which is able to merge two selected models into one.
Merge Models   Simple Merge	Merges two selected models, bypassing a lot of options in <b>Merge Model</b> Wizard.
Convert Model   Run	Opens <b>Convert Model</b> Wizard which converts selected model to another database platform or version.
Convert Model   Simple Conversion	Converts selected model to another database platform or version, bypassing a lot of options in the <b>Convert Model</b> Wizard.
Manage Model Actions	Opens <b>Model Actions</b> tool which contains key model functions.
Model Items	Allows you to see all objects in selected object group in and add, edit or delete.
To Do	You can create your own tasks and assign them priority and deadline. These are only for organizing your work and they do not have impact on the model itself.
Categories	This option allows you to create, edit and delete categories, which are used to graphically organize objects in your model. Each category has a certain color and objects which are part of a category will share its color (e.g. entity headers).
Export Dictionary	Exports dictionary types, user data types and

Option	Description
	domains into <b>.txl</b> file.
Import Dictionary	Imports dictionary types, user data types and domains from <b>.txl</b> file.
Statistic	Displays information and various statistics about your model and workspaces.
Properties	Opens the <b>Model Properties</b> dialog which contains information about the model itself, its description and statistics.

## Tools Menu

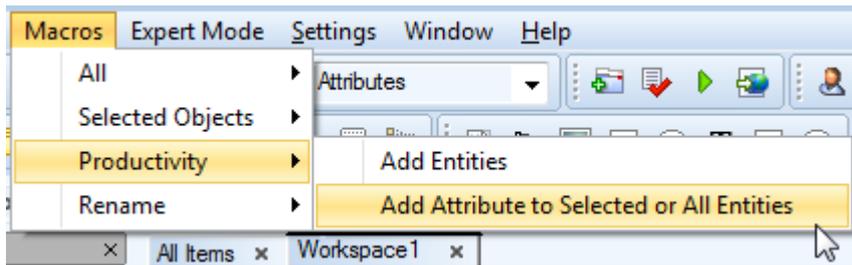
Tools Menu is not shown in Logical Models since it only contains options applicable to Universal and Physical Models.



Option	Description
Refactoring Utility	If you rename object in your model, the Refactoring Utility can be used for changing the old object name in SQL properties of other objects. <b>Example:</b> Rename an attribute and use the Refactoring Utility to replace the old name with new name in SQL properties of database views, triggers, stored procedures etc.
Schema/Owner Assignment	Allows you to assign multiple objects to a Schema/Owner/Database at once.
Naming Conventions	Manage rules and naming standards for object groups. Also contains an option to verify current names.
Infer Relationships	Creates relationships between identically named Primary or Alternate Key attributes, if they don't exist already.

## Macros Menu

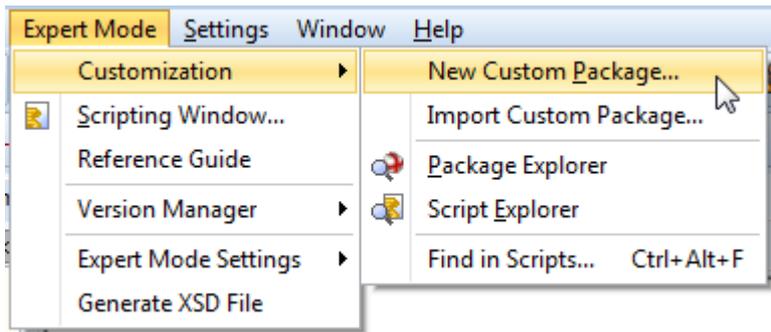
The Macros menu contains items that can be customized by users or new features developed by users.



Option	Description
All	<p>Displays sample macros that you can use for your currently active model.</p> <p>The selected macro will be applied to all objects on all Workspaces of the model.</p> <p>Sample macros:</p> <ul style="list-style-type: none"> <li>• <i>Convert Names to Lower Case</i></li> <li>• <i>Convert Names to Upper Case</i></li> <li>• <i>Alphabetic Autolayout—Autolayout</i></li> </ul>
Selected Objects	<p>Displays sample macros that you can use for your currently active model.</p> <p>The selected macro will be applied only to selected objects on the currently active Workspace.</p> <p>Sample macros:</p> <ul style="list-style-type: none"> <li>• <i>Remove Spaces from Names</i></li> <li>• <i>Display Entities Note on Workspace</i></li> </ul>
Productivity	<p>Displays macros which allow you to do multiple actions at once, enhancing your productivity.</p> <ul style="list-style-type: none"> <li>• <i>Add Entities</i>— This macro allows you to add multiple entities in your model at the time.</li> <li>• <i>Add Attribute to Selected or All Entities</i>— Fast and easy way to add new attributes to entities.</li> </ul>
Rename	<p>Displays macros which serve to rename multiple objects at once.</p>

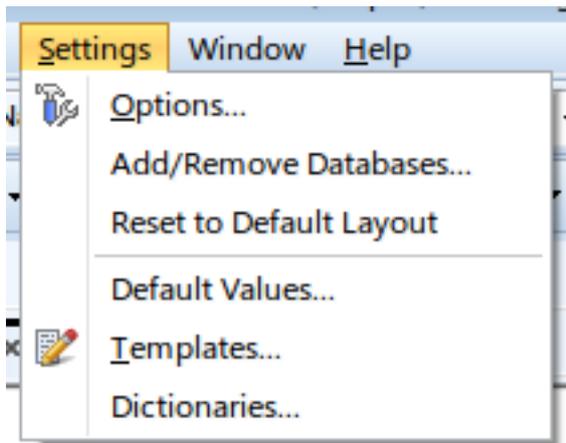
## Expert Mode Menu

The Expert Mode Menu appears only if you enabled Expert Mode in **Settings | Options | General**.



Option	Description
Customization   New Custom Package	Allows you to create a new custom Package.
Customization   Import Custom Package	Imports Packages or Metamodels saved as .txg and .txm files.
Customization   Package Explorer	Opens an instance of Package Explorer window.
Customization   Script Explorer	Opens an instance of Script Explorer window.
Customization   Find in Scripts	Allows you to quickly search through all system and user scripts.
Scripting Window	Opens an instance of Scripting Window.
Reference Guide	Opens the Reference Guide.
Version Manager   Internal Version Manager	Allows you to use TDM integrated Version Manager.
Version Manager   Add to Version Manager	Adds currently active model to TDM integrated Version Manager project.
Expert Mode Settings   Data Type Conversion Settings	Here you can affect how Toad Data Modeler converts various data types during Model Conversion to another database platform or version.
Expert Mode Settings   Available OTPs	Advanced settings that can be set for: <ul style="list-style-type: none"> <li>• Model Definition - specify model structure depending on database platform</li> <li>• Other Model Features - define structure of selection trees in certain dialogs</li> </ul>
Generate XSD File	Generates an XSD file which contains the structure definition of Toad Data Modeler XML model files(*.txp, *.txl).

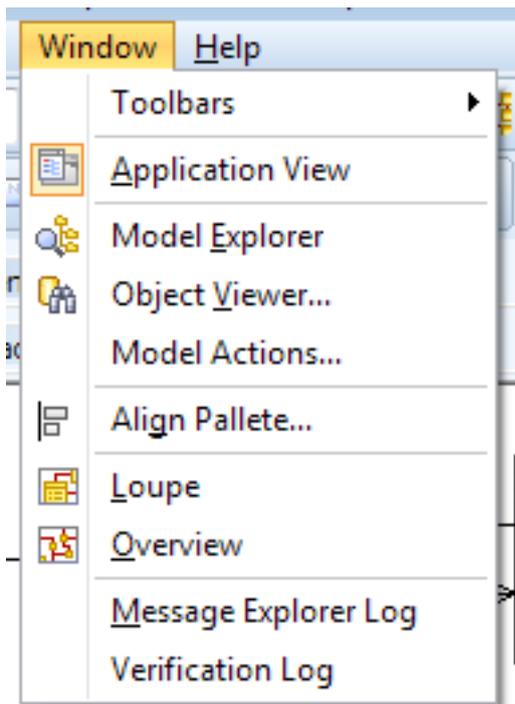
# Settings Menu



Option	Description
Options	Displays the Options dialog, where most of the Toad Data Modeler settings are located.
Add/Remove Databases	Opens a dialog where you can enable/disable installed databases.
Reset to Default Layout	Resets the application layout to default, requires restart.
Default Values	Allows you to set various default properties (e.g. Names, Captions, Values).
Templates	Displays the Template Editor where you can add or edit your own templates. These can be used when editing SQL code of some objects.
Dictionaries	Contains terms used in generated reports. Feel free to add your own new terms, export/import dictionaries or translate them to another language.

# Window Menu

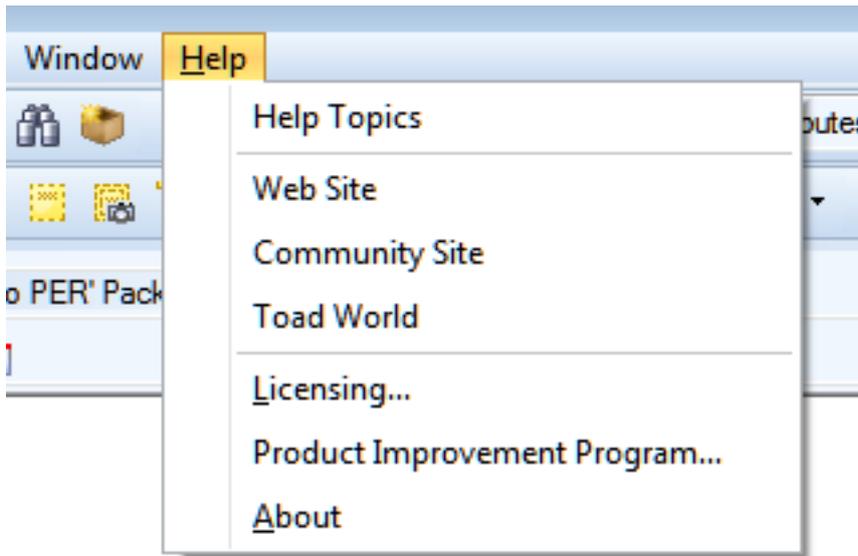
Window Menu is a basic access point for displaying dockable windows, panes and toolbars.



Option	Description
Toolbars	Used to show/hide all available toolbars.
Application View	Shows/hides Application View window.
Model Explorer	Opens an instance of Model Explorer window.
Object Viewer	Opens an instance of Object Viewer window.
Model Actions	Opens Model Actions window.
Align Palette	Opens Align Palette where you can click buttons to align selected objects on workspace.
Loupe	Shows/hides Loupe window.
Overview	Shows/hides Overview window.
Message Explorer Log	Opens an instance of Message Explorer window.
Verification Log	Opens an instance of Verification Log window.

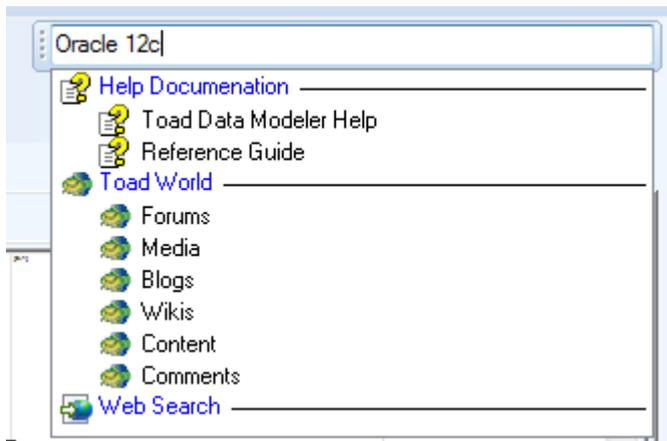
## Help Menu

Through the Help menu you can access many helpful resources (such as product documentation, Toad Web Site, community page on [ToadWorld.com](http://ToadWorld.com)...) and manage Licenses for Toad Data Modeler.



## Search Bar

Using Search Bar you can easily search for a term in multiple destinations, such as Help, ToadWorld.com and the Internet.



## Toolbars

The following toolbars are available in Toad Data Modeler:

- [Main Toolbar](#)
- [Window Toolbar](#)
- [Views Toolbar](#)

- Model Toolbar
- Display Toolbar
- Users Toolbar
- Naming Conventions Toolbar
- Grid Toolbar
- Undo/Redo Toolbar
- Model Objects Toolbar
- Graphics Objects Toolbar
- Zoom Toolbar
- Colors and Alignment Toolbar
- Layout Toolbar
- Expert Mode Toolbar
- Scripting Window
- Alignment Toolbar
- Help Search
- 
- Style

## Main Toolbar



Option	Description
New Model	<p>Opens the <b>New Model</b> dialog.</p> <p>Supported models:</p> <ul style="list-style-type: none"> <li>• Physical Data Model</li> <li>• Universal Data Model</li> <li>• Logical Data Model</li> </ul> <p><b>i</b> TIP: Right-click the dialog to select the display of the options (Large Icons, Small Icons, List).</p> <p><b>Model Name</b> - Define a name of your model. (Also, you can change the name later in the Application View or Model Explorer (press F2).)</p>

Option	Description
	<p><b>Notes:</b></p> <p> Note:</p> <ul style="list-style-type: none"> <li>• <b>Database Name</b> - A database for which the model has been created (e.g. Oracle 10g). The database name information can be found at the bottom of the Application Window, or it is displayed in the pop-up hint when you point your mouse cursor at the Model Name in the Application View or Model Explorer.</li> <li>• <b>Model Name</b> - In Toad Data Modeler, Model Name should be understood as a title of a document that is saved.</li> <li>• <b>File Name</b> - A name of file under which the model is saved. File Name is defined after you select <b>Save Model</b> or <b>Save Model as</b>.</li> </ul>
Open Model	Displays the <b>Open</b> dialog in your default Models folder.
Save Model	Saves opened model or opens the <b>Save</b> file dialog.
Connections	Opens the <b>Connections</b> dialog.
Reverse Engineering	Opens the <b>Reverse Engineering Wizard</b> .
Run Compare	Opens <b>Model Compare</b> Wizard, showing you the differences between two selected models. You can generate a report by clicking <b>Report</b> button in Compare Tree dialog.
Run Generate Change Script	Opens <b>Generate Change Script</b> Wizard which compares two different models (or a model and a database) and generates a SQL script reflecting all the changes you made in the first model to your target.
Update Model	Compares local model with target database and changes the model so it's the same as the database.
Run Merge	Opens <b>Merge Model</b> Wizard which is able to merge two selected models into one.
Run Convert	Opens <b>Convert Model</b> Wizard which converts selected model to another database platform or version.
Print	Opens the <b>Print</b> dialog to configure printer settings.
Print Preview	Shows preview of the model as it will be printed.

Option	Description
Options	Displays the Options dialog, where most of the Toad Data Modeler settings are located.

## Window Toolbar



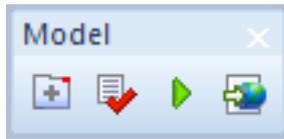
Option	Description
Application View	Toggles <b>Application View</b> on/off.
Physical Model Explorer	Opens an instance of <b>Physical Model Explorer</b> .
Object Viewer	Opens an instance of <b>Object Viewer</b> .
Quick Search	Displays <b>Quick Search</b> Window.
New Gallery	Creates a new <b>Gallery</b> and displays <b>Gallery Edit</b> , where you can edit its properties.

## Views Toolbar



Option	Description
Move	Turns your cursor into a hand allowing you to move around easily by dragging on the workspace
Loupe	Toggles Loupe on/off. Use it to see the area around cursor more clearly.
Overview	Toggles Overview on/off. Useful for seeing the whole diagram when working with large models.
Minimize All Undocked Forms	Minimizes all undocked windows and forms so they don't block your view.

# Model Toolbar



Option	Description
New Workspace	Creates a new Workspace.
Verify Model	Displays <b>Model Verification</b> window, where you can select the items you want to verify and change verification settings.
Generate DDL Script	Displays DDL Script Generation window, where you can select the items you want to generate and change generation settings.
Report	Opens <b>Report Wizard</b> , a tool that you can use to generate HTML, RTF and PDF reports.

# Display Toolbar



Option	Description
	<p>Determines what property should be displayed in object headers and workspace properties.</p> <ul style="list-style-type: none"> <li>• Captions - The label of an object</li> <li>• Names - The identification of an object used in database</li> <li>• Full Names - Shows the Schema/Owner/User of the object followed by object name</li> </ul>
	<p>The level of detail shown in entities. Selecting an option will cause all options above it to be shown as well.</p> <ul style="list-style-type: none"> <li>• Entities</li> <li>• Primary Keys</li> <li>• PK and FK Keys</li> <li>• All Keys</li> </ul>

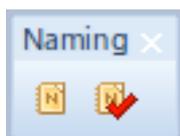
Option	Description
	<ul style="list-style-type: none"> <li>Attributes</li> </ul>

## Users Toolbar



Option	Description
Users	Displays <b>Users</b> dialog where you can manage users and their memberships in groups.
User Groups	Displays <b>User Groups</b> dialog where you can manage user groups and their members.

## Naming Conventions Toolbar



Option	Description
Set up Naming Conventions to Model	Displays dialog in which you can link existing Naming Convention to models, or create new Naming Conventions.
Naming Convention Verification and Synchronization	Verifies if the model meets the criteria of currently linked Naming Convention. Displays an overview of items, whose names are invalid.

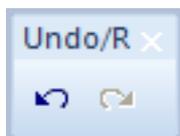
## Grid Toolbar



Option	Description
Snap to Objects	Toggles Snap to Objects function on/off. When on, shows guidelines when you move an object to help

Option	Description
	you align it better.
Snap to Grid	Toggles Snap to Grid function on/off. When on, it helps you to align objects while moving by snapping them to the Grid.
Show Grid	Shows/hides the Grid.
Grid settings	Opens Grid Settings. Grid size and minimum size is configured here.

## Undo/Redo Toolbar



Option	Description
Undo	Reverses the last action.
Redo	Redoes the last undone action.

## Model Objects Toolbar



Option	Description
<b>Select Tool</b>	<p>Defines what object types you can select by creating a selection box with your mouse. Default is Select All. Other options are:</p> <ul style="list-style-type: none"> <li>• Select Entities</li> <li>• Select Views</li> <li>• Select Materialized Views</li> <li>• Select Relations</li> <li>• Select View Relations</li> <li>• Select All Shapes</li> <li>• Select All Lines</li> </ul>

Option	Description
Create	<p>The remaining buttons in the toolbar create specific objects. These are:</p> <ul style="list-style-type: none"> <li>• Non-identifying relationships</li> <li>• Identifying relationships</li> <li>• M:N relationships</li> <li>• Self relationships</li> <li>• Views</li> <li>• Materialized Views (available only in supported databases)</li> <li>• View Relationships</li> <li>• Categories</li> <li>• Stamps</li> </ul>

## Graphics Objects Toolbar



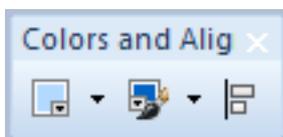
Option	Description
Create...	<p>The buttons in this toolbar create specific graphical objects:</p> <ul style="list-style-type: none"> <li>• Notes</li> <li>• Lines</li> <li>• Images</li> <li>• Rectangles</li> <li>• Ellipses</li> <li>• Text fields</li> <li>• Label Quadrangles</li> <li>• Label Ellipses</li> </ul>

## Zoom Toolbar



Option	Description
Zoom Box	Turns your cursor into a magnifying glass that allows you to draw a selection box. The area will be then zoomed in.
Zoom Out	Turns your cursor into a magnifying glass that allows you to zoom out.
Zoom In	Turns your cursor into a magnifying glass that allows you to zoom in.
Fit to Screen	Sets the display level so the whole diagram can be seen completely.
	Sets the display level of the workspace. You can either enter a number, use the arrows or choose a preset by clicking the little arrow on the right.

## Colors and Alignment Toolbar



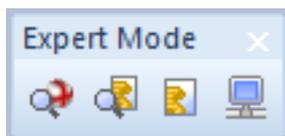
Option	Description
Brush Color	Sets the main color of selected objects that are not already part of a category.
Pen Color	Sets the border color of selected objects.
Align	Displays/hides <b>Alignment Toolbar</b> .

## Layout Toolbar



Option	Description
Autolayout	Automatically organizes selected objects in one of these three ways: <ul style="list-style-type: none"> <li>• Top to Bottom</li> <li>• Left to Right</li> <li>• Alphabetic/Square</li> </ul>
Line Autolayout	Automatically tries to reorganize lines on Workspace to the most optimal variant.
Fill Relationships to the Workspace	Adds missing relationships between entities in workspace. The relationships already have to be a part of the model, this function does not create new ones.
Hide Captions	Hides relationship captions.
Optimal Style On Move	As you move the object shapes on Workspace, all lines will automatically change their style to the most optimal one. This function can change the number of break points on lines.

## Expert Mode Toolbar



Option	Description
Package Explorer	Opens an instance of <b>Package Explorer</b> .
Script Explorer	Opens an instance of <b>Script Explorer</b> .
Scripting Window	Opens <b>Scripting Window</b> for writing scripts.
Internal Version Manager	Opens <b>Version Manager</b> dialog which contains internal version manager.

## Scripting Window



Option	Description
Show Windows Automatically	Displays a corresponding side tab. When you are writing a script, Code Explorer is displayed. When a

Option	Description
	script is being executed, Log is displayed.
Show Log	Displays a log window that shows log messages and errors related to <b>Scripting Window</b> .
Show Code Explorer	Displays a side tab that lists code segments.
Execute Script	Executes a script in <b>Scripting Window</b> .
Stop Script	Stops a running script.
Type	Switch between: <ul style="list-style-type: none"> <li>• JScript</li> <li>• VBScript</li> <li>• Internal Script</li> </ul>
Load Script from File	Load a script from a file.
Save Script	Saves a script.
Save Script as	Saves a script under a new name.

## Alignment Toolbar



Option	Description
<b>Align</b>	Aligns selected objects: <ul style="list-style-type: none"> <li>• Left</li> <li>• Horizontal Center</li> <li>• Right</li> <li>• Top</li> <li>• Vertical Center</li> <li>• Bottom</li> </ul>
<b>Match Size</b>	Matches selected objects: <ul style="list-style-type: none"> <li>• Width</li> <li>• Height</li> <li>• Both</li> </ul>
<b>Space Equally</b>	Offsets selected objects by the same distance: <ul style="list-style-type: none"> <li>• Vertical</li> </ul>

Option	Description
	<ul style="list-style-type: none"> <li>Horizontal</li> </ul>
<b>Same Space Between Anchor Points</b>	<p>Offsets anchor points of selected objects by the same distance:</p> <ul style="list-style-type: none"> <li>On Left Edge</li> <li>On Right Edge</li> <li>On Top Edge</li> <li>On Bottom Edge</li> <li>On All Edges</li> </ul>
<b>Line Style</b>	<p>Changes the selected lines shape:</p> <ul style="list-style-type: none"> <li>Letter styles (A, U, C, D)</li> <li>Straight line styles (horizontal, vertical)</li> <li>Z line styles (horizontal, vertical)</li> <li>L line styles (left, right, top, bottom)</li> <li>Line style Optimal - Toad Data Modeler automatically tries to change the shape of the line to the best possible variant. This function does not change the number of lines break points.</li> </ul>
Unhide Lines	Changes shape of all lines hidden under object shapes, making them visible.
Align Self Relationships	Resets Self Relationship lines to their default position.
Insert Break Point	Click once to activate this function. Select a line and then click anywhere on the line to create a new break point. Finish the action by moving one of the parts of the line divided by the newly added break point.

## Help Search



This tool allows you to search in multiple locations to find the information you need. Simply enter a query and click on the location you wish to search. These locations are:

**Toad® Data Modeler Documentation**

- Help
- Reference Guide

#### Toad World

- Forums
- Media
- Blogs
- Wikis
- Content
- Comments

#### Web search

## Intelligence Central Toolbar



Option	Description
Intelligence Central	Shows/hides <b>Intelligence Central</b> window.
Download File from Intelligence Central	Displays <b>Pull Model File</b> dialog, where you can download a model from Toad Intelligence Central server. The model will be then immediately opened.
Publish to Intelligence Central	Displays a dialog where you can publish your current model to Toad Intelligence Central server. See <a href="#">Publishing Models/Reports</a> for more information.
Notifications	Displays <b>Notifications</b> log. Notifications are fetched from TIC server, their purpose is to notify users about changes to objects made by other users.

## Style

Select your preferred visual style from the popup menu.

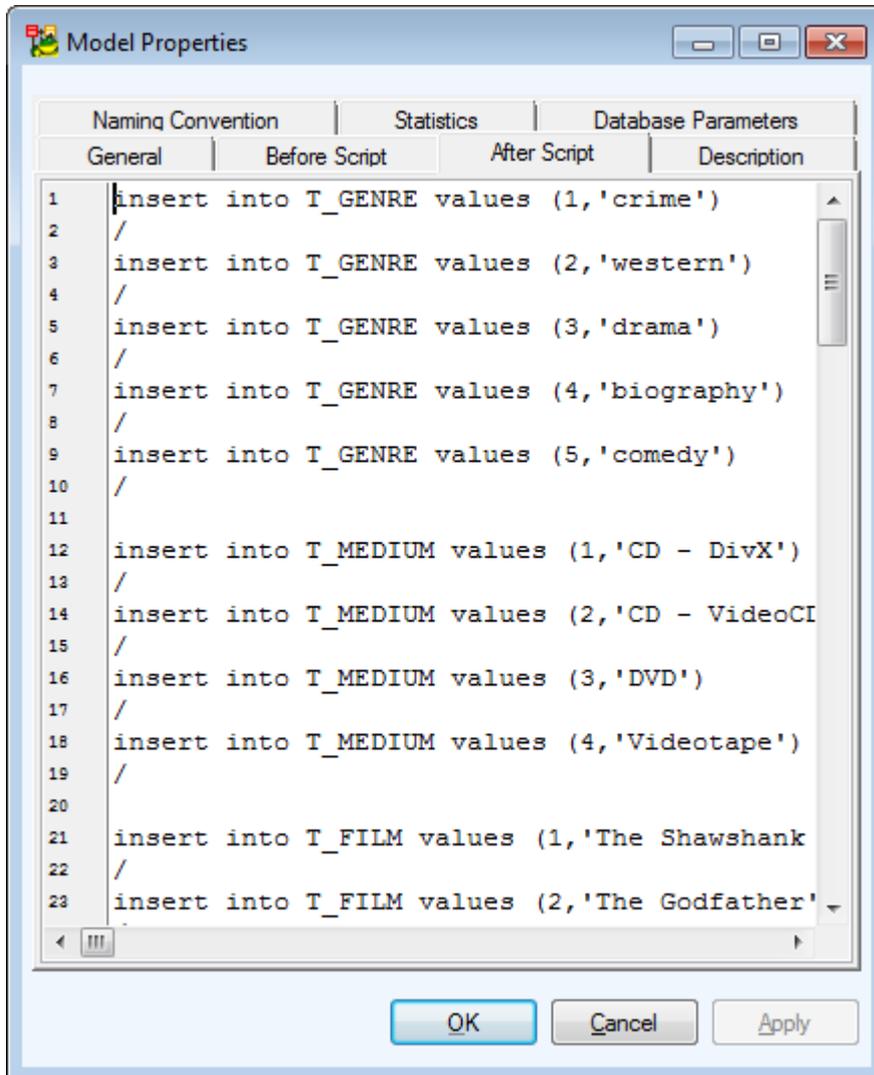
# Models and Model Objects

There are three types of models in Toad Data Modeler:

- [About Physical Data Modeling](#)
- [About Logical Data Modeling](#)
- and [About Universal Data Model](#)

# Model Properties

Select **Model | Properties**.



Tab	Description
General	General information on the model (project, model, author, company, version etc.)
Before Script	Whatever you write on the Before Script tab, it will be generated before the generated SQL code.
After Script	Whatever you write on the After Script tab, it will be generated after the generated SQL code.

Tab	Description
Description	Notes on the model
Naming Convention	Here you can select a naming convention for the model.
Database Parameters	<p>The value in the <b>Length Semantics</b> box shows which default length semantics (BYTE or CHAR) was set in the database for reverse engineering of Oracle database.</p> <p> Note: This tab is available only in Oracle models.</p>
Naming Convention	<p>Here you can link or unlink a naming convention to/from your model.</p> <p><b>Manage Naming Conventions</b> - Opens the <b>Naming Convention</b> dialog (available also from the <b>Tools</b> menu)</p> <p><b>Edit Naming Convention</b> - Opens the <b>Naming Convention Properties</b> dialog.</p>

# Model Statistics

**To display details on your model, its Workspaces, objects etc.**

Select **Model | Statistic....**

**To change some of the General Information (e.g. model name, author or company)**

Select **Model | Properties....**

**i** TIP: You can also change it via the shortcut of Stamp on the Workspace - just double-click the Stamp.

## About Physical Data Modeling

Toad Data Modeler allows you to design **Entity Relationship Diagrams** of specific database platforms, convert physical model from one database platform to another, create an ER Diagram directly from your database (Reverse Engineering feature), update physical models, generate DDL/SQL scripts and Change Scripts, create Dictionary Types, Views, Triggers, Functions, generate detailed documentation to your model (in HTML, RTF, PDF, XSLT formats) and much more.

This chapter describes features and functions related to Physical Data Modeling. Look around each section to get the information you need.

**i** Note: See the sample physical model *Videorental* (Oracle 10g db) that is included in the installation package of Toad Data Modeler. Default location is: C:\Program Files (x86)\Quest Software\Toad Data Modeler 8.0\Samples.

## Benefits of Physical Data Model

- Detailed definition of database structure, including database specific items, for example:
  - Stored procedures
  - Functions
  - Triggers
  - Views
  - Materialized views
  - Sequences (auto increments) etc.
- Possibility to synchronize local model with existing database.
- Possibility to specify logical names for objects (captions for tables, attributes and other objects).
- Detailed database specific information can be exported to HTML/RTF/PDF or XML/XHTML/CSV reports.
- Automatic generation of SQL code for selected objects (SQL code generation is not available in Logical and Universal Model)
- Automatic migration of PK attributes to child entities (Attributes don't migrate to child entities in Logical Model)

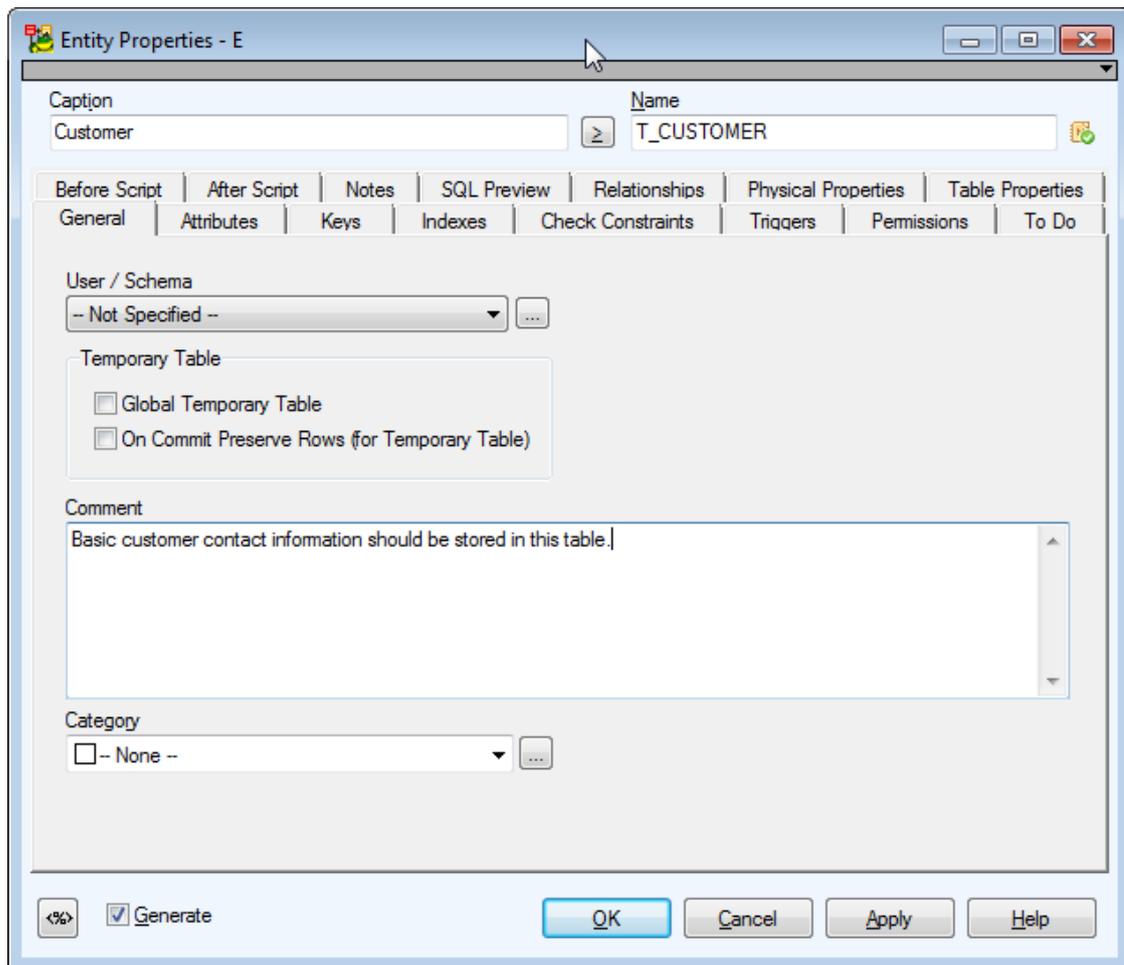
## Create Entities

In Toad Data Modeler, there are several ways how to create entities - on the Workspace, via **Model Explorer** and in the **Entities** dialog (**Model | Model Items | Entities**). Create entities directly on the Workspace.

### Scenario

Create entity *Customer* on the Workspace in your *Videorental* model.

1. Click  on the toolbar (also CTRL+E) and then click anywhere on the Workspace.
2. Double-click the entity to edit it.
3. Define the entity caption and name



Caption Logical entity name - *Customer*.

Name Physical entity name - *T\_CUSTOMER*.

4. Define other properties on tab **General** and other tabs (e.g. **Notes**, **Comments** etc.). To save the changes simultaneously and leave the form open, click **Apply**.

## Edit Entities

To edit entities:

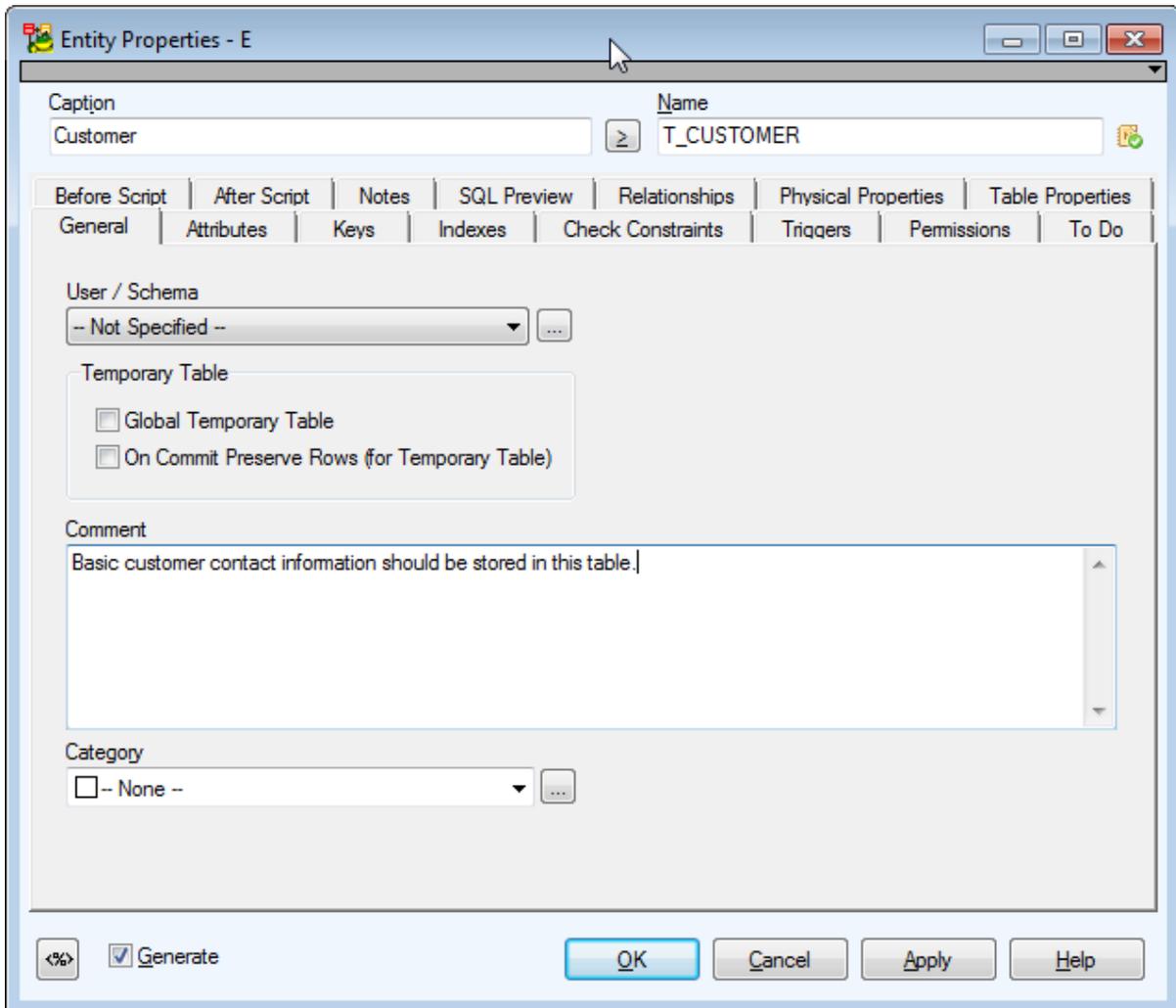
- Double-click the entity on the Workspace.

or

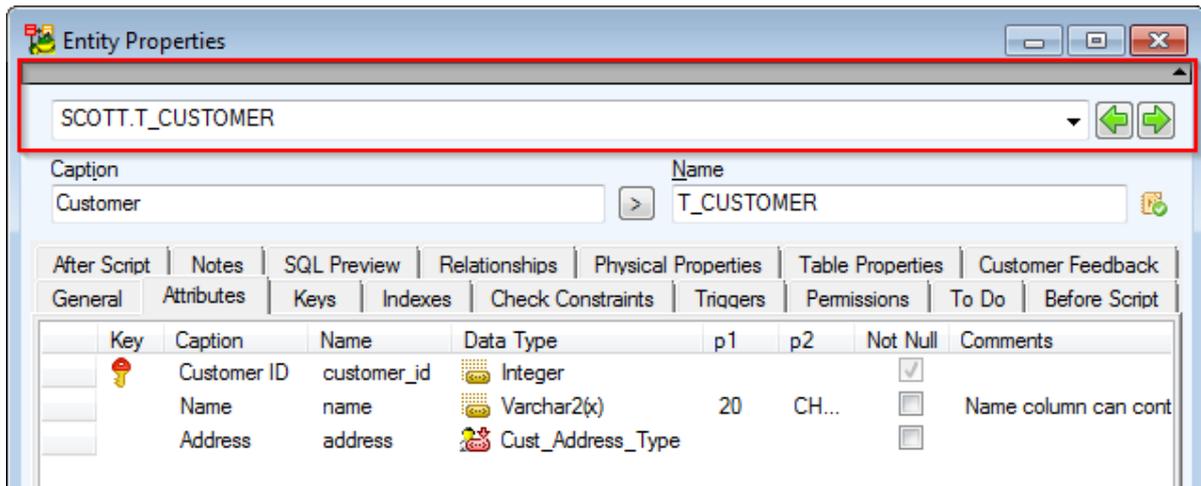
- Edit the entity in **Model Explorer | Entities** folder | double-click the selected entity (or right-click | **Edit**).

**i** Note: You can also edit entities in the **Entities** dialog (**Model Menu | Model Items | Entities | Edit**).

**Example:** The **Entity Properties** form (Oracle 10g db). Object navigator collapsed.



Example of tab **Attributes**, Object Navigator expanded.



### Option

### Description

Object Navigator  
Dropdown Menu

All entities in your model are listed here. It allows you to edit entities quickly and comfortably from one place.

**Tip:** After you finish editing an entity, click **Apply** to confirm changes and select another entity from the Object Navigator box.



Buttons **Previous** and **Next** for quick navigation among entities.

### General Tab

General properties of entity

Caption

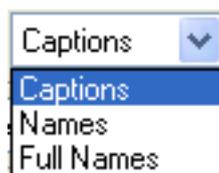
Logical name of entity

Name

Physical name of entity

[About Naming Conventions](#)

**Tip:** You can choose if you want to display objects physical names, logical names, or full names by selecting an option from dropdown menu on **Display Toolbar**.



Schema

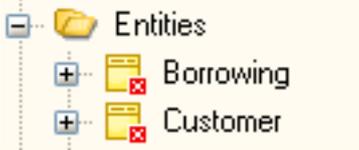
Schema selection box. Click the small button on the right to access the **Schema** dialog.  
(User/Schema stands only for Oracle models.)

Category

Category selection box. To see a list of Categories of your model, click the small button on the right.

Generate

Select it to generate the entity in final SQL (DDL) script. (It is selected by default.)

Option	Description
	<p><b>Note:</b> Objects of your model that have the <b>Generate</b> checkbox disabled in their <b>Properties</b> dialogs are displayed in Model Explorer this way:</p> 
<p><b>i</b></p>	<p>Note: Other options on the <b>General</b> tab vary depending on a target database. See the specific options for your database in the "Databases" chapter.</p>
<p><b>Attributes, Keys, Indexes, Check Constraints, Triggers, Permissions Tabs</b></p>	<p>On these tabs, you can add, edit and delete particular items. More details on each of them can be found in appropriate topics that follow.</p>
<p><b>To Do Tab</b></p>	<p>On this tab, you can write some tasks on the selected entity. To see all To Do tasks, select <b>Model   To Do</b>.</p>
<p><b>Before Script Tab</b></p>	<p>Whatever you write into the Before Script section, it will be generated before the Entity definition.</p>
<p><b>After Script Tab</b></p>	<p>Whatever you write into the After Script section, it will be generated after the Entity definition.</p>
<p><b>Notes Tab</b></p>	<p>Write notes on the entity on this tab. The text will display in a pop-up box when you point your mouse cursor at the entity name in ER diagram.</p>
<p><b>SQL Preview Tab</b></p>	<p>Click  at the bottom of this tab to see the part of SQL code for the entity. Select the <b>WordWrap</b> checkbox to wrap the code on this tab.</p>
<p><b>Relationships Tab</b></p>	<p>Information on parent and child entities, relationships and key attributes. Double-click the selected object to open the <b>Properties</b> dialog.</p>
<p><b>Physical Properties Tab</b></p>	<p>Storage characteristics of table can be defined on this tab.</p>
<p><b>Table Properties Tab</b></p>	<p>Other table characteristics can be defined here.</p>
<p><b>Comment Tab</b></p>	<p>Write comments on the entity on this tab. <b>Note:</b> CTRL+A, CTRL+C, CTRL+X and CTRL+V functions are available on this tab.</p>

## Buttons:



- opens the **Application Variables** form

**OK** - confirms all changes and closes the form

**Cancel** - cancels the changes you have made and closes the form

**Apply** - confirms the changes (the form will remain opened for further edit)

**Help** - opens Help file

## Copy Entities

In Toad Data Modeler there are several methods to copy entities. You can select from the following options.

**i** | TIP: If you find yourself copying objects too much, consider using the **Gallery** feature. See [Gallery](#) for more information.

### A. Drag&Drop techniques + CTRL on Workspace

**Where to use:** On a Workspace

**Example:**

1. Click an entity on Workspace and hold the mouse button down.
2. Press CTRL key.
3. Drag the entity to another place.
4. Release the mouse button and the key.

### B. Copy&Paste functions: Edit | Copy/Paste (CTRL+C, CTRL+V)

**Where to use:**

- On a Workspace
- Between Workspaces
- Between Models

**Example:** Copying an entity to another Workspace

1. Select an entity.
2. Press CTRL+C. Activate the Workspace in which the entity copy will be created. It can be in your current model or in another model.
3. Press CTRL+V
4. A copy of the entity is created on the Workspace you selected.

### C. In the Entities dialog: Drag&Drop techniques + CTRL

**Where to use:**

- In the **Entities** dialog itself (**Model Menu | Model Items | Entities**)
- Between the **Entities** dialog and a Workspace

- Between the **Entities** dialog and **Model Explorer | Entities** folder
- Between the **Entities** dialogs of two different models

**Example:** An entity in the **Entities** dialog of *Model A* needs to be copied to **Model Explorer** of *Model B*:

1. Open the **Entities** dialog in *Model A* in **Model Menu | Entities**.
2. Open **Model Explorer** in *Model B*.
3. Select an entity in the **Entities** dialog of *Model A*, hold down CTRL key.
4. Drag the entity over the **Entities** folder in **Model Explorer** and drop.

#### D. In Model Explorer: Drag&Drop techniques + CTRL

**Where to use:**

- In **Model Explorer** itself
- Between **Model Explorer** and a Workspace
- Between **Model Explorer** and the **Entities** dialog (in a different model as well)
- Between **Model Explorer** and **Entities** dialog of two different models

**Example 1:** Making a copy of an entity in Model Explorer.

1. Unfold the **Entities** folder in the Model Explorer tree.
2. Click an entity and hold the mouse button down.
3. Press CTRL.
4. Drag the entity to the **Entities** folder and drop.

A copy of the selected entity is listed in the Model Explorer tree and the entity appears on all the Workspaces of your model where the **Auto Complete** option is enabled.

**Example 2:** Copying an entity from Model Explorer to a Workspace.

1. Unfold the **Entities** folder in the Model Explorer tree.
2. Click an entity and hold the mouse button down.
3. Press CTRL.
4. Drag the entity to a Workspace.

**Example 3:** Creating an entity copy between Model Explorers of two models. The entity will be copied from *Model A* to *Model B*:

1. Open Model Explorers in both models. You may want to undock at least one of the Model Explorer windows.
2. Click an entity in *Model Explorer A* and hold the mouse button down.
3. Press CTRL.
4. Drag the entity to the **Entities** folder of the *Model Explorer B* and drop.

**i** Note:

- A copied entity has the same properties as its source.
- In some cases, entities cannot be copied between models. It's because they contain specific properties, that do not exist in the target mode. For example, copying two entities from Microsoft SQL 2019 model to MySQL 8.0 model. *Entity A* has Partition Schema, *Entity B* has Fulltext Catalog. Since these features do not exist in MySQL 8.0, the entities will not be copied.
- Even though it is possible to copy and paste objects to a different model of different database platform or version, it is encouraged to use **Model Convert** function, even for single objects. The copy-pasting method is faster, but more error-prone, while the **Model Convert** method is slower, but more robust.

## Move Entities

In Toad Data Modeler there are several methods to move entities. You can select from the following options.

### **To move entities on a Workspace, use**

- Drag&Drop techniques

or

- Keyboard arrows

**i** TIP:

1. Select **Settings | Options | Graphics | Move Objects by (mm/10)** to set the size of a step to move (in tenths of millimeters).
2. Select an entity, press SHIFT, hold it down and use the keyboard arrows to change size of the entity box.

### **To move multiple entities at once**

1. Make multiple selection of entities (see **Select Objects** for more information).
2. Point your mouse cursor at any of the selected entities, click and hold the mouse button down.
3. Drag the objects to the required position and drop. All selected entities will be moved there, including their relationships.

### **To move entities to another Workspace or another model, select from the following options:**

- Cut&Paste functions: **Edit | Cut/Paste**, or via shortcuts CTRL+X, CTRL+V
- Drag&Drop techniques between the **Entities** dialogs of two different models
- Drag&Drop techniques between the **Entities** dialog and **Model Explorer**
- Drag&Drop techniques in or from **Model Explorer**
- Drag&Drop techniques between **Entities** dialog/**Model Explorer** and Workspace

**Example:** Moving an entity from *Model A* to *Model B* using **Model Explorer**:

**Method 1:** Moving an entity between Model Explorers of the two models.

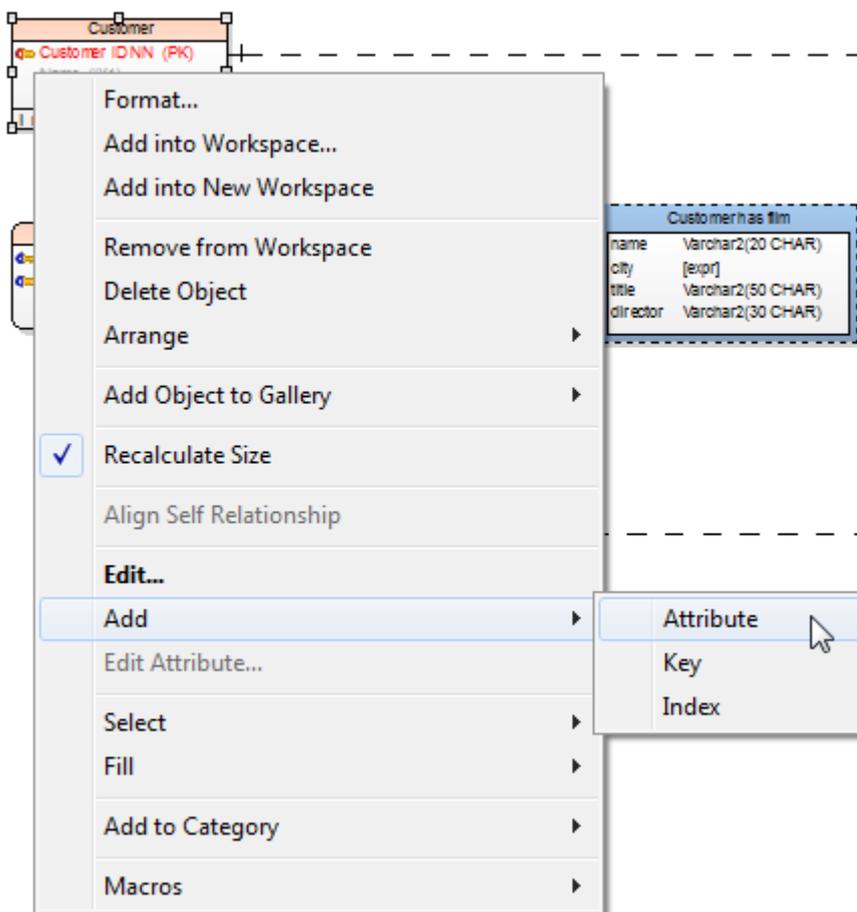
1. Open Model Explorers in both models. You may want to undock at least one of the Model Explorer windows.
2. Select an entity in *Model Explorer A* and hold the mouse button down.
3. Drag the entity to the **Entities** folder in the *Model Explorer B* and drop.

**Version B:** Moving an entity from *Model Explorer A* directly to a Workspace of *Model B*.

1. Undock the *Model Explorer A* if it's docked, and click the Workspace tab of *Model B* in the Application Window.
2. Select an entity in *Model Explorer A* and hold the mouse button down.
3. Drag the entity to a Workspace of *Model B* and drop.

## Entity Right-Click Options

**Right-click the selected entity on the Workspace to see the following options:**



Option	Description
Format	Opens the <b>Object Format</b> dialog for the selected entity.
Add into Workspace	Opens the <b>Workspaces</b> dialog where you can select

Option	Description
	a Workspace to add the entity to.
Add into New Workspace	Creates a new Workspace and adds the entity to it.
Remove from Workspace	Removes the selected shortcut from particular Workspace.
Delete Object	Deletes selected entity from model.
Arrange	Arranges the entity in another layer. <a href="#">Arrange Objects in Layers</a>
Copy Object Layout to	Copies the layout of the selected object to another <b>Workspace</b>
Add Object to Gallery	Adds objects to selected Gallery.
Recalculate Size	Adjusts the entity size to the length of its columns.
Align Self Relationship	Aligns self relationship.
Edit	Opens the <b>Entity Properties</b> form.
Add	Creates a new object (Attribute, Key or Index) in the selected entity.
Change Script	Contains one option that allows you to compare selected entity with any other entity in another model or database and generate Change Script.
Edit Attribute...	Opens <b>Attribute Properties</b> form (the item is active if attribute is selected)
Select	
Select Parent Objects	Selects parent objects of the selected entity on Workspace.
Select Child Objects	Selects child objects of the selected entity on Workspace.
Select Parent and Child Objects	Selects parent and child objects of the selected entity on Workspace.  <div style="display: flex; align-items: center;"> <div style="border-left: 1px solid black; padding-left: 5px; margin-right: 5px;">i</div> <div>TIP: Use these options for next multiple copy, move, format change etc.</div> </div>
Fill	
Fill Parent Objects	Displays shortcuts of parent objects of the selected entity on Workspace.
Fill Child Objects	Displays shortcuts of child objects of the selected entity on Workspace.
Fill Parent and Child Objects	Displays shortcuts of parent and child objects of the

Option	Description
	selected entity on Workspace.
Add to Category	Adds object to selected category.
Macros	Shows available macros for the selected entity.

## Create Attributes

You can create attributes in:

- Entity right-click menu on Workspace
- **Entity Properties** dialog
- **Attribute Properties** dialog
- Model Explorer

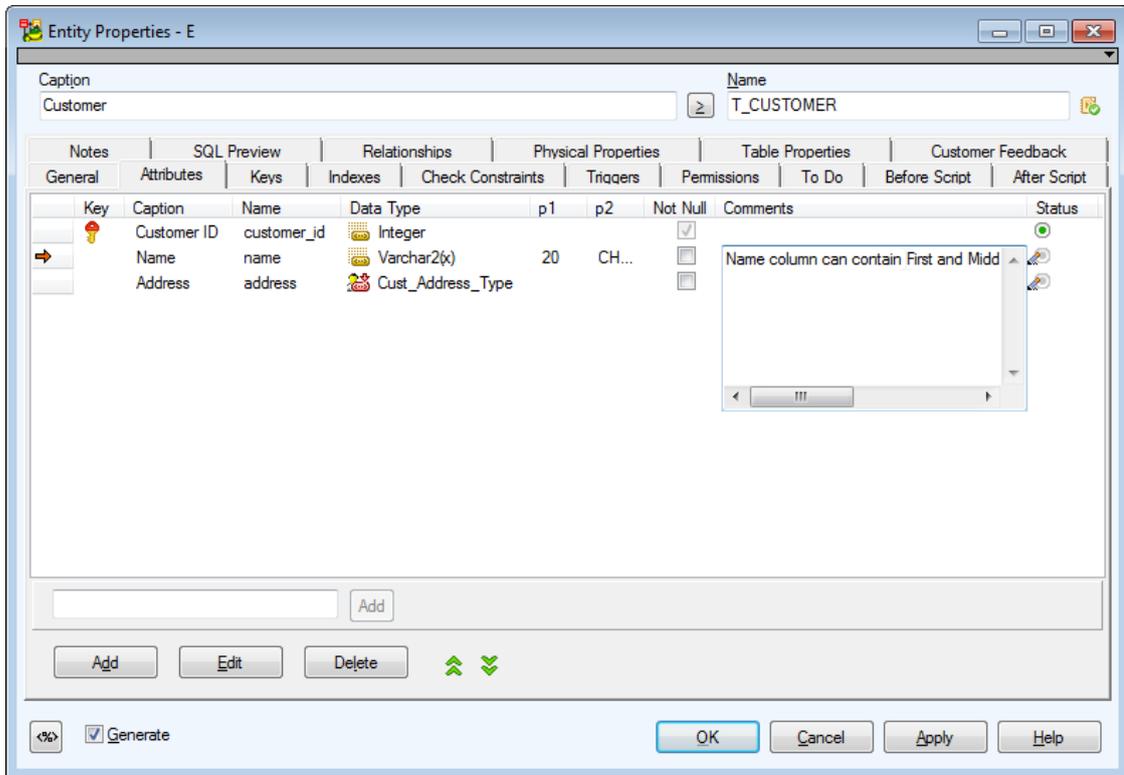
### *To create an attribute on the Workspace*

Right-click an entity on the Workspace and select **Add | Attribute**. The **Attribute Properties** dialog opens.

**i** Note: Select the **Primary Key (Unique)** checkbox to set the attribute as PK or unique attribute.

## To create an attribute in the Entity Properties form

1. Double-click an entity on Workspace.
2. In the **Entity Properties** form, switch to the **Attributes** tab and click **Add**.



## Entity Properties attribute columns

Column/Option	Description
Key	Graphical representation of keys of a particular attribute
Caption	Logical attribute name
Name	Physical attribute name
Data Type	Data Type of an attribute <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p><b>i</b> TIP: You can set a default data type for newly created attributes. See <b>Settings Menu   Options   Physical Model   *database platform and version*   Default Data Type</b> combo-box. (The selected data type will be also applied to Dictionary Types and Domains.)</p> </div>
p1	Parameter 1. Only some of the available data types have this parameter. Defines properties of the selected data

Column/Option	Description
	type, e.g. length in case of the Char data type.
p2	Parameter 2. Only some of the available data types have two parameters. E.g. the Decimal data type has two parameters, which define precision and scale.
Not Null	When checked, the attribute cannot be empty.
Comments	Comments or descriptions related to the attribute
Status	Shows status of attributes in grid. <a href="#">Status of Items in Grids</a>

## Buttons:



- opens the **Application Variables** form

**Smaller Add button**- adds new attributes quickly, just enter attribute name and then click **Add**.

**Bigger Add button** - adds an attribute

**Edit** - opens **Attribute Properties** of the selected entity

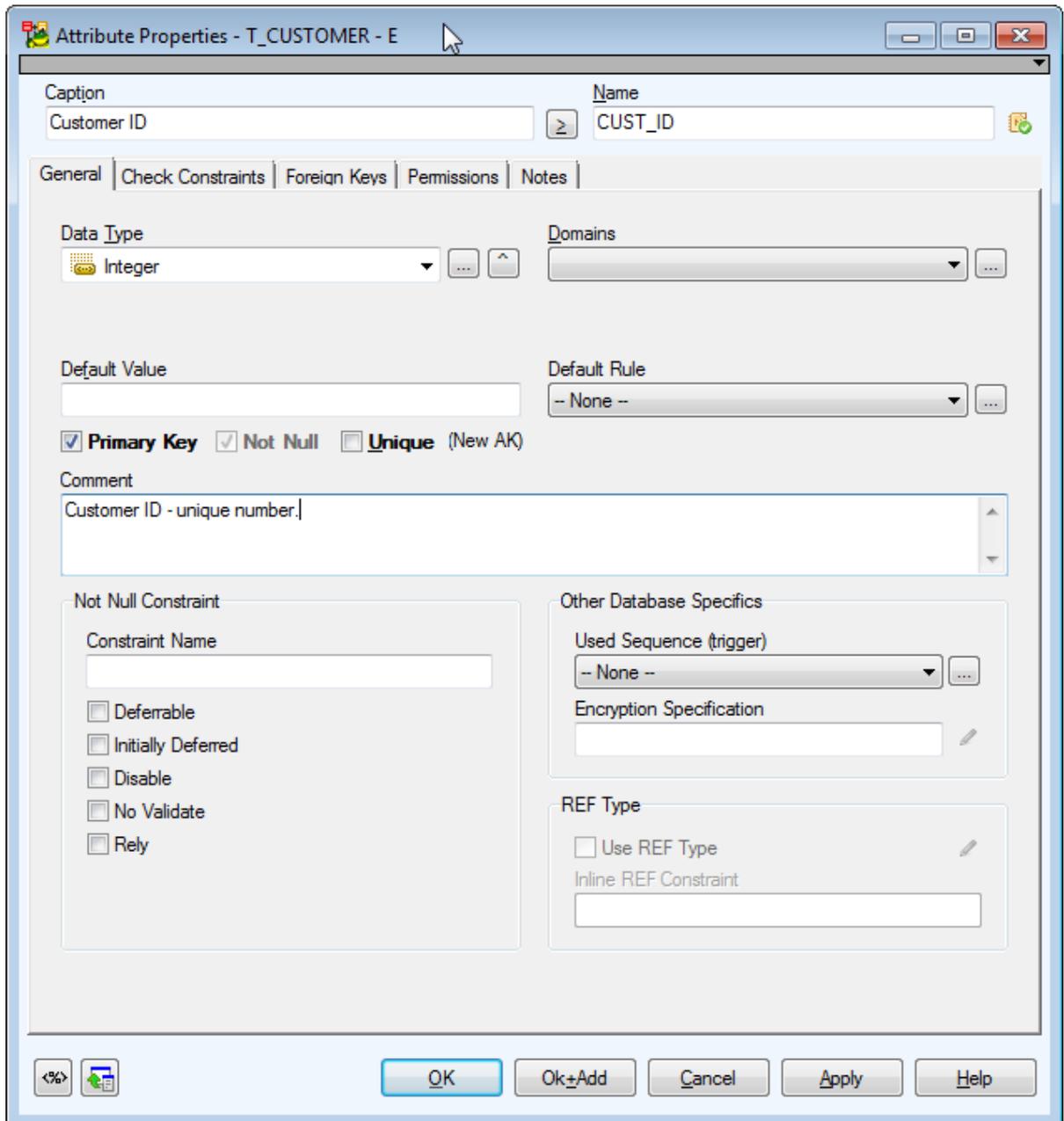
**Delete** - deletes the selected attribute



- moves the selected attribute up or down in the list

### **To create another attribute in the Attribute Properties form**

Click **OK+Add**.



### To create an attribute in Model Explorer

Unfold the **Entities** folder | Unfold the specific entity folder | Right-click the **Attributes** folder | **Add Attribute**.

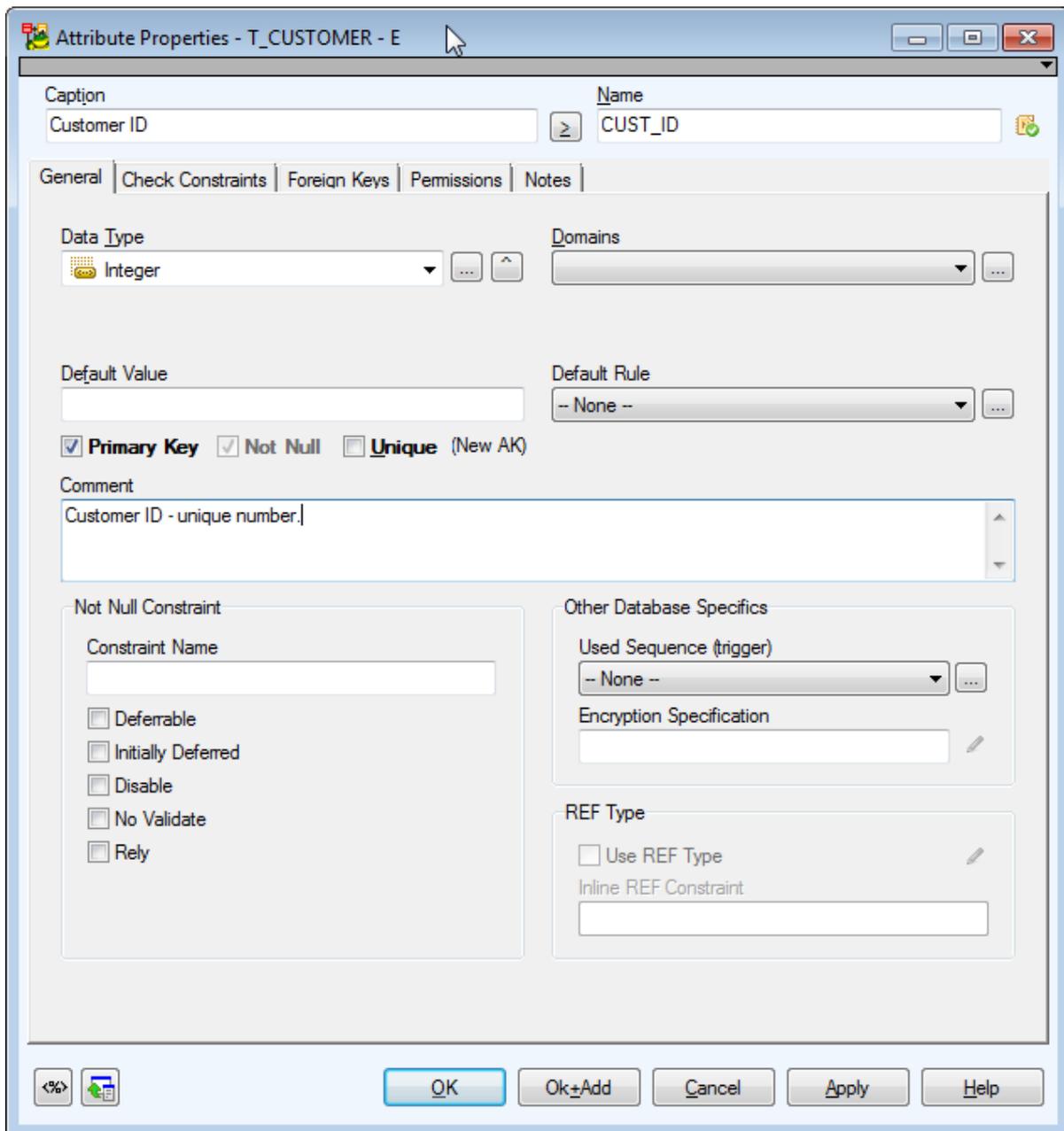
### Edit Attributes

- In the **Entity Properties** dialog | **Attributes** tab, double-click the selected attribute.

or

- Find the attribute in **Model Explorer** | *EntityName* folder | **Attributes** | Double-click the selected attribute.

**Example:** The **Attribute Properties** dialog (Oracle 10g db):



Tab/Option	Description
Object Navigator Dropdown Menu	<p>Object Navigator Dropdown Menu can be expanded or collapsed by clicking the right black arrow in top right corner.</p> <p>All attributes in your entity are listed here. This allows you to edit them quickly and comfortably from one place.</p> <p><b>Tip:</b> After you finish editing an entity, click <b>Apply</b> to confirm changes and select another entity from the Object Navigator box.</p>

Tab/Option	Description
<b>General Tab</b>	General properties of attribute
Caption	Logical column name
Name	Physical attribute name See <a href="#">About Naming Conventions</a> for more information.
Primary Key	Select this checkbox to set the attribute to be a part of primary key.
Not Null	Select this checkbox to set the attribute Not Null. See <a href="#">NotNull Property for PK and AK Attributes</a> for more information.
Unique	Select this checkbox to set the attribute as unique. See <a href="#">Unique Attributes</a> for more information.
Data Type	Data Type selection box <div style="border-left: 1px solid #0070C0; padding-left: 10px; margin-left: 20px;"> <p><b>i</b> TIP:</p> <ul style="list-style-type: none"> <li>• Before you create new attributes, set a default data type, including parameters, for particular database. See the <b>Settings</b> menu   <b>Options</b>   <b>Physical Model</b>   particular database version   <b>Default Data Type</b> combo-box. (The selected data type will be applied also for Dictionary Types and Domains.)</li> <li>• Set maximal length for display of data types in the physical ERD. Select <b>Settings</b>   <b>Options</b>   <b>Graphics</b>   <b>Maximal Number of Characters for Displayed Data Type</b>.</li> </ul> </div>
Domains	Domains selection box <div style="border-left: 1px solid #0070C0; padding-left: 10px; margin-left: 20px;"> <p><b>i</b> Note: It's not possible to assign an attribute a data type and a domain at the same time. If you select a domain, data type will be set automatically from the domain.</p> </div>
Default	Default value definition
Default Rule	Default rule selection
	<div style="border-left: 1px solid #0070C0; padding-left: 10px; margin-left: 20px;"> <p><b>i</b> Note: Other options on the <b>General</b> tab vary according to a database type. Options specific for your database can be found in the <b>Databases</b> chapter.</p> </div>
<b>Check Constraints Tab</b>	On this tab, you can add, edit and delete check constraints.
<b>Foreign Keys Tab</b>	Details of a foreign key attribute are listed here.
Entity	Name of parent entity

Tab/Option	Description
Relationship	Name of relationship connecting the parent entity and child entity
Attribute	Name of attribute
<b>Permissions Tab</b>	On this tab, you can assign a User or User Group permissions for the selected attribute.
<b>Notes Tab</b>	A tab for notes on the attribute. The text will display in a pop-up box when you point your mouse cursor at the attribute name in ER diagram (the Attributes item has to be selected in Display Level).
<b>Not Null Constraint Tab</b>	Options related to Not Null Constraint definition.
	Click this button to open the parent form ( <b>Entity Properties</b> form). <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p><b>i</b> Note: Above the Object Navigator Dropdown Menu, you can see name of the entity that the attribute belongs to. The Properties dialog of this entity will open.</p> </div>

## Display Attributes in Model

### *To display attributes in your ER diagram*

From the **Display Level** box on the toolbar, select **Attributes**.

### *To define a display level for particular Workspace*

Right-click the work area | **Workspace Format...** | **Entities** tab | **Display Level** | select **Attributes**.

### *To set the default Display Level properties for all models that you will create*

Select **Settings** | **Options** | **Physical Model** | **Entity** tab | **Display Level**.

### *To define different colors for attributes on your Workspace*

Select **Settings** | **Options** | **Model** section | **Physical Model** | **Entity** tab | **Attribute Colors** area.

The screenshot shows the 'Physical Model' dialog box with the 'Entity' tab selected. The 'Attributes' sub-tab is active, displaying various settings for attribute display. The 'Default' section includes a 'Display Level' dropdown set to 'Attributes' and several checked options: 'Align', 'Display Data Types', 'Display Keys Graphically', 'Display Key and Index Marks', 'Display Indexes', 'Display Not Null Mark', and 'Gradient Effect'. The 'Display Dictionary Types as Data Types' option is unchecked. The 'Attribute Colors' section has five dropdowns: 'Primary Key' (Red), 'Primary Foreign Key' (Blue), 'Foreign Key' (Green), 'Mandatory Attribute' (Black), and 'Optional Attribute' (Medium Gray). The 'Form Settings' section has one unchecked option: 'Display Data Warehouse Type and Size'.

## Order Attributes

In the **Entity Properties** form | **Attributes** tab, you can order attributes:

- Manually via the green arrows at the bottom of the form
- Automatically (e.g. by Name, Caption, Data type etc.)

### Manual Order

The order you will set via the green arrows will be applied in particular entity box in the Designer (all workspaces).

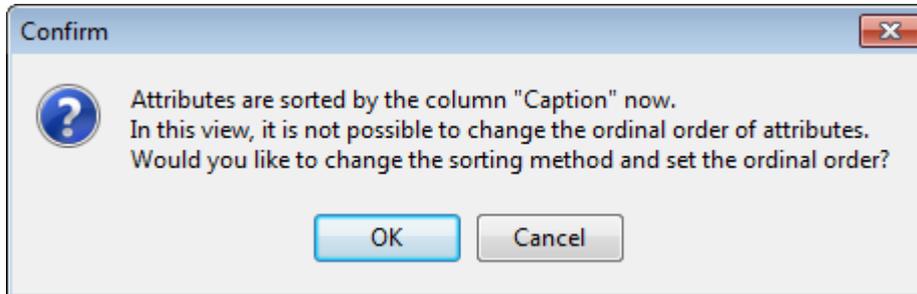
**Example:** Attributes in the *Film Entity Properties* form have been ordered manually using the green arrows and finally confirmed **Apply**. Now see the *Film* entity in Designer and compare its order of attributes with the order set in the *Film Entity Properties* form. The order is identical.

## Automatic Order

Click the appropriate column name - e.g. click the *Name* column to sort attributes alphabetically.

This function offers you just a different view on the attributes in the **Entity Properties** form. It does not influence order of attributes in entity box in the Designer.

If you sort attributes this way, you cannot manually change order of attributes via the green arrows. If you click a green arrow, this message will display:



Click **OK** to activate the green arrows and order the attributes manually.

### **i** Note:

- No green arrow is enabled when there is no attribute or when there are attributes but none is selected.
- If any sorting function is turned on (e.g. sorting by name), green arrows are both active, so you can click any of them to turn off the sorting and set the ordinal order.

## Copy Attributes

In Toad Data Modeler there are several methods and places where to copy attributes. You can select from the following options.

### A. CTRL + Drag&Drop techniques on the Workspace

**Use:** This way, you can copy attributes within an entity and between entities on one WS. You can copy one or more attributes at one jump.

**Example:**

1. See [Display Attributes in Model](#) for more information.
2. Select an entity of which attributes you want to copy.
3. Press CTRL and hold the key down. Click the attributes that you want to copy.
4. Release CTRL key.



5. Now click the selected objects, hold the mouse button down. Press CTRL again and hold the key down.
6. Drag the selected attributes to a target entity. (If you want to copy the attributes within the entity, drag it a little in the entity box itself.)
7. Release your mouse button and CTRL.

## B. In the Entity Properties form | Attributes tab: CTRL + Drag&Drop techniques

### Use:

- In the **Entity Properties** form, you can copy attributes within an entity and between entities of the same model or a different model.
- From the **Entity Properties** form and Workspace, and vice versa
- Between the **Entity Properties** form and **Model Explorer**
- Multiple selection is possible here - use SHIFT or CTRL keys

### Example:

1. Open the *Customer Properties* form of the *Customer* entity and the *Employee Properties* form of the *Employee* entity.
2. Click the **Attributes** tabs in both forms.
3. Press CTRL and hold the key down.
4. Click the *Name* attribute in the *Customer Properties* form and drag it to the *Employee Properties* form | **Attributes** tab.
5. Release your mouse button and CTRL.

## C. In Model Explorer

### Use: You can make copies of attributes:

- In **Model Explorer** itself
- From **Model Explorer** to a Workspace and vice versa
- Between **Model Explorer** and the **Entity Properties** form | **Attributes** tab
- Between two **Model Explorers** of two different models (see the following examples)
- Multiple selection is possible here - use SHIFT or CTRL keys

### Example: Making a copy of attributes from Model Explorer to a WS.

1. Unfold the **Entities** folder | '*EntityName*' | **Attributes** tab in **Model Explorer** tree.
2. Make selection of attributes.
3. Press CTRL, hold it down.
4. Click any selected attribute and drag the attribute to the appropriate entity box on the Workspace.
5. Release the mouse button and CTRL.

**i** Note:

- An attribute is copied with all its properties.
- It's possible to copy objects between models of a different database. However, some specific object properties cannot be copied - properties that are in the source model but do not exist in target model. E.g. Copy of entity from Microsoft SQL 2019 to MySQL 8.0. - *Entity A* has Partition Schema, *Entity B* has Fulltext Catalog. In these cases, the entities will not be copied to MySQL model.

## Move Attributes

In Toad Data Modeler there are several methods how to move attributes. You can select from the following options.

### A. Drag&Drop techniques on the Workspace

**Use:** This way, you can move one or more attributes to another entity at one jump.

**Example:**

1. Display attributes in your model.
2. Select an entity of which attribute/attributes you want to move.
3. Press CTRL and hold the key down. Click the attributes that you want to move.
4. Release CTRL.
5. Drag the selected attributes to a target entity.

### B. In Entity Properties Form | Attributes tab: Drag&Drop techniques

**Use:**

- In the **Entity Properties** form, you can move attributes within an entity and between entities. (To change order of attributes within an entity, use the green arrows.)
- From the **Entity Properties** form to a Workspace, and vice versa
- Between the **Entity Properties** form and **Model Explorer**
- Multiple selection is possible - use SHIFT or CTRL keys

**Example:**

1. Open the *Customer Properties* form of the *Customer* entity and the *Employee Properties* form of the *Employee* entity.
2. Click the **Attributes** tabs in both forms.
3. Select attributes in the *Customer Properties* form and drag them to the *Employee Properties* form | **Attributes** tab.

### C. In Model Explorer

**Use:** You can move attributes:

- Between entities in **Model Explorer** itself
- From **Model Explorer** to an entity on the Workspace directly
- Between **Model Explorer** and the **Entity Properties** form | **Attributes** tab

- Between two **Model Explorers** of two different models
- Multiple selection is possible - use SHIFT or CTRL keys

**Tip:** Use CTRL to select attributes of various entities in Model Explorer, and move them at one jump to another entity (of the same model or even a different model).

**Example:** Moving an attribute in Model Explorer.

1. Unfold the **Entities** folder in Model Explorer tree.
2. Select attributes (SHIFT or CTRL) and drag them to the **Attributes** or '*EntityName*' folder of another entity.

## Delete Attributes

- In the **Entity Properties** form | **Attributes** tab, select the attribute and click **Delete**.

or

- Find the attribute in **Model Explorer** | *EntityName* folder | **Attributes** | right-click and select **Delete Item**.

## Parent Attributes (Rolenames)

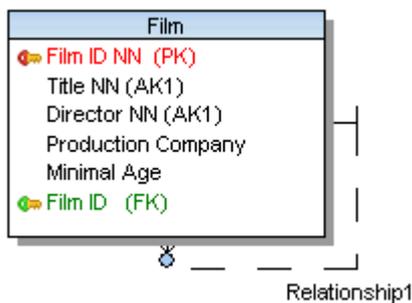
You can change the attribute name e. g. in the following cases:

- When you create a self-relationship.
- Whenever you want to change a Foreign Key attribute name.

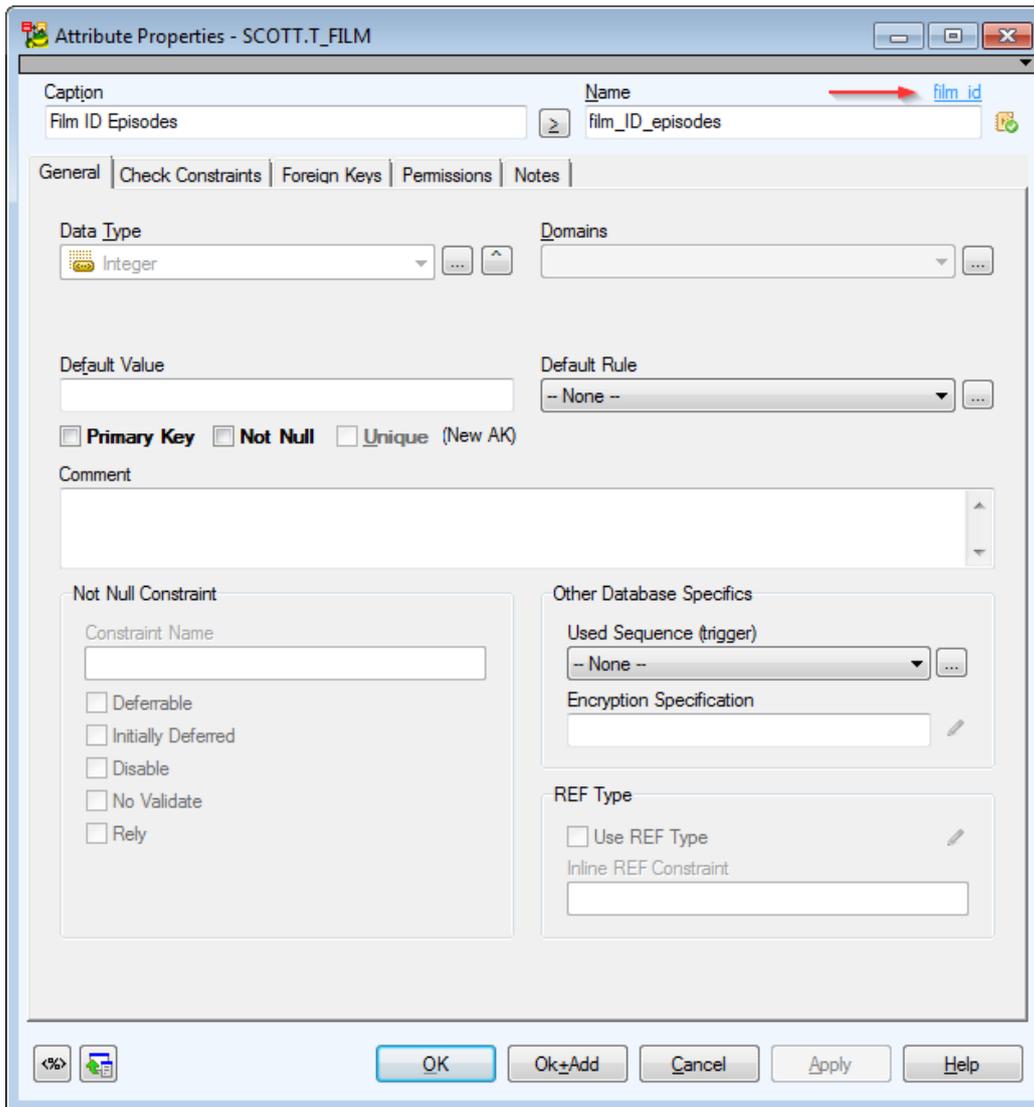
### Scenario:

You have just added a self-relationship to the *Film* entity.

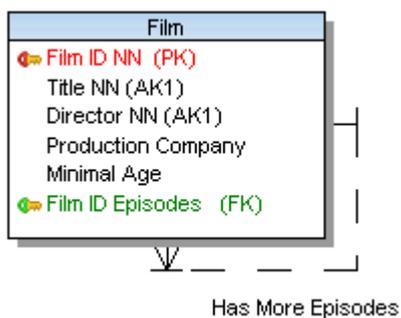
1. A copy of your identifying key attribute has been created in the entity - *Film ID*. Now you have two *Film ID* attributes in the *Film* entity.



2. Edit the newly created *Film ID* attribute, define a new **Name**. The link to parent attribute is displayed on top right-hand corner of the form.



3. Confirm **OK** and see the change on the Workspace.



## Display Keys in Model

### To display keys in your ER diagram

From the **Display Level** dropdown menu on the **Display Toolbar**, select **Primary Keys** or **PK and FK Keys** or **All Keys**.



### To define display level for a specific Workspace

Right-click the work area | **Workspace Format** | **Entity** tab | **Display Level**.

### To display keys graphically in your ER diagram

Right-click the work area | **Workspace Format** | **Entity** tab | check the **Display Keys Graphically** checkbox.

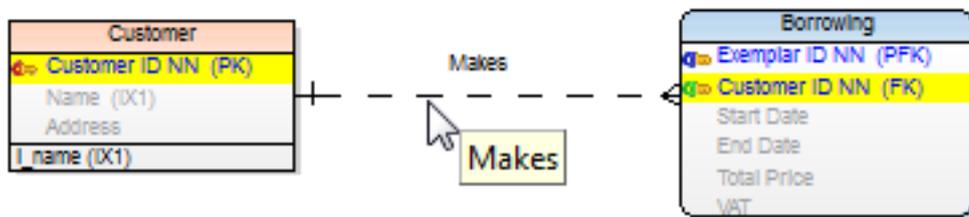
### To change the default key attributes color

Select **Settings** | **Options** | **Model** section | **Physical Model** | **Entity** tab | **Attribute Colors** area.

### To highlight key attributes on your Workspace

Point your cursor at a relationship line.

**i** TIP: When you point your mouse cursor at a relationship line, parent and child attributes are highlighted on the Workspace. You can set the color at: **Settings** | **Options** | **Graphics** | **Colors** area | **Highlight Color**.



## Create Keys

An entity can have a primary key and many alternate keys. The keys are stored in the **Keys** tab in the **Entity Properties** form.

### To create a key

- Right-click an entity on the Workspace and select **Add** | **Key**. The **Key Properties** dialog opens.

### To create a PK (unique) attribute

1. Right-click an entity on the Workspace and select **Add | Attribute**. The **Attribute Properties** dialog opens.
2. On tab **General**, select the **Primary Key (Unique)** checkbox.

### To assign an attribute to key

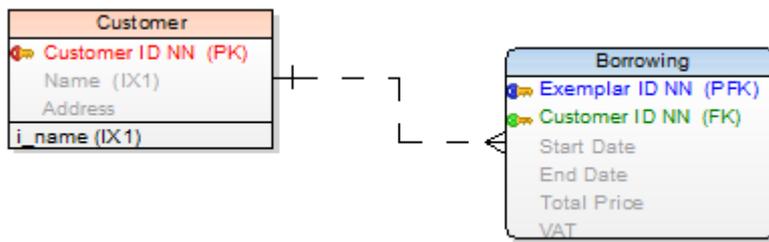
- In the **Entity Properties** dialog, **Attributes** tab, double-click the empty space in the **Key** column next to the selected attribute.

or

- In the **Key Properties** dialog | **Attributes** tab.

**i** Note:

- Primary keys are graphically marked by red key by default.



- A primary key can be added to more attributes.
- It is possible to delete primary keys. It is not possible to delete alternate keys of unique attributes.

### NotNull Property for PK and AK Attributes

Toad Data Modeler allows you to check/uncheck the NotNull property for PK and AK attributes. This is possible only if the **Allow Null Attributes in Keys** checkbox is selected (unchecked by default).

#### To select this checkbox

Select **Settings | Options | Physical Model** and select the checkbox.

If this checkbox is not selected, Toad Data Modeler doesn't allow you to uncheck the NotNull checkbox in key attributes.

The availability of this option differs by database:

Database	Allow NULL in PK	Allow NULL in AK
DB2 z/OS v. 11	n/a	n/a
DB2 v. 8, 9.x, 10.x	n/a	n/a
Greenplum	n/a	available
Ingres 9.3, 10	n/a	n/a
Microsoft Access	available	available
Microsoft Azure SQL Database	n/a	available

Database	Allow NULL in PK	Allow NULL in AK
SQL Server 2012	n/a	available
SQL Server 2014	n/a	available
MySQL 5.5 and higher versions	n/a	available
Oracle	available	available
PostgreSQL	n/a	available
SQLite 3.7	available	available
Sybase ASE, Sybase IQ	n/a	n/a
Sybase SQL Anywhere	n/a	n/a
Teradata	n/a	n/a

If the **Allow Null Attributes in Keys** checkbox is selected:

- When you assign an attribute to a key (PK, AK), the **Not Null** checkbox of the attribute will be checked. (But you are able to uncheck it)
- During propagation of the key via the identifying relationship, the Not Null property in child attribute is inherited from the parent attribute. The only exception is when a database doesn't support Null value in primary key, which would be created via the propagation.
- During Model Conversion, different settings of the Not Null property and its support in different databases are taken into consideration.
- For databases that support Null value in child attribute, the **Mandatory Parent** checkbox is selected in the **Relationship Properties** dialog and should behave coherently to Null value in child attribute as well as it behaves for non-identifying relationships.
- For databases that support Null value in child attribute, the settings of the **Synchronize NotNull with Mandatory Parent** option work the same way for PFK as for FK.

See [Synchronization of NotNull and Mandatory Parent](#) for more information.

## Edit Keys

- In the **Entity Properties** dialog | **Keys** tab | double-click the key.

or

- Find the key in **Model Explorer** | *EntityName* folder | **Keys** | Double-click (or right-click and select **Edit**).

Attributes Tab	Description
Available	A list of all non-assigned attributes of the entity
Selected	Attribute(s) that have been assigned the key.
	Assigns selected attribute to the key. <b>Tip:</b> - For multiple selection of attributes, use SHIFT or CTRL.



Removes the selected attribute from the key.  
**Tip:** - For multiple selection of attributes, use SHIFT or CTRL.



Assigns all attributes at once.



Removes all attributes from the key at once.

#### Notes Tab

You can write notes related to the key you are editing.

#### Using Index Properties Tab

Definition of other Key properties, dependent on the database platform and version you are using.

## Delete Keys

- In the **Entity Properties** form | **Keys** tab, select a key and click the **Delete** button.

or

- Find the key in **Model Explorer** | *EntityName* folder | **Keys** | right-click the key and select **Delete Item**.

**i** Note: It is possible to delete a primary key. It's not possible to delete an alternate key of unique attribute.

## Select Parent Key for Relationship

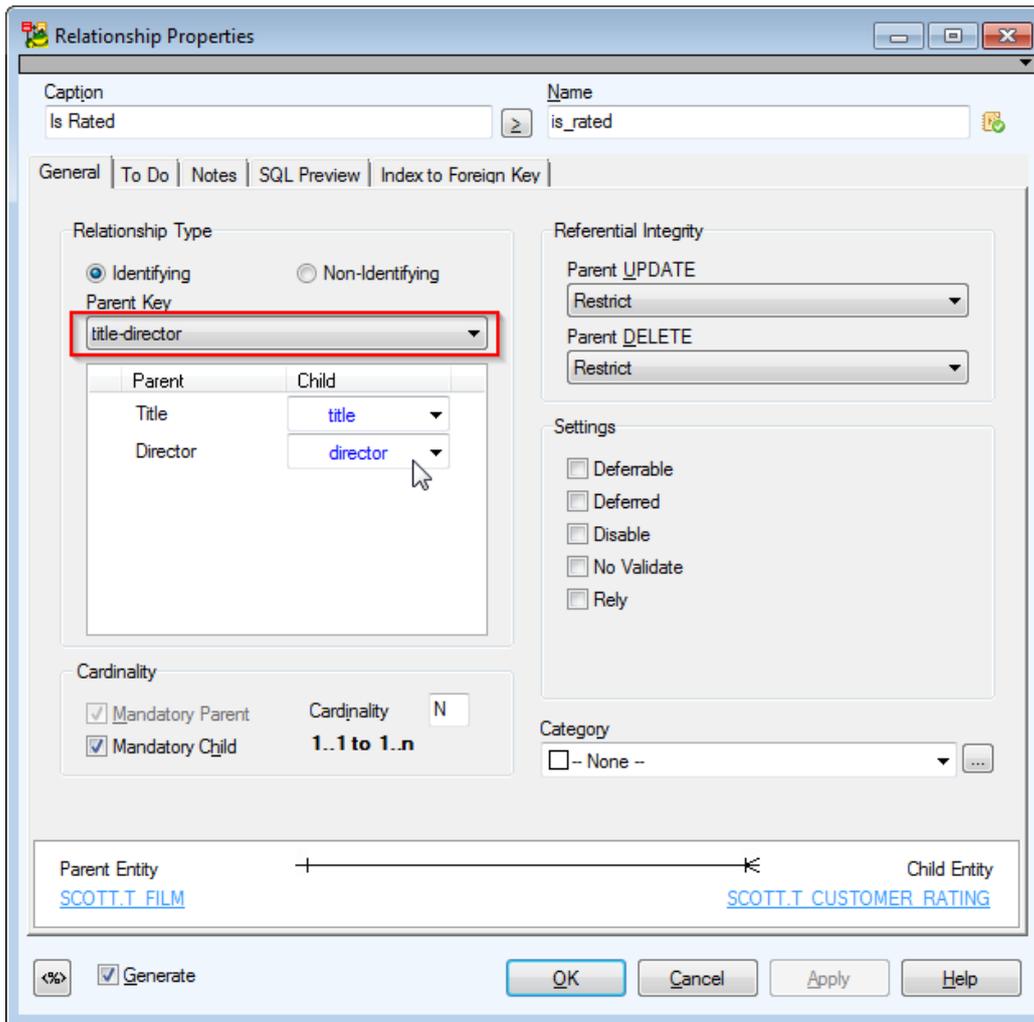
### Linking Method

In Toad Data Modeler, you can link parent and child entities:

- Through a Key (Primary or Alternate key) of parent entity
- Through Unique Index of parent entity

### To set a linking method

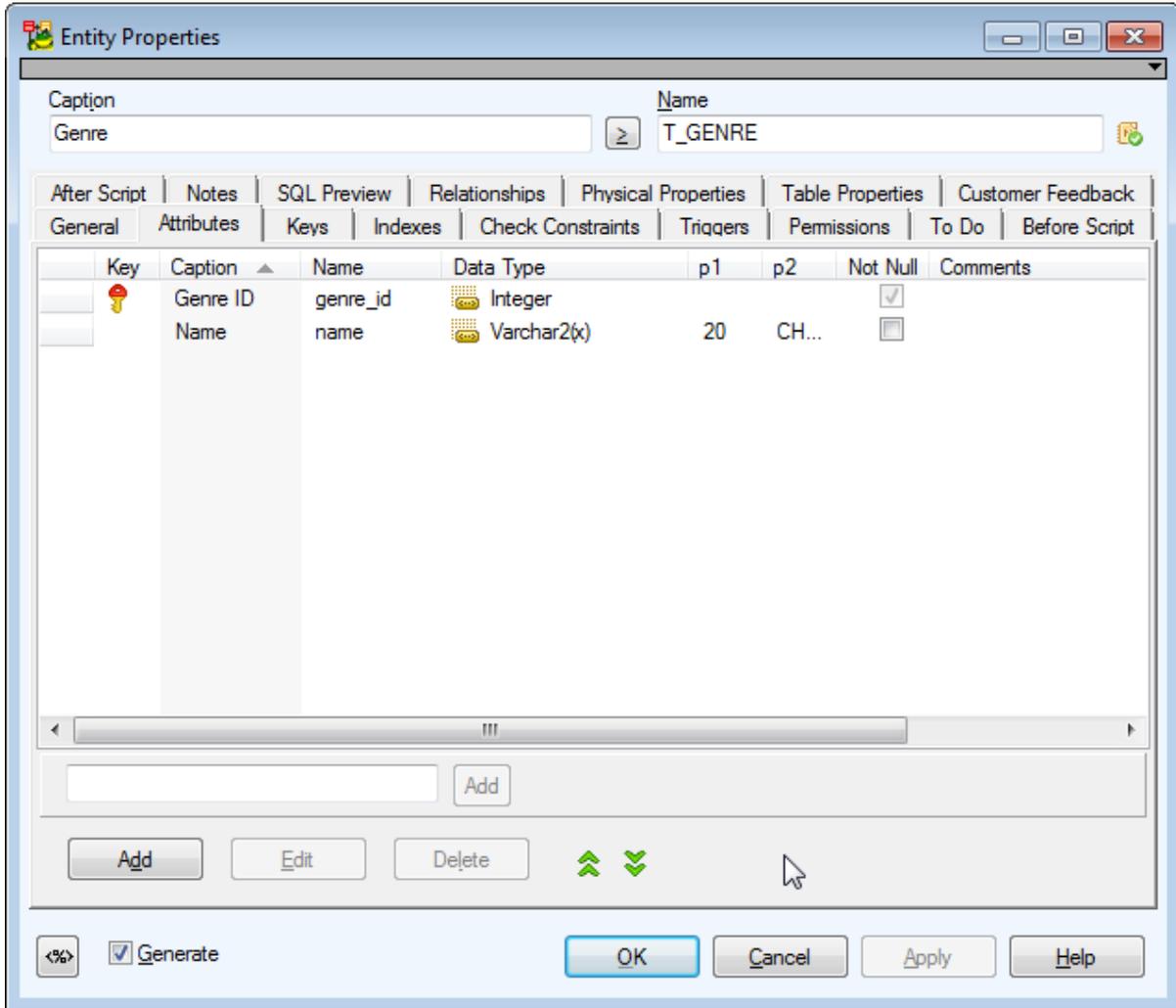
1. Double-click the selected relationship.
2. From the **Parent Key** box, select the appropriate linking method.



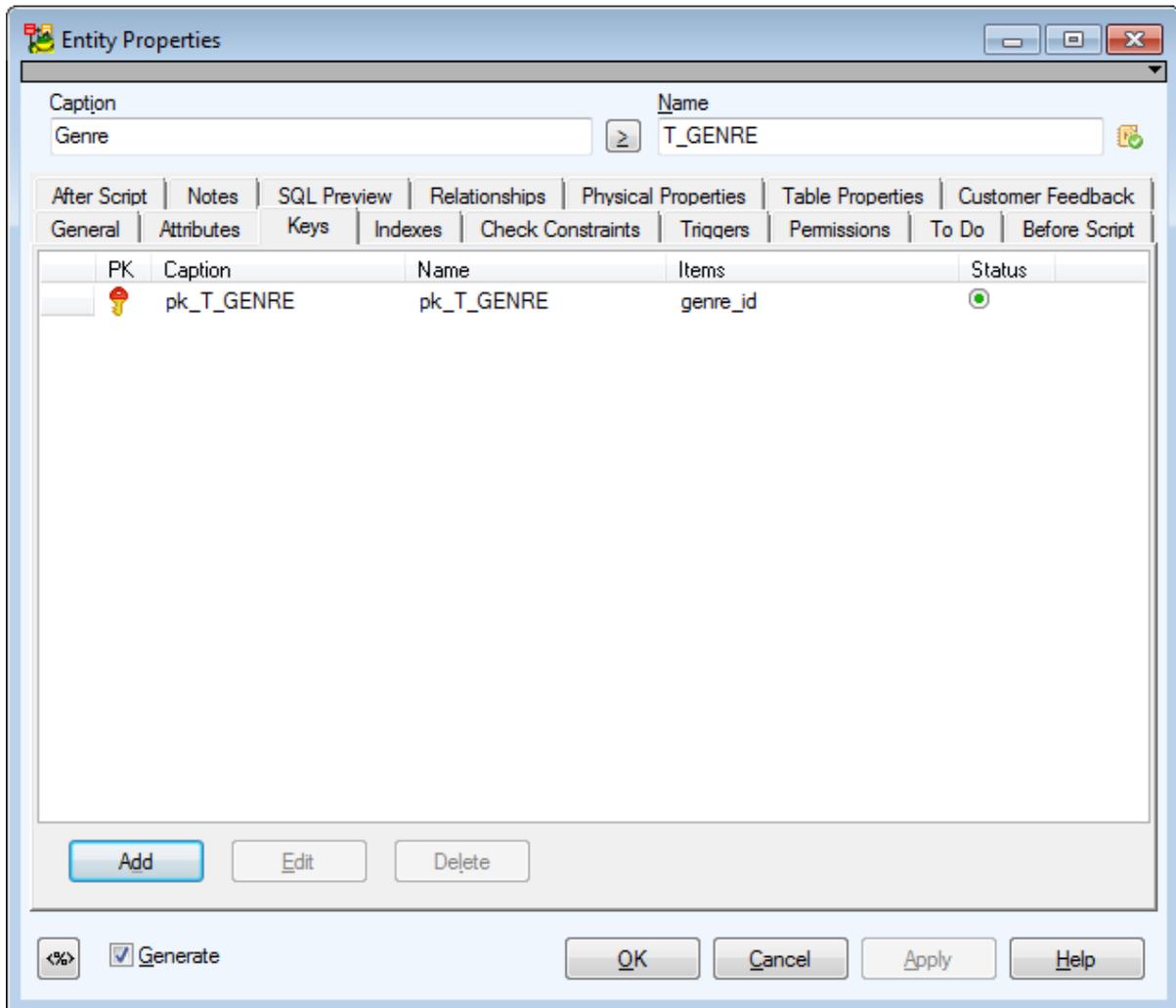
## Unique Attributes

Toad Data Modeler creates new alternate keys for unique attributes automatically.

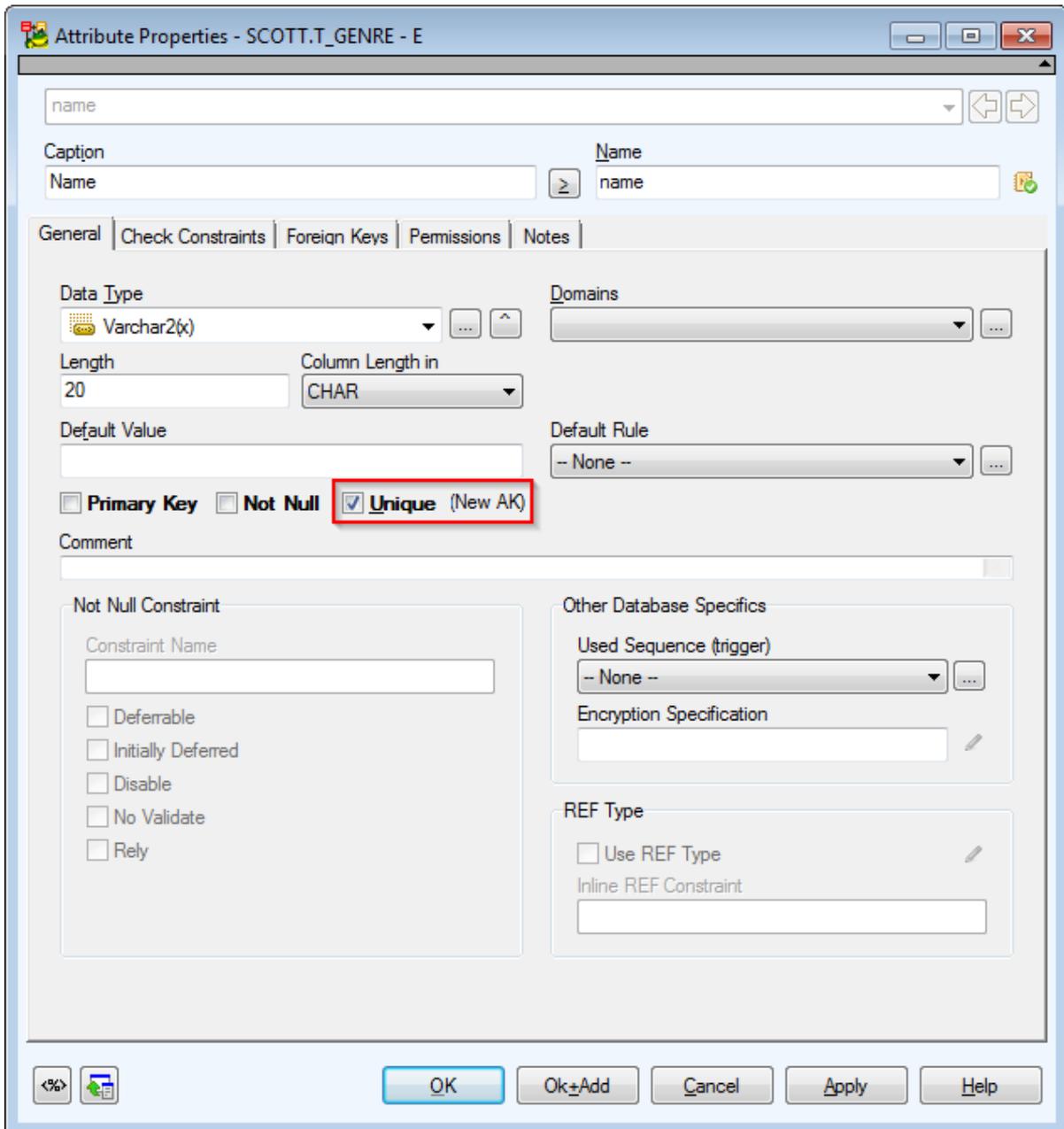
**Example:** There are two attributes in the *T\_GENRE* entity.



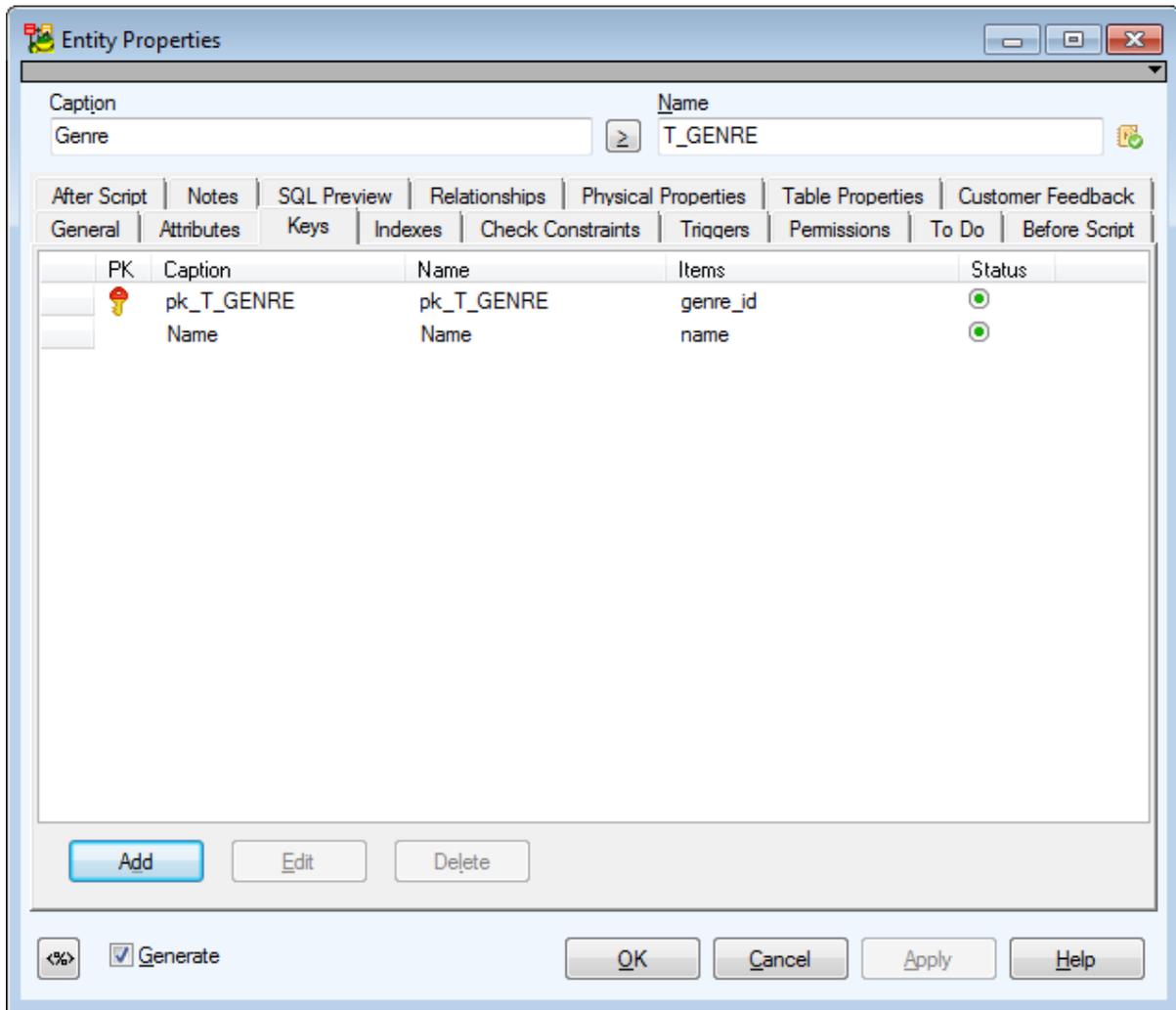
No alternate key exists in this entity.



Edit the *Name* attribute and select the **Unique** checkbox.



New alternate key has been added to the entity automatically.



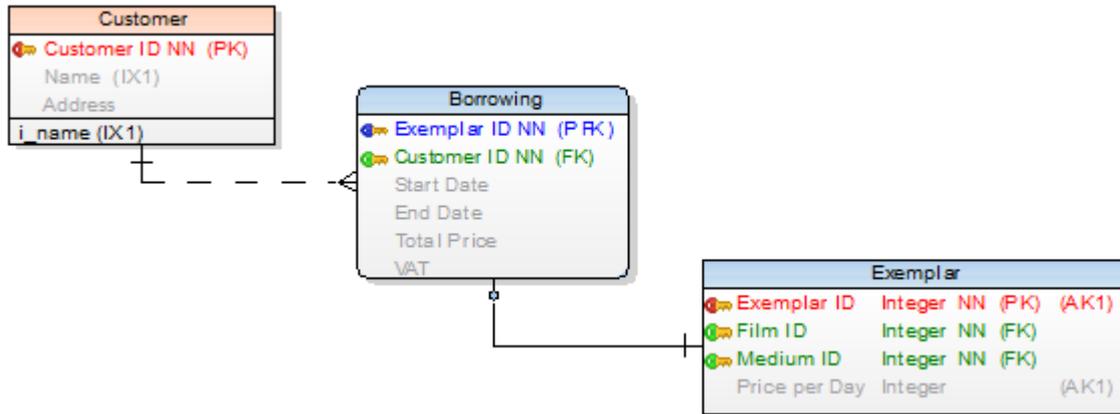
## Foreign Keys in Toad Data Modeler

As soon as you create a relationship in Toad Data Modeler, foreign keys are created automatically in the child entity (automatic key migration).

There are two types of foreign keys:

- Foreign keys (FK, graphically marked by green key by default)
- Primary Foreign keys (PFK, graphically marked by blue key by default)

The type of FK depends on what relationship you create, see [Relationship Types](#) for more information.



### Synchronization of Not Null and Mandatory Parent

Select **Settings | Options | Physical Model | General tab | Synchronize Not Null with Mandatory Parent**.

By default, this checkbox is selected. You can change the settings during your modeling, the new setting will be used by default in all new items you create in your model.

- Synchronization enabled: If cardinality has set Mandatory Parent, a FK attribute will be automatically Not Null. If you clear the **Not Null** checkbox of the FK attribute, Mandatory Parent option will be unchecked.
- Synchronization disabled: If you uncheck the **Not Null** checkbox of a FK attribute, the Mandatory Parent current status remains unchanged.

## Foreign Keys in the Attribute Properties Dialog

If you open the **Attribute Properties** dialog | **Foreign Keys** tab, you can see details on particular foreign key:

- Name of the parent entity
- Name of relationship connecting the parent and child entity
- Name of PK Attribute

## Foreign Keys in the Relationship Properties Dialog

In the **Relationship Properties** dialog | **Foreign Keys** tab, you can set a linking method between parent and child entities.

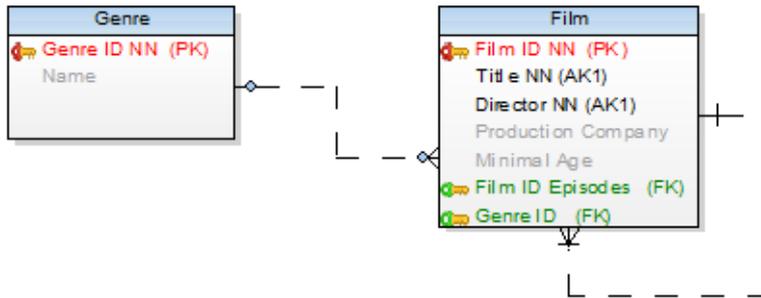
See [Select Parent Key for Relationship](#) for more information on available linking methods.

### FK Mapping

In Toad Data Modeler, you can control your foreign keys. Foreign keys mapping feature allows you to create Compound Keys, use existing keys etc.



3. Select the first *Genre ID* item and confirm **OK**.
4. **Result:** There is only one *Genre FK* in the *Film* attribute.



**i** TIP: To restore the original foreign key later, go to the **Relationship Properties**, double-click the *genre\_id* item in Child column and select *genre\_id (New)*. This created a new *genre\_id* Foreign Key instead of mapping it to the already existing *genre\_id* attribute.

## Automatic FK Mapping

You can also set **Automatic FK Mapping** in **Settings | Options | Model | Physical Model**.

# Physical Model

General | **Workspace** | Shape | Note Line | Entity

Word Wrap for SQL Preview

Word Wrap for SQL, Before Script and After Script

Synchronize NotNull with Mandatory Parent

Allow Null Attributes in Keys

Alphabetic Order in Attribute Navigator List

Inverse Relation Name

Attribute Properties Propagation

Self Relation Attribute Name:

Self Relation Attribute Caption:

Relation Attribute Name:

Relation Attribute Caption:

Automatic FK Mapping:

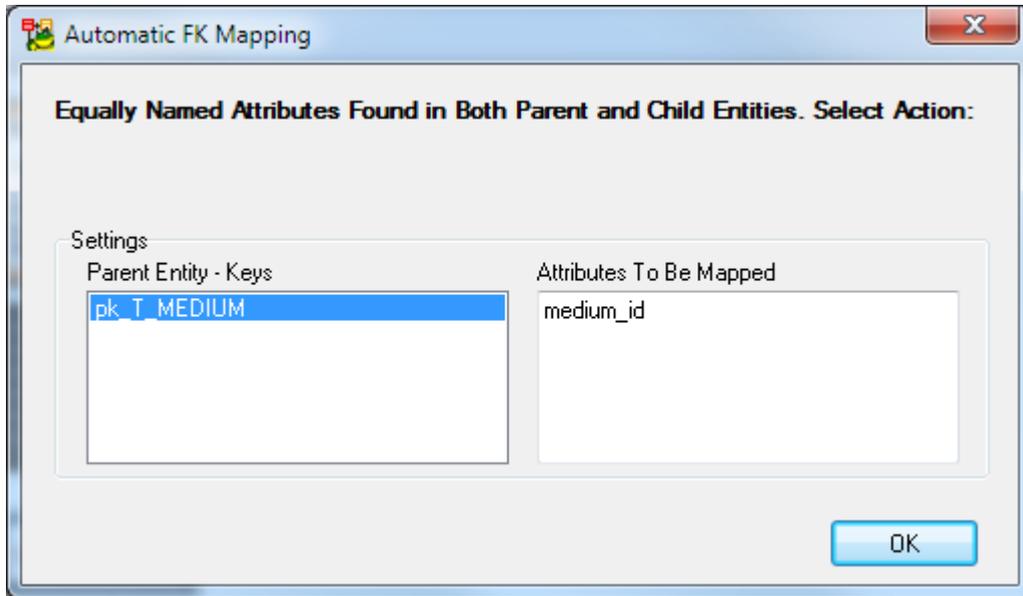
SQL script

Encoding Used for SQL Scripts

Code Editor Type

External Editor for generated Code

- **Disable** - FK mapping is OFF
- **Enable** - Toad Data Modeler searches for matching attributes. If a single possible mapping option is found, the FK is mapped. If multiple possible mapping options are found, a dialog appears where you choose the attribute the FK should be mapped to.
- **Always Show Dialog** - Displays a dialog whenever there is one or more possible mapping options. You get to choose the attribute the FK should be mapped to.



## Indexes

### To create an index

- Right-click an entity on Workspace and choose **Add | Index**. The **Index Properties** dialog opens.

or

- In the **Entity Properties** form, select the **Indexes** tab and click **Add**.

### To edit indexes

- In the **Entity Properties** dialog | **Indexes** tab, double-click the index or **Edit**.

Option	Description
	Above the Object Navigator Dropdown Menu, you can see name of entity that the index belongs to. Click the button on top right-hand corner to open the parent form (Entity Properties form).
Object Navigator Dropdown Menu	All indexes of selected entity are listed here. The box allows you to edit indexes quickly and comfortably from one place. Use buttons to change order of indexes. Use Ctrl + Up to move index upwards or Ctrl + Down to move it downwards.

Option	Description
<b>General Tab</b>	<b>Description</b>
Caption	Logical attribute name.
Name	Physical attribute name.
Schema	Schema selection.
Unique	Select this option to set the index as unique. Via unique indexes, you can link entities together. See <a href="#">Select Parent Key for Relationship</a> for more information on available linking methods.
Bitmap Index	Database dependent item (Oracle). Select this checkbox to define the index as bitmap index.
Generate	Select it to generate the index in final SQL (DDL) script. (It is selected by default.)

**i** Note: Other options on the **General** tab vary according to the database platform you're using. Options specific to your database can be found in the "Databases" chapter.

Items Tab	Option
Available	A list of all attributes of the entity.
Selected	Attribute(s) that have been assigned to the index.
<b>Notes Tab</b>	Tab for notes on the index.

Index Properties Tab	Description
Tablespace	Select a tablespace or click the button on the right to define a new tablespace.
Indextype	Index type definition
ODCI Parameters	ODCI parameters definition

### To display indexes on the Workspace

Right-click the WS, select **Workspace Format | Entity** tab and select the **Display Indexes** checkbox. See how indexes are displayed:

T_CUSTOMER
customer_id NN (PK)
name NN (IX1,IX2)
address NN (IX2)
i_name (IX1)
i_name_address (IX2)

**i** Note: Even when the indexes are not displayed, you can see which attribute belongs to which index (e.g. attribute *Name* is assigned to indexes *i\_name* and *i\_name\_address* - (IX1,IX2))

### To delete an index

In the **Entity Properties** dialog select the **Indexes** tab, choose the index and click **Delete**.

## Check Constraints

Check constraints can be created in the **Check Constraints** tab in **Entity Properties** form (for multiple column check constraints) or in the **Attribute Properties** dialog (for single column check constraint).

### To add a check constraint

In **Entity Properties** form, select the **Check Constraints** tab and click **Add**.

### To edit a check constraint

In **Entity Properties** form | **Check Constraints** tab, double-click the selected check constraint or press **Edit**.

General Tab	Description
Caption	Logical check constraint name
Name	Physical check constraint name
Check Constraint Rule	Select rule or click the button on the right to define a new rule.
Generate	Select it to generate the check constraint in final SQL (DDL) script (selected by default).
SQL Tab	Write SQL script for the check constraint here. See <a href="#">About Templates</a> for more information.
Notes Tab	Space for your notes on the check constraint.

**i** Note:

- To copy a check constraint, press CTRL and drag the constraint over the **Check Constraints** folder of a target entity in **Model Explorer**.
- To move a check constraint, drag it over the **Check Constraints** folder of a target entity in **Model Explorer**.
- To delete a check constraint, select it and click **Delete** in the **Check Constraints** tab of the **Entity Properties** form.

## Triggers

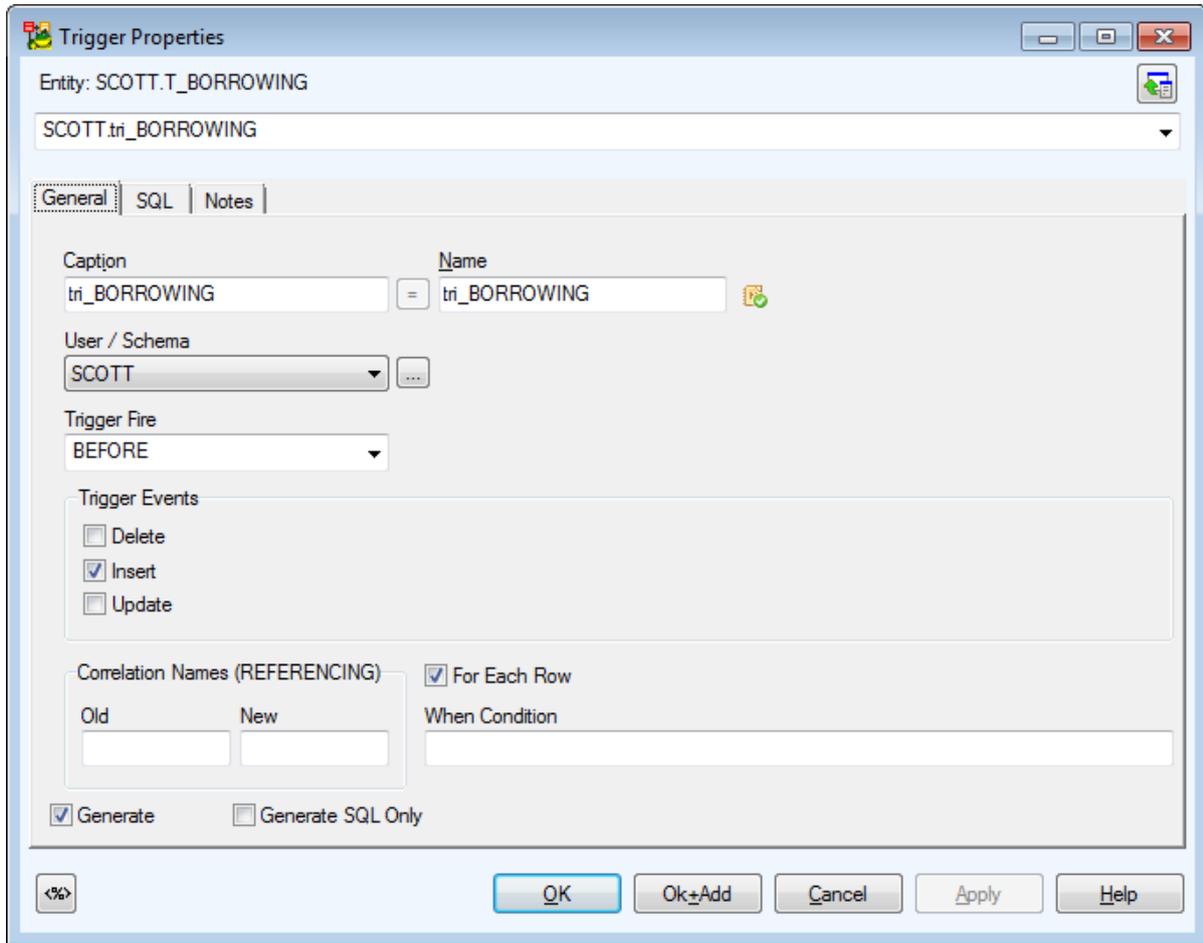
### To add a trigger

In **Entity Properties** form, select the **Triggers** tab and click **Add**.

### To edit a trigger

In **Entity Properties** form, **Triggers** tab, double-click the selected trigger or press **Edit**.

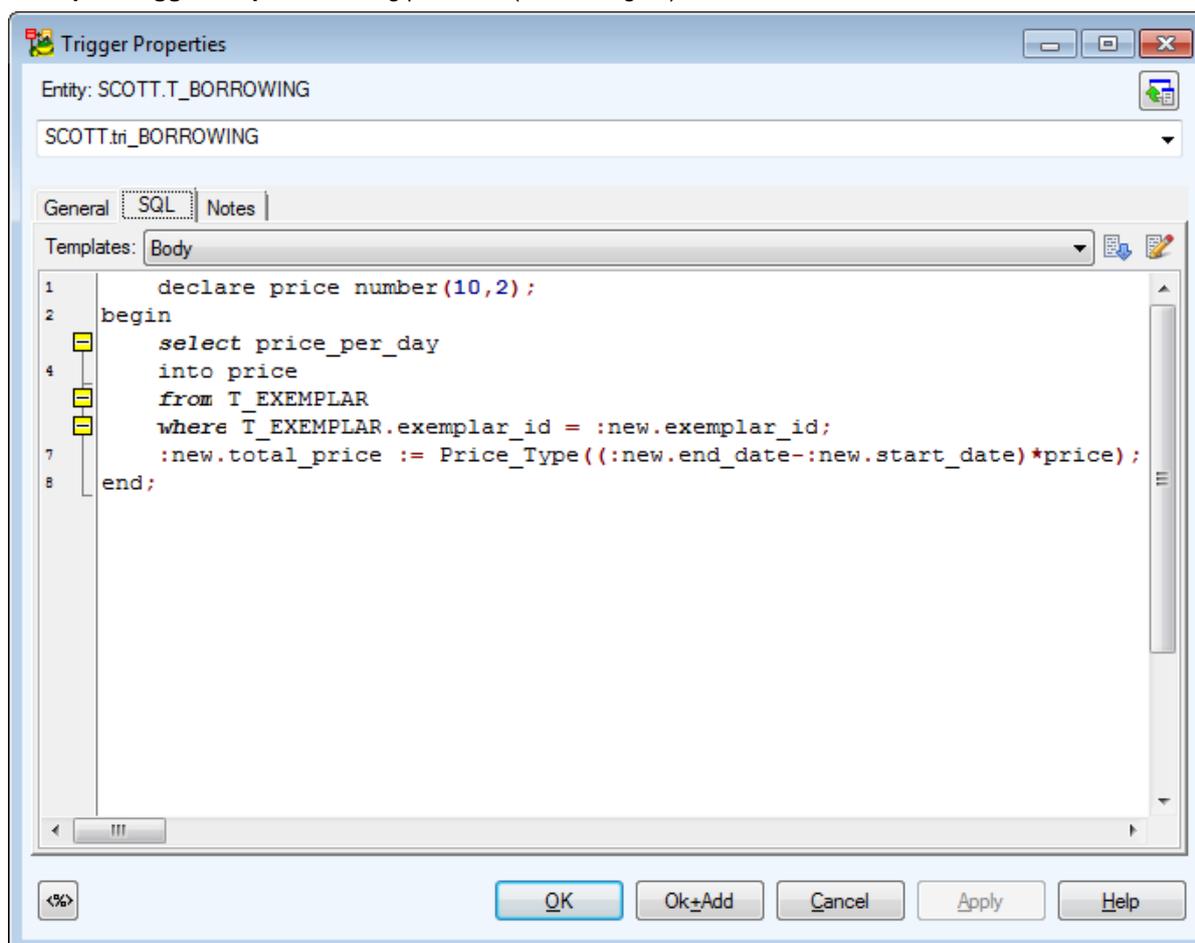
**Example: Trigger Properties** dialog (Oracle 10g db)



Option	Description
	Above the Object Navigator Box, you can see name of entity the trigger belongs to. If you click the button in top right-hand corner, the parent form will open ( <b>Entity Properties</b> in this case).
<b>General Tab</b>	<b>Description</b>
Caption	Logical trigger name
Name	Physical trigger name
Schema	Schema selection box
Trigger Fire	<b>Before, After</b> (database dependent) - select a trigger fire.
Trigger Events	<b>Delete, Insert, Update</b> - select a trigger event.

Option	Description
Generate	Select to generate the trigger in final SQL (DDL) script (selected by default.)
Generate SQL Only	Select to generate the SQL code written in tab <b>SQL</b> only.
<b>SQL Tab</b>	Write SQL script for the trigger here. <a href="#">About Templates</a>
<b>Notes Tab</b>	Space for your notes on the trigger.

**Example: Trigger Properties dialog | SQL tab (Oracle 10g db)**



**i** Note:

- To copy a trigger, press CTRL and drag the trigger over the **Triggers** folder of a target entity in **Model Explorer**.
- To move a trigger, drag it over the **Triggers** tab (folder) of a target entity in **Model Explorer**.
- To delete a trigger, select it and click **Delete** on the **Triggers** tab in the **Entity Properties** form.

## Permissions

In Toad Data Modeler, you can assign permissions to the following objects:

- Entity
- Attribute
- User Data Type
- View
- Procedure
- Schema
- Users and User Groups.

This list is dependent on your current database platform and version. For example, some databases do not support assigning permissions to Users.

For every object, different permissions can be set (SELECT, INSERT, UPDATE etc.), depending on current database platform.

Options for permissions are described in the following example. Permissions for attributes, user data types etc. are set in the **Properties** dialog of particular object | **Permissions** tab (e.g. **Attribute Properties** | **Properties**).

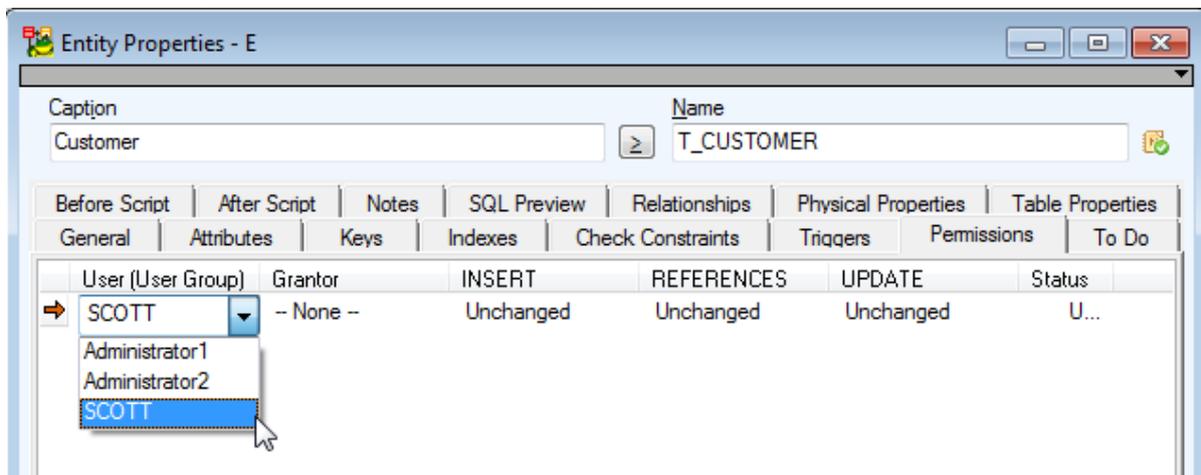
### Add Permissions

To be able to add a Permission, **it's necessary to define a User or User Group**. In case you forget, Toad Data Modeler allows you to define them directly from any **Permissions** tab of a particular object - click **Users** or **User Groups**.

#### To add a permission for entity

In the **Entity Properties** form, select the **Permissions** tab and click **Add**.

**Example:** Creating permission for user **SCOTT** in **Entity Properties** form:



If you want to change the permission User or User Group, select the permission **User (User Group)** column, **press F2** and choose from the list.

**i** TIP: This kind of editing properties is usable anywhere in Toad Data Modeler. For more information, see [Inplace Editor](#).

Permissions Tab	Description
User (User Group)	Name of user (group) that the permission has been assigned to.
Grantor	Name of user (group) that assigns the permission.
Permissions: SELECT, INSERT, UPDATE, DELETE, RULE, REFERENCES, TRIGGER	

### To edit a permission

In the **Entity Properties** form | **Permissions** tab, double-click the selected permission or press **Edit** .

General Tab	Description
Permissions	List of all available permissions to a specific object.
Status	Shows if the particular permission has been assigned or not. <b>Unchanged</b> - No change has been made. <b>Grant</b> - Permission has been granted. <b>Deny</b> - Permission has been denied. (E.g. in Microsoft SQL 2005 models.)
with Grant Option	<b>Yes/No</b> - Determines if the permission User (Group) can assign the permission to another User (Group).

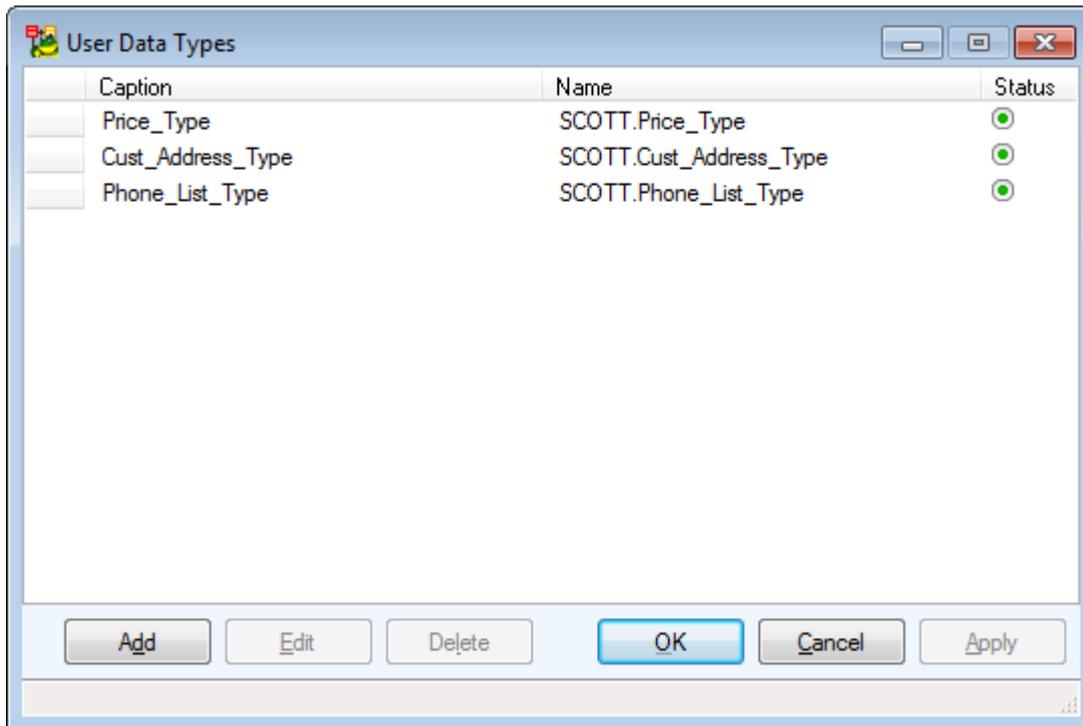
## User Data Types

In Toad Data Modeler, you can define your own data types (in case the database platform you use supports this feature).

### To add a user data type

Select **Model Menu** | **Model Items** | **User Data Types** and click **Add** in the **User Data Types** dialog.

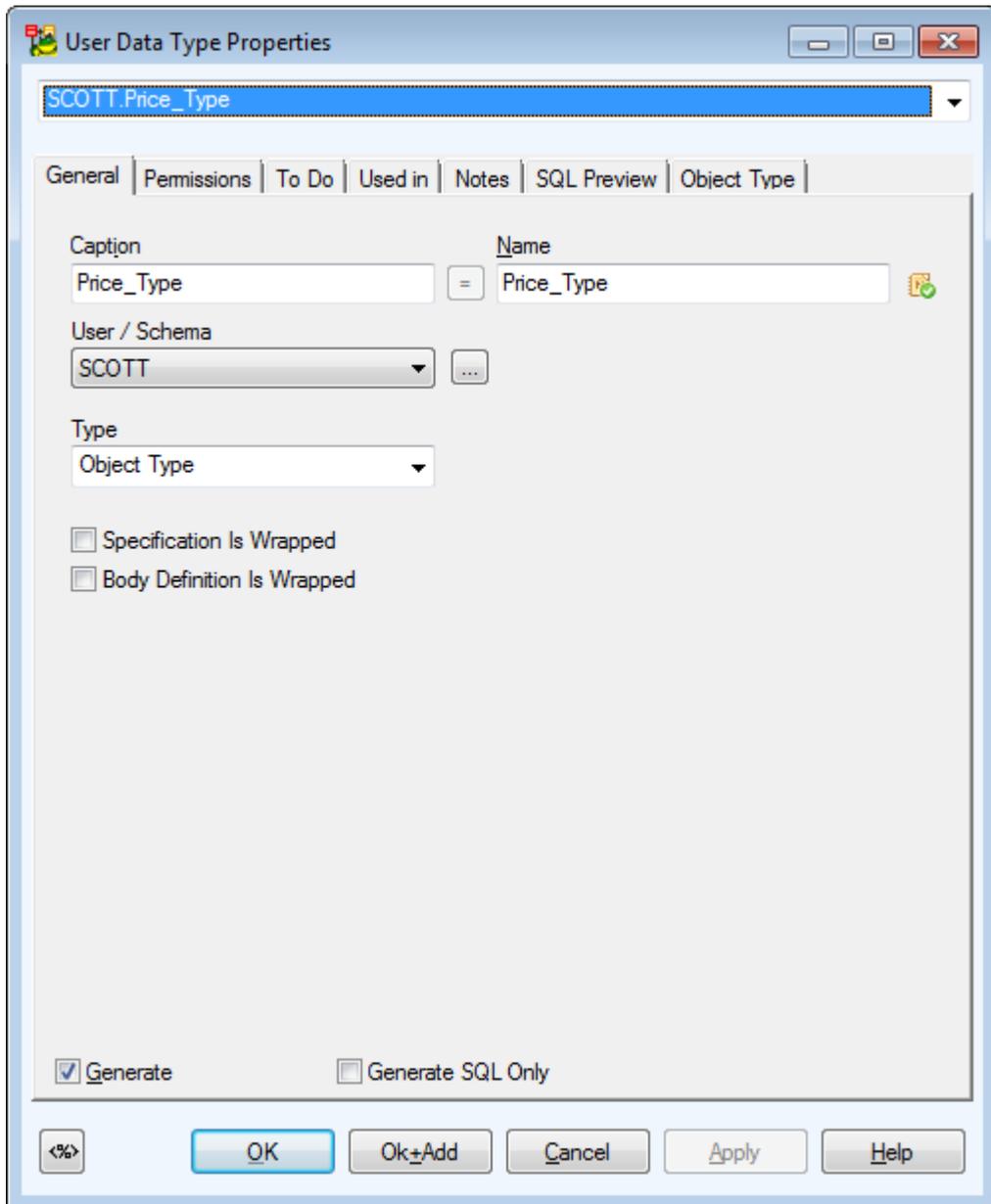
**Example:** User Data Types listed in **User Data Types** dialog



### ***To edit a user data type***

Double-click the selected data type or click **Edit** in the **User Data Types** dialog.

**Example:** The **User Data Type Properties** dialog (Oracle 10g)



General Tab	Description
Caption	Logical user data type name
Name	Physical user data type name
Schema	Schema selection box
Type	This option is available in Oracle models only. Contains specific types of user defined types.
Generate	Select this option to generate the user data type in final SQL script.

**Generate SQL Only** This option is available only for Oracle user data types. If this checkbox is selected, only the SQL code that you manually entered in appropriate tab (**Object Type** or **Varray Type** or **Nested Table Type**) will be generated in final DDL script.

**i** Note: All changes in your selected user data type will be automatically applied to all attributes of this data type.

**To Do Tab** On this tab, you can write some tasks related to the user data type.

**i** TIP: To see all To Do tasks in your entire model, select **Model Menu | To Do**.

**Used In Tab** All objects where the user data type is used are listed here.

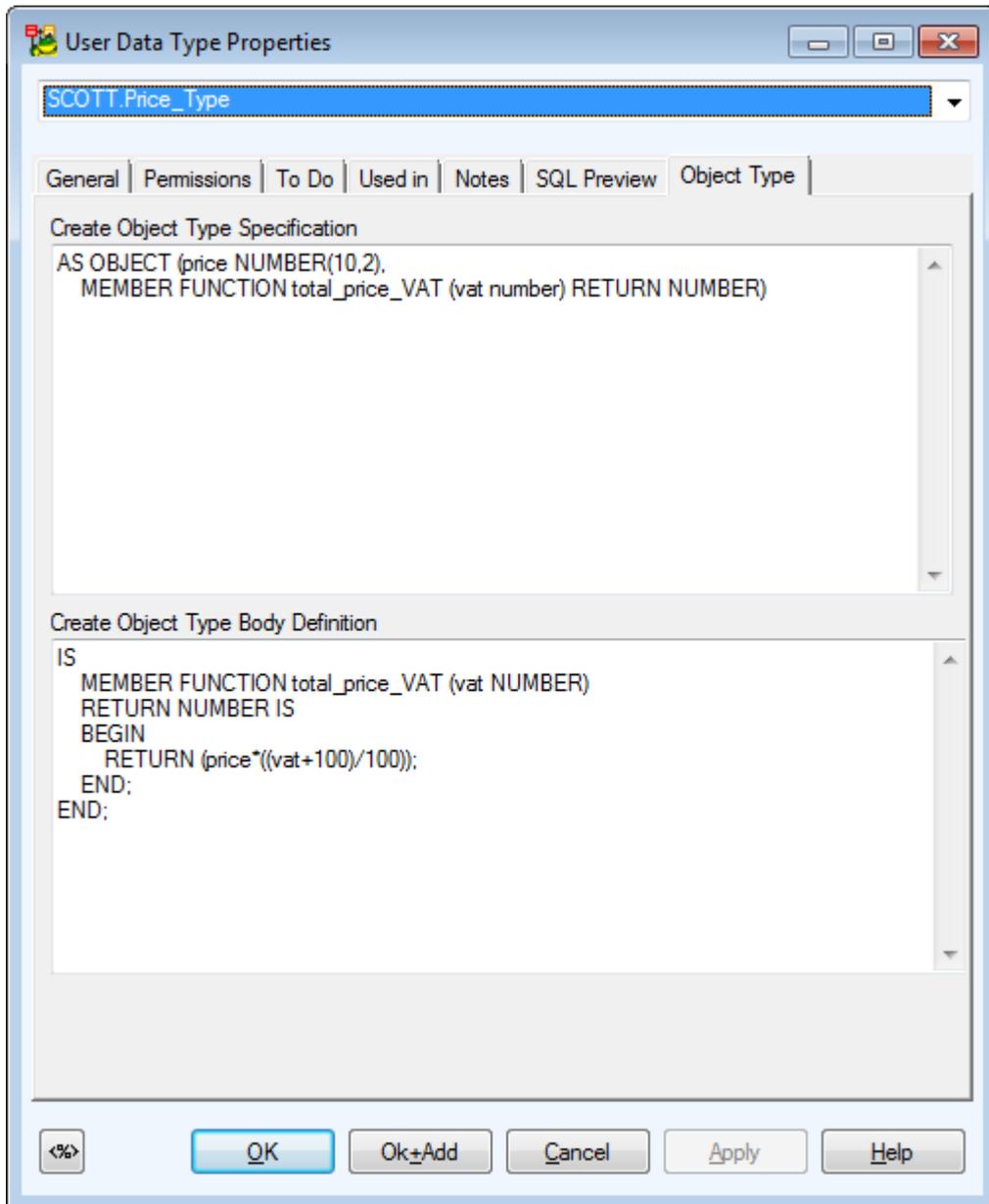
**Permissions Tab** On this tab, you can assign a User or User Group permissions for selected user data type.

**Notes Tab** Write notes related to the user data type.

**Object Type Tab** See example of specification and definition in the following screenshot.

**i** TIP: Click **OK+Add** to create another user data type.

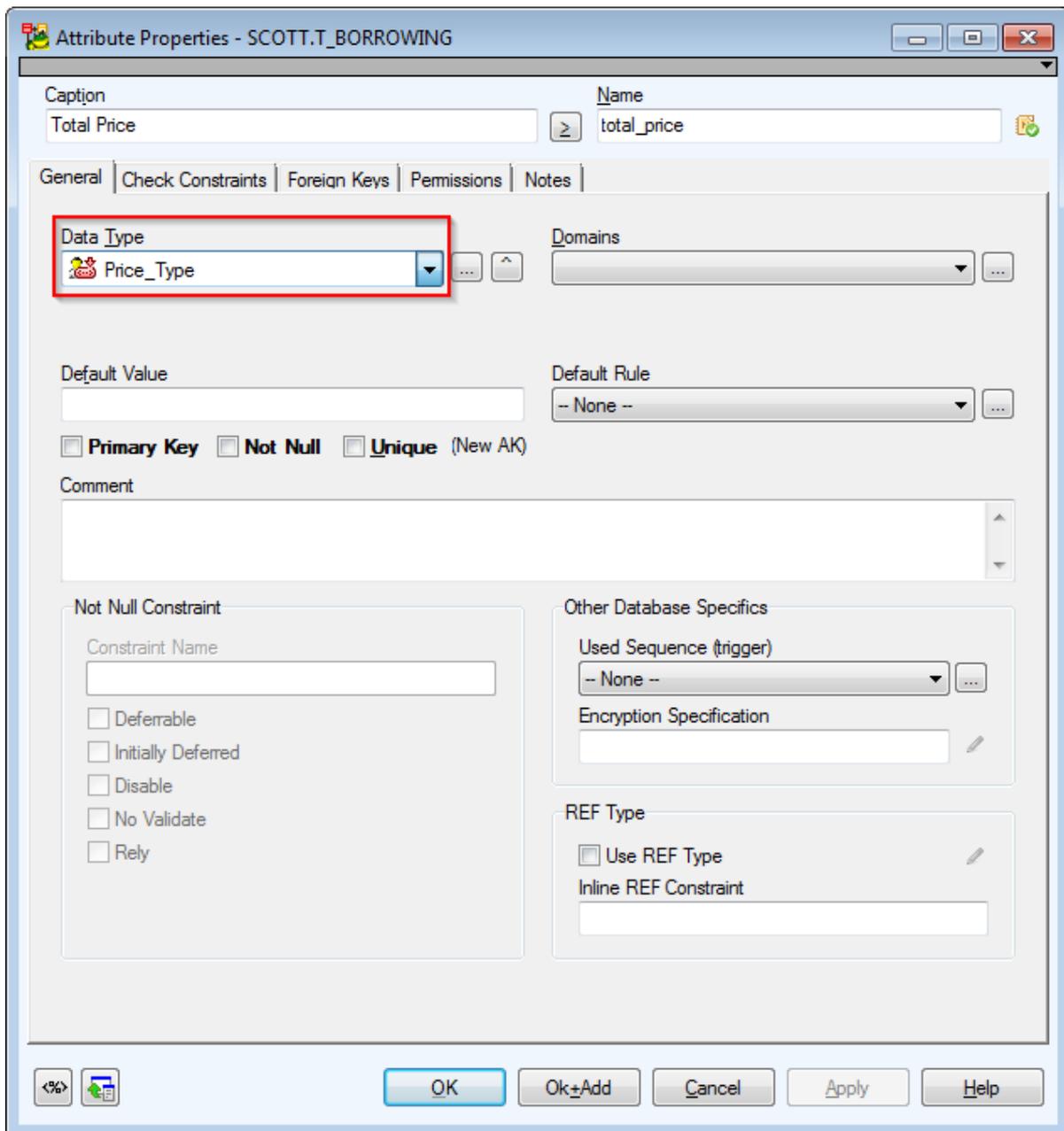
**Example:** User Data Type Properties dialog | **Object Type** tab (Oracle 10g)



### **To assign a User Data Type to an attribute**

1. Open the **Attribute Properties** dialog of the selected attribute | **General** tab.
2. From the **Data Type** box, select the required user data type.

**i** TIP: You can reach the **User Data Types** dialog from the **Attribute Properties** dialog - see the small icons next to the **Data Type** box.



**i** Note:

1. To copy user data types, use CTRL + Drag&Drop techniques.
2. To move user data types, use Drag&Drop techniques.  
You can copy and move your user data types:
  - In **User Data Types** dialog (**Model Menu** | **Model Items**)
  - Between **Model Explorer** and **User Data Types** dialog
3. To delete user data types:
  - Go to **Model Menu** | **User Data Types**, select a user data type and click **Delete**.
  - In **Model Explorer** | **User Data Types** folder Right-click and select **Delete Item**.

**i** Note: When you delete an user data type, the changes will reflect in all attributes that had this type assigned. There, the user data type will be replaced with data type that it was originally created from. To find out which attributes use a specific user data type, go to **User Data Type Properties** dialog | **Used in** tab.

## Dictionary Types

In Toad Data Modeler, you can define dictionary types. Dictionary is an alias of data type (must be supported by database).

**i** Note: Oracle databases support user data types only, Microsoft SQL Server supports both user data types and dictionary types.

### To add a dictionary type

Select **Model Menu** | **Model Items** | **Dictionary Types** and click **Add** in the dialog.

### To edit a dictionary type

Select **Model** | **Model Items** | **Dictionary Types** and double-click the selected dictionary type or click **Edit**.

General Tab	Description
Caption	Logical dictionary type name
Name	Physical dictionary type name
Data Type	Data Type selection box

**i** Note:

Other options on the **General** tab vary according to the current database platform. Options specific for your database can be found in the "Databases" chapter.

All changes in your selected dictionary type will be automatically applied to all attributes of the dictionary type.

### To Do Tab

You can enter To Do tasks related to the object here.

**i** Note: To display all To Do tasks, select **To Do** from **Model Menu**.

**Used In Tab** All objects using this dictionary type are listed here.

**Notes Tab** Tab for notes related to the dictionary type.

### **To select a dictionary type for attributes**

1. Open the **Attribute Properties** dialog of a selected attribute.
2. In **General** tab, select the dictionary type from the **Data Type** dropdown menu.

#### **i** Note:

1. You can reach the **Dictionary Types** dialog also from the **Attribute Properties** dialog - see the small icons next to the **Data Type** box.
2. Click **OK+Add** in the **Dictionary Type Properties** dialog to apply changes and create another dictionary type at once.
3. To copy dictionary types, use CTRL + Drag&Drop techniques.
4. To move dictionary types, use Drag&Drop techniques.  
You can copy and move your dictionary types within a model and between models of the same and different databases:
  - In the **Dictionary Types** dialog (**ModelMenu**)
  - Between **Model Explorer** and the **Dictionary Types** dialog
5. To delete dictionary types, select:
  - **Model | Dictionary Types** | Select a dictionary type and click **Delete**.
  - **Model Explorer | Dictionary Types** folder | Right-click and select **Delete Item**.

**i** Note: When you delete a dictionary type, it will be also removed from all attributes in which it was used and replaced with the data type that it was based on (e.g. deleting dictionary type based on char data type will cause all attributes to revert back to the char data type). To find out which attributes will be changed when you delete a dictionary type, open the **Dictionary Type Properties dialog | Used In** tab.

## **Domains**

Domain is another dictionary item in Toad Data Modeler. Domains only have logical meaning and are not generated. If a domain is used in a attribute, only the values of the domain are transferred to the attribute and are generated.

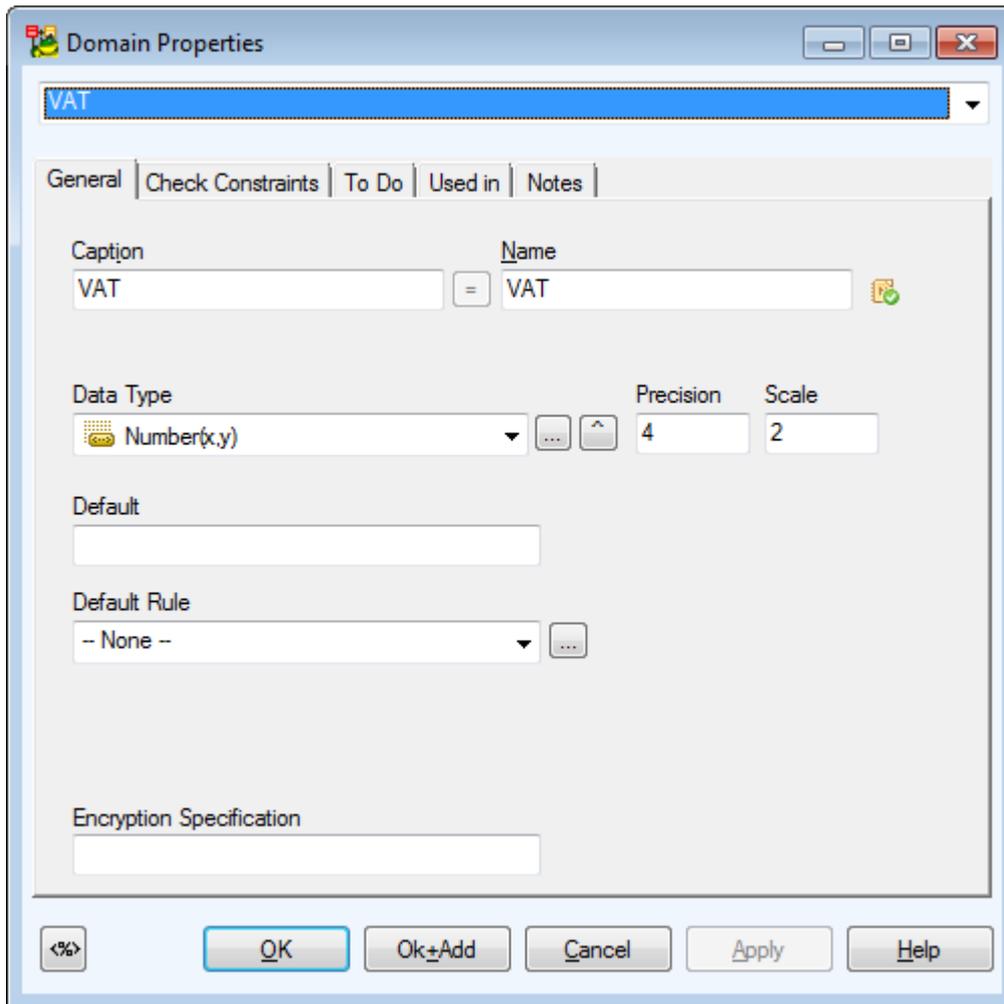
### **To add a domain**

Select **Model Menu | Model Items | Domains** and click **Add** in the **Domains** dialog.

### **To edit a domain**

Select **Model Menu | Model Items | Domains** and double-click the selected domain or click **Edit** in the **Domains** dialog.

**Example:** The **Domain Properties** dialog (Oracle 10g)



General Tab	Description
Caption	Logical domain name
Name	Physical domain name
Data Type	Data Type selection box



Note:

1. Other options on the **General** tab vary according to the current database platform. Options specific for your database can be found in the "**Databases**" chapter.
2. All changes made to your domain will be automatically applied to every attribute which uses the domain.

**Check Constraints Tab** On this tab, you can add, edit and delete check constraints of a domain.

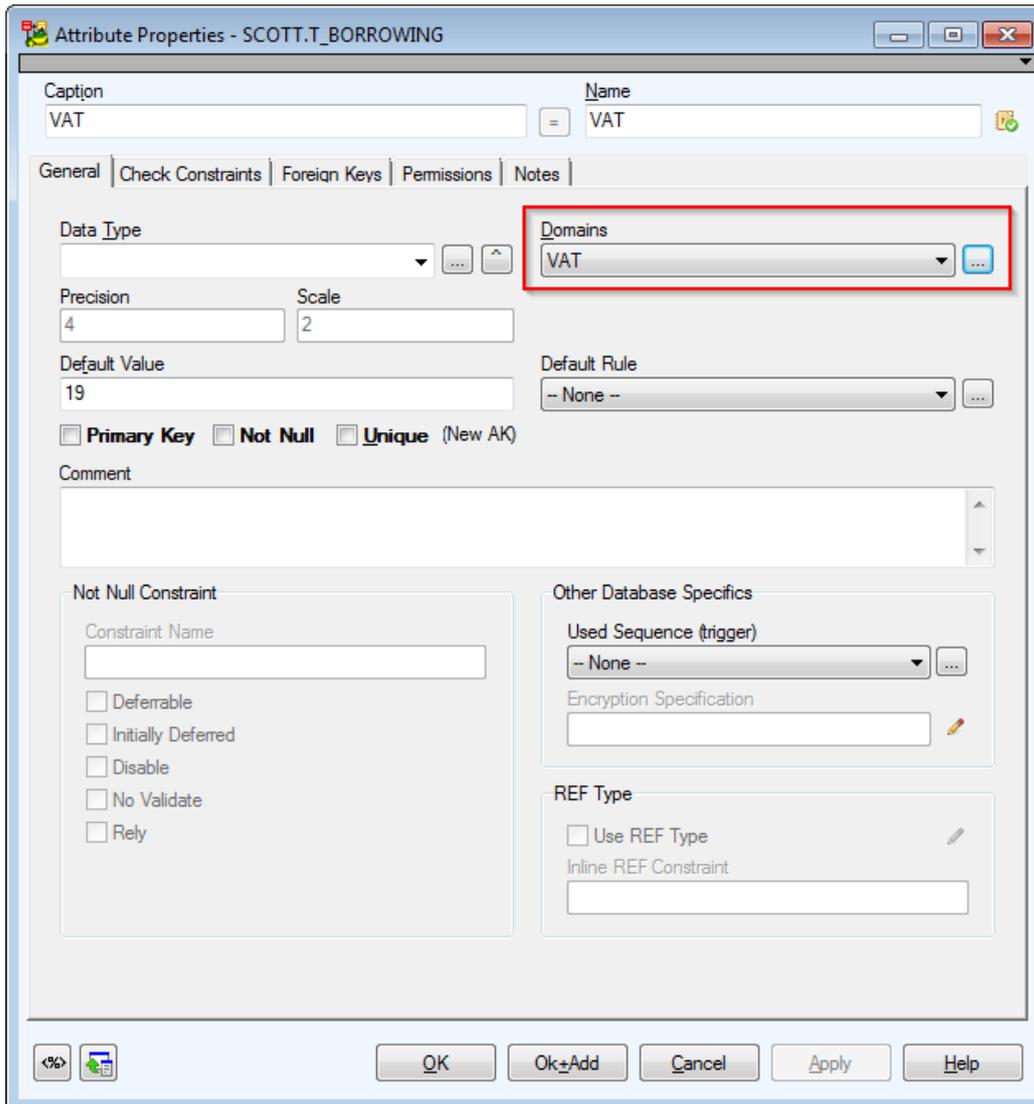
**To Do Tab** On this tab, you can write some tasks related to the domain.

**Used In Tab** All attributes with this domain are listed on this tab.  
Double-clicking any attribute will open its **Attribute Properties**.

**Notes Tab** Tab for notes on the domain.

### To select domains for attributes

1. Open the **Attribute Properties** dialog of the selected attribute | **General** tab.
2. From the **Domains** box, select a domain.



**i** Note:

1. To copy domains, use CTRL + Drag&Drop techniques.
2. To move domains, use Drag&Drop techniques.  
You can copy and move your domains within a model and between models of the same and different databases:
  - In **Domains** dialog (**Model** menu)
  - In **Model Explorer** | **Domains** folder
  - Between **Model Explorer** and **Domains** dialog
3. To delete domains, select:
  - **Model** | **Domains** | Select a domain and click **Delete**.
  - **Model Explorer** | **Domains** folder | Right-click and select **Delete Item**.

**i** Note: Deleting a domain will remove it from all attributes that are using it. To find out which attributes they are, open the **Domain Properties dialog** | **Used In** tab.

## Export/Import Dictionary

Toad Data Modeler allows you to use dictionary items also in other models. You can simply export all of them to the .TXI file, and then import them to any model at any time. You can save the .TXI file where you want, no default path is defined.

Dictionary items are:

- User Data Types
- Dictionary Types
- Domains

## Domains

They have only a logical meaning. They are not generated in DDL/SQL script. If a domain is used in attribute, only values of the domain are transferred to the attribute during the DDL script generation process.

## User Data Types

They are data types defined by users and can be generated in final DDL script. User data types are not derived from data types.

## Dictionary Types

They are data types that are derived from other data types. They can be generated in final DDL script.

## How to Export/Import Dictionary

### *You want to use dictionary items of Model A in Model B:*

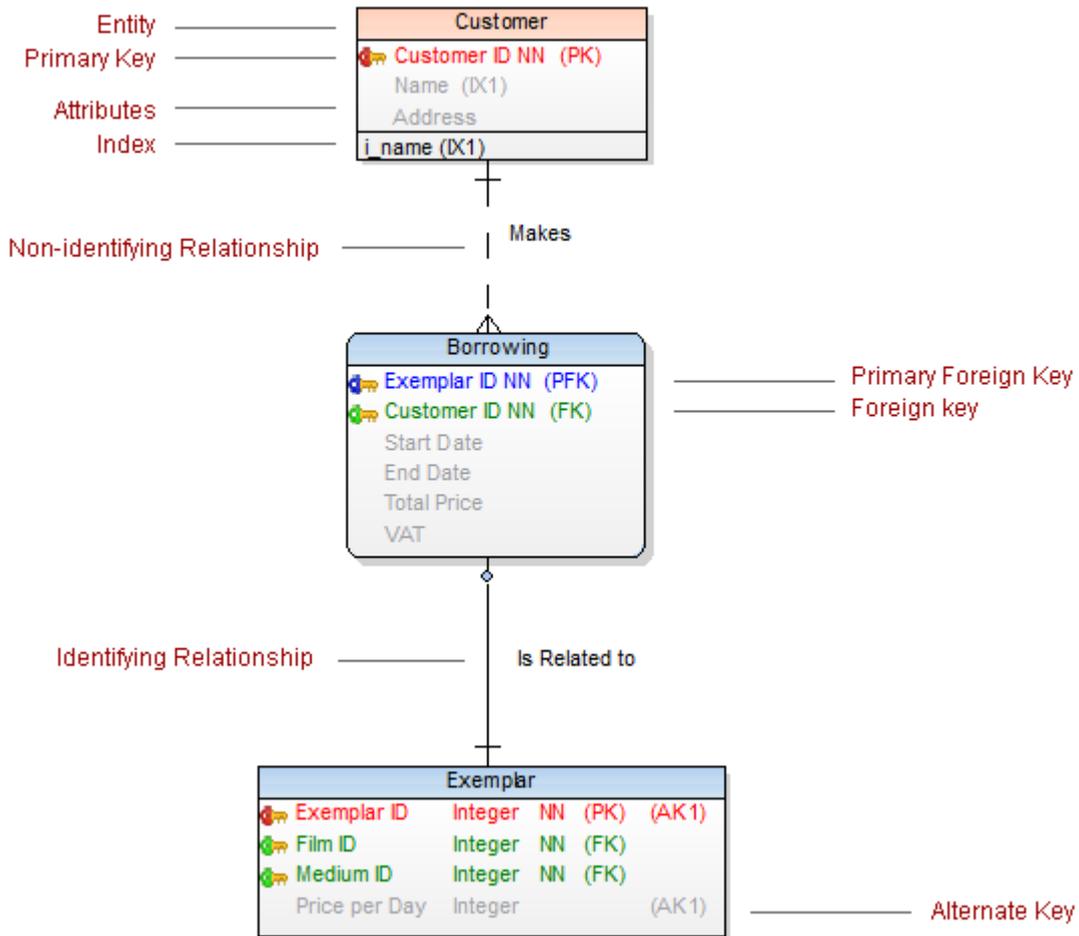
1. Open *Model A*.
2. Select **Model | Export Dictionary**.
3. Save the .txi file.
4. Open *Model B*.
5. Select **Model | Import Dictionary**.
6. Select the .txi file and click **Open**.

**i** Note:

- Domain Check Constraints are imported/exported too.
- It's not possible to make selection of the dictionary items for the import/export. All the dictionary items are always imported/exported at one jump.

## Notation and Cardinality

### IE Notation

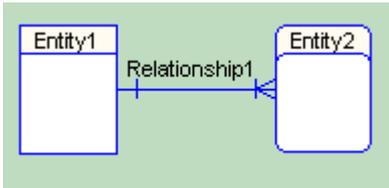


## Cardinality

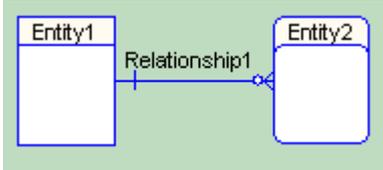
One-to-many relationship is represented by this symbol:

One-to-one relationship is represented by this symbol:

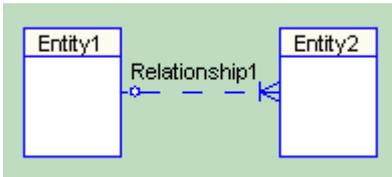
**Parent:** mandatory **Child:** mandatory



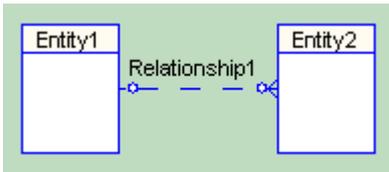
**Parent:** mandatory **Child:** optional



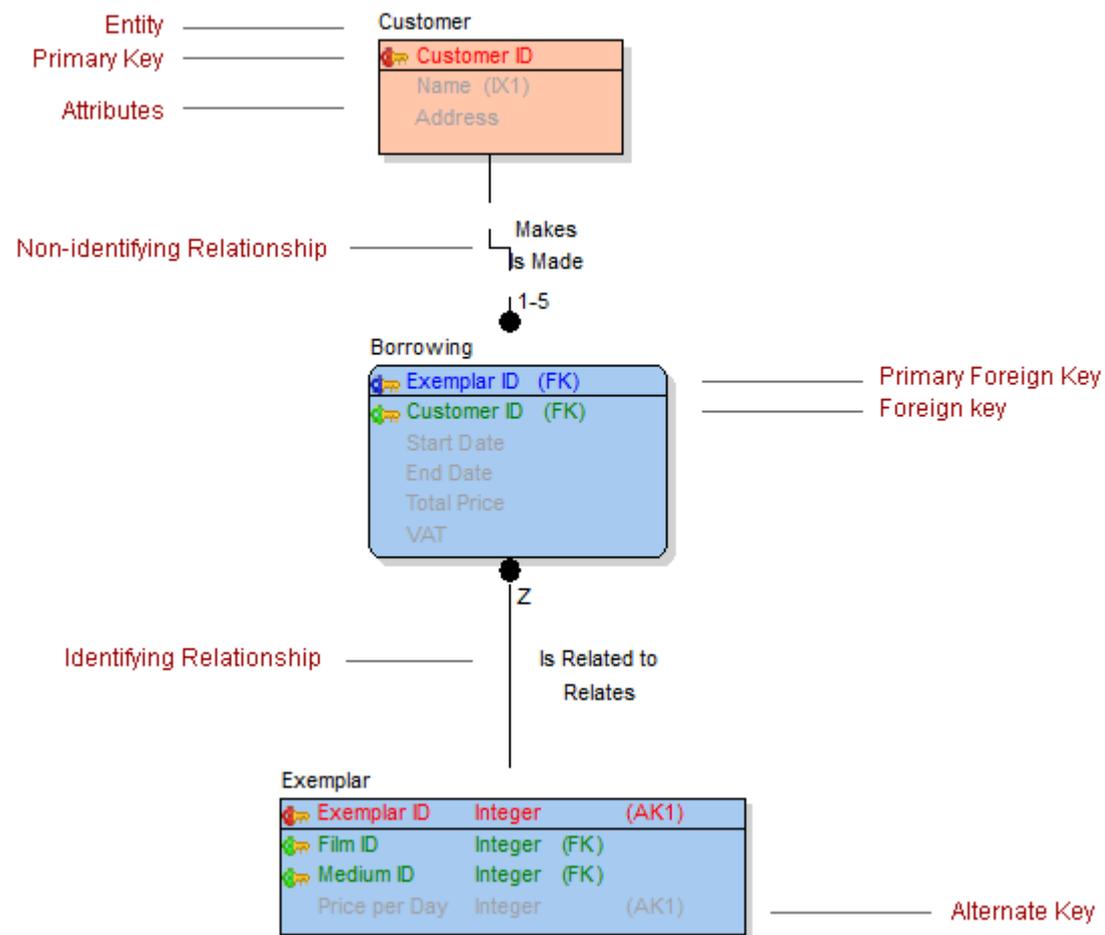
**Parent:** optional **Child:** mandatory



**Parent:** optional **Child:** optional



## IDF1X Notation



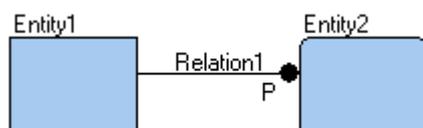
## Cardinality

zero, one or more

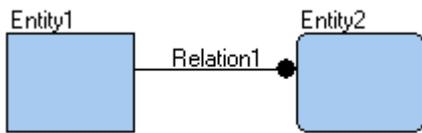
one or more

zero or one

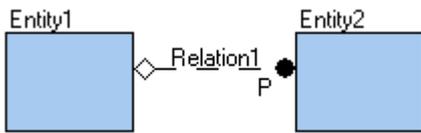
**Parent:** mandatory **Child:** mandatory



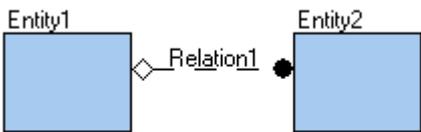
**Parent:** mandatory **Child:** optional



**Parent:** optional **Child:** mandatory



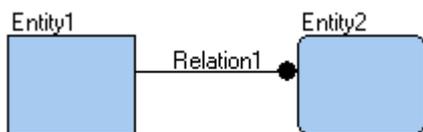
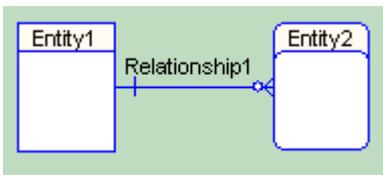
**Parent:** optional **Child:** optional



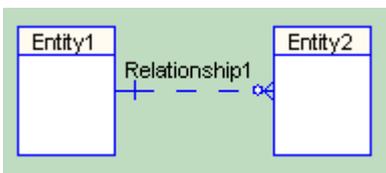
See [Synchronization of Not Null and Mandatory Parent](#) for more information.

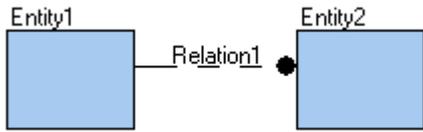
## Relationship Types

# Identifying Relationship

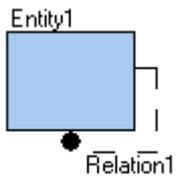
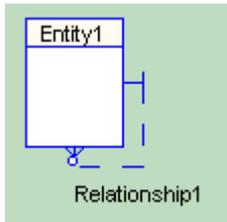


# Non-identifying Relationship

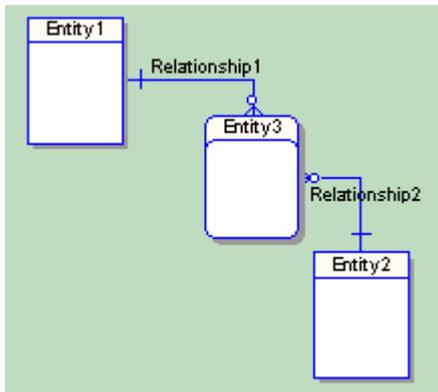


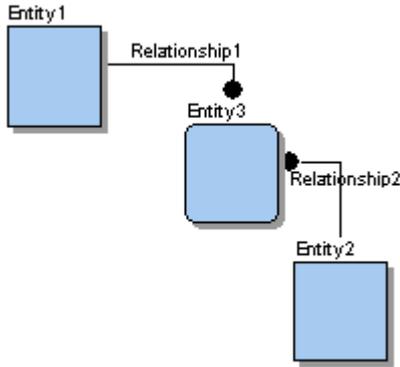


## Non-identifying Self-relationship



## M:N Relationship





## Relationship Types

Toad Data Modeler supports the following relationship types (physical model):

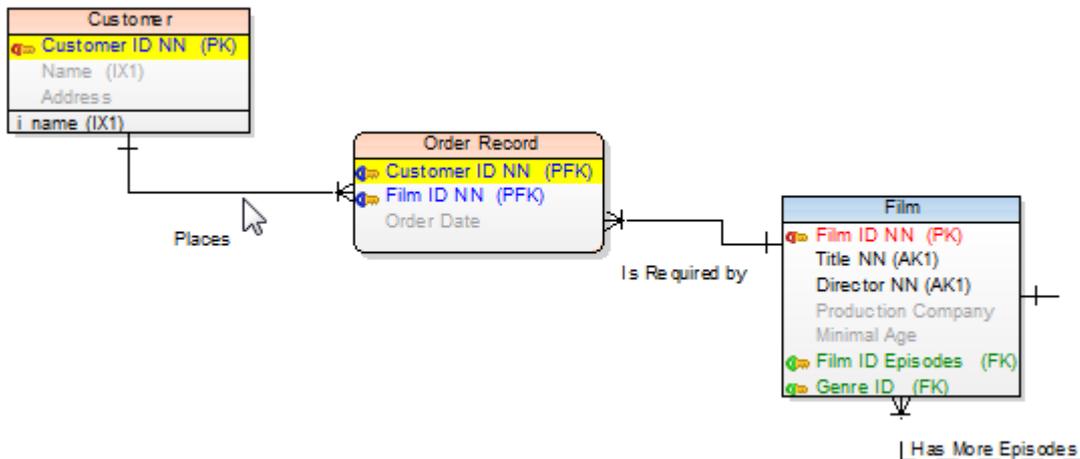
- Identifying
- Non-identifying
- Self-relationship for non-identifying relationship
- M:N relationship

## Identifying Relationship



Primary key migrates from parent entity to child entity and there becomes a part of the primary key. It is used when the primary key of the child entity is unable to provide definite identification.

An entity, connected with a parent entity through an identifying relationship, is called "dependent" entity and is shown in a model with rounded corners.

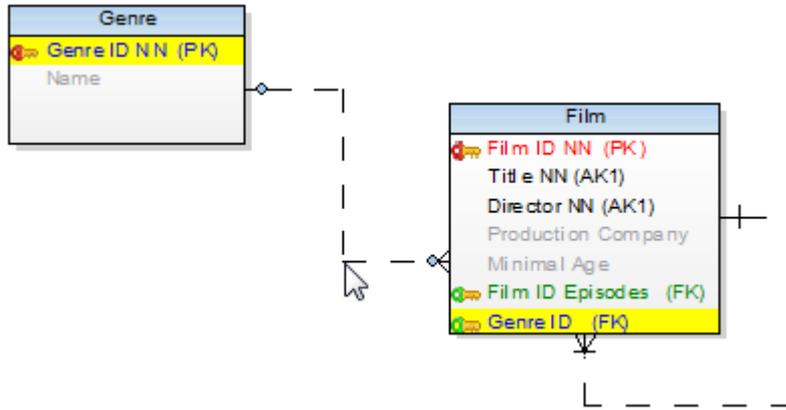


The Order Record entity cannot exist itself. It is dependent on entities Customer and Film. Therefore the Identifying relationship is used. The Order Record entity is a dependent entity, and the Customer ID and Movie ID items are its unique record identifiers.

## Non-Identifying Relationship

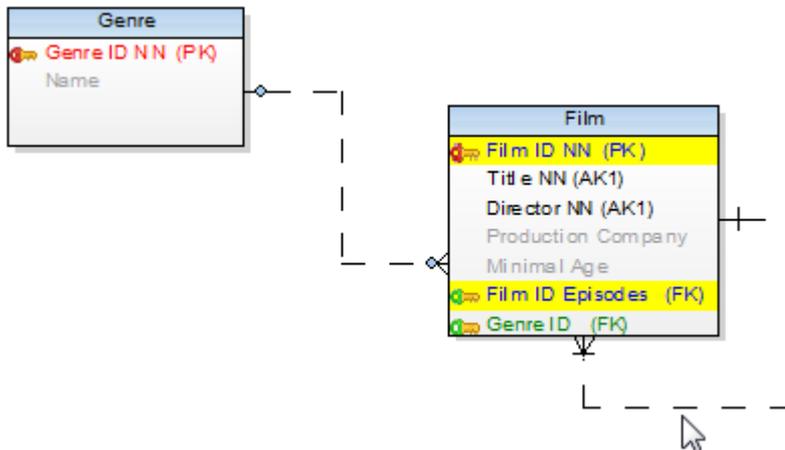


Primary key migrates from parent entity to child entity and does not become a part of the primary key. Non-identifying relationships are represented by dashed lines. In the dependent table, the attribute is referred to as a foreign key.



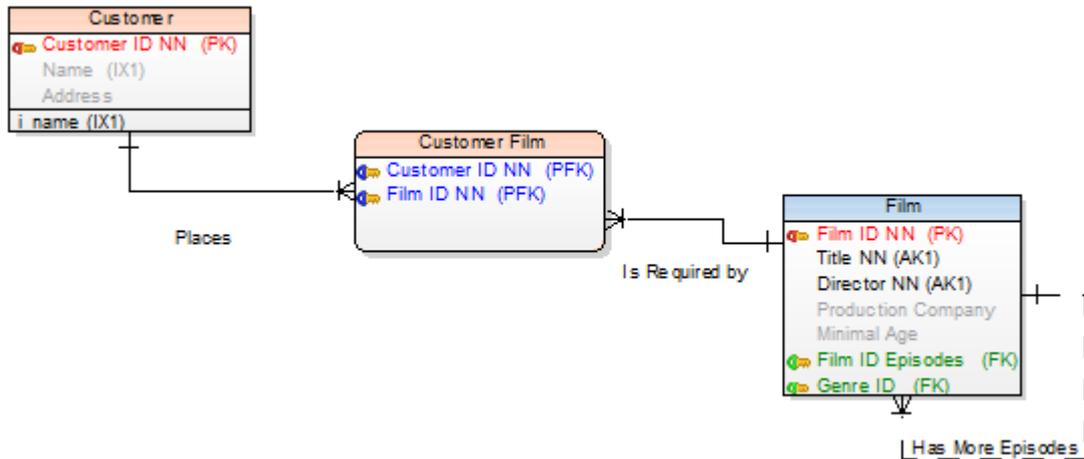
The *Film ID* as the unique identifier for Film is sufficient. Therefore the non-identifying relationship is used. The *Genre ID* is only a foreign key. The film can exist without being assigned to a genre, therefore the Film entity is an Independent entity.

## Self-Relationship for Non-identifying Relationship



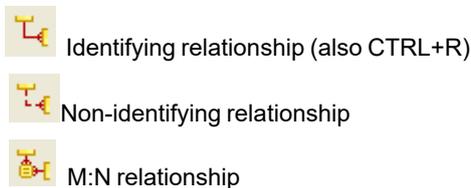
## M:N Relationship





## Create Relationships

1. Select a relationship type - click the appropriate relationship icon on the toolbar:



2. Move your mouse cursor over the work area. (The cursor changes its appearance.)
3. Click the first entity (parent) and then the target entity (child).

**i** TIP: Hide relationship names:

1. Right-click the Workspace and select **Workspace Format**.
2. In the **Workspace Format** dialog | **General** tab, select **Hide Line Captions** checkbox.

### To create self-relationship for non-identifying relationship

1. Click the Non-identifying relationship icon on the toolbar .
2. Move your mouse cursor over the work area.
3. Double-click the selected entity on the Workspace.

**i** TIP: Before you create a self-relationship, select **Settings | Options | Physical Model | Self Relation Attribute Name/Caption** and define a name for propagated attributes there (e.g. via prefix, suffix, application variable.)

### To add multiple relationships

1. Press SHIFT and click the **Relationship** icon. See the blue frame in the icon now. 
2. Create as many relationships in you model as you need.

- Right-click the work area (or click the **Relationship** icon again) to turn this function off.

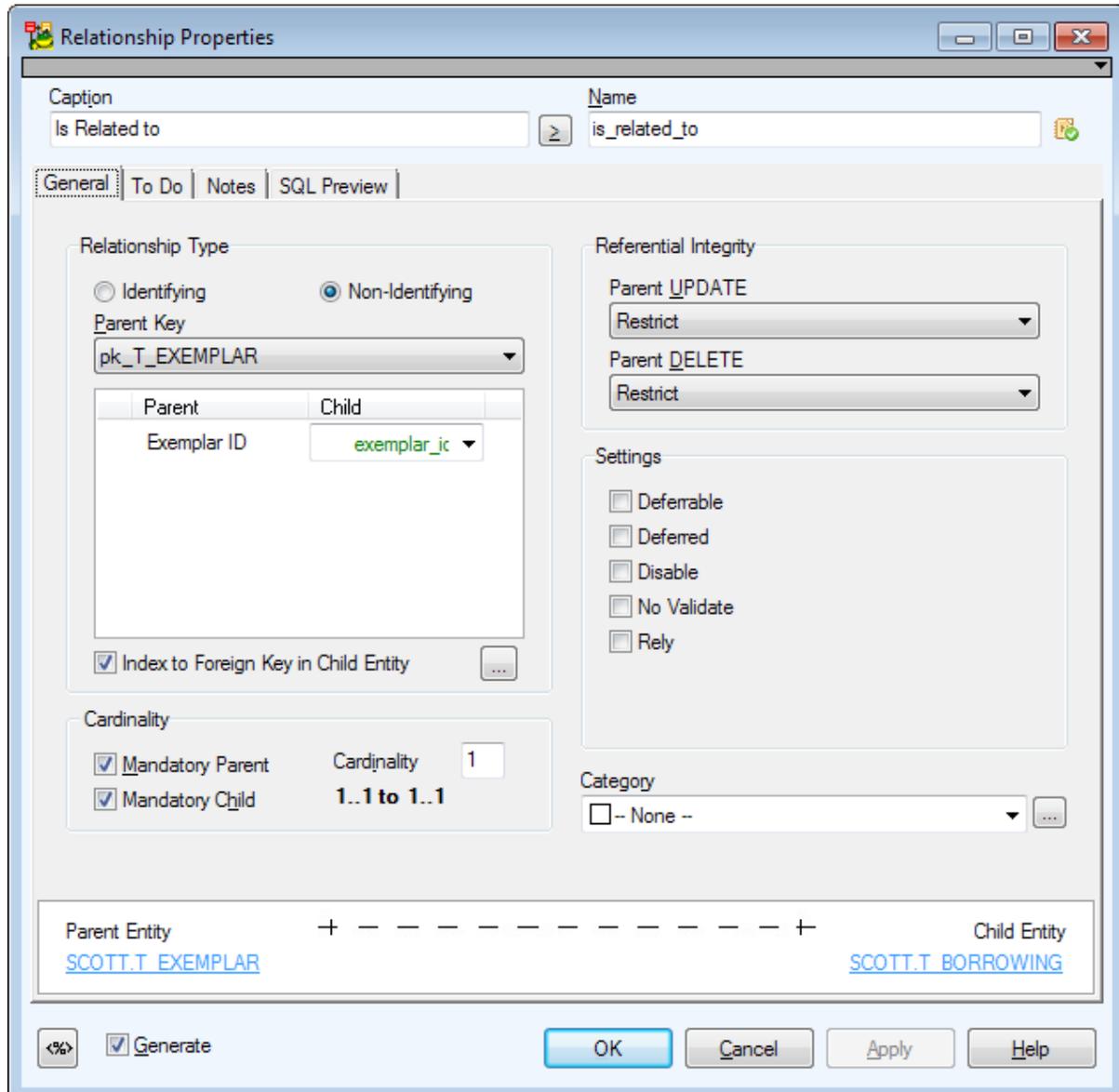
## Edit Relationships

Double-click the relationship on the Workspace.

or

Edit the relationship in **Model Explorer | Relationships** folder | double-click the selected relationship (or right-click | **Edit**).

**Example:** The **Relationship Properties** dialog (Oracle 10g db)



**i** TIP: When you point your mouse cursor at a relationship line, parent and child attributes are highlighted on the Workspace. You can set the color at: **Settings | Options | Graphics | Colors** area | **Highlight Color**.

Option	Description
Caption	Logical name of relationship
Name	Physical name of relationship <a href="#">About Naming Conventions</a>
<b>General Tab</b>	<b>Description</b>
Relationship Type	Select this option to set the relationship as identifying or non-identifying. There are also several setting related to Parent and Child entities. The last checkbox gives you the option to create Index to Foreign Key in Child Entity.
Parent Key	Linking Method selection box. Available options: primary key, alternate key, unique item. <a href="#">Select Parent Key for Relationship</a>
Referential Integrity	<a href="#">Referential Integrity</a>
Cardinality Area	Cardinality settings definition. Toad Data Modeler allows you to set up synchronization of NotNull and Mandatory Parent. <a href="#">Synchronization of Not Null and Mandatory Parent</a>
Settings	Database dependent options (Oracle in our example).
Parent Entity Name and Child Entity Name	Names of entities that the relationship connects. Click the buttons next to these boxes to open the <b>Entity Properties</b> form of appropriate entities.
<b>To Do Tab</b>	On this tab, you can write some tasks on the selected relationship. <b>Note:</b> To see all To Do tasks, select <b>Model   To Do</b> .
<b>Notes Tab</b>	Tab for notes on particular relationship. The text will display in a pop-up box when you point your mouse cursor at the relationship name in ER diagram.
<b>SQL Preview Tab</b>	Click  at the bottom of this tab to see the part of SQL code for the relationship. Select the <b>WordWrap</b> checkbox to wrap the code on this tab. <b>Tip:</b> To select this option by default, select <b>Settings   Options   Physical Model   General tab   Word Wrap for SQL Preview</b> .

## Buttons:



- opens the **Application Variables** form

**OK** - confirms all changes and closes the form

**Cancel** - cancels the changes you have made, and closes the form

**Apply** - confirms the changes (The form will remain opened for further edit.)

**Help** - opens the Help file

## Referential Integrity

*To define integrity rules of a relationship related to changing, adding or deleting a record inside parent or child entity*

Set values in **Referential Integrity** group box.

Relationship Properties

Caption: Makes      Name: makes

General | To Do | Notes | SQL Preview | Index to Foreign Key

Relationship Type

Identifying       Non-Identifying

Parent Key: pk\_T\_CUSTOMER

Parent	Child
Customer ID	customer_jr

Cardinality

Mandatory Parent      Cardinality: 5

Mandatory Child      1..1 to 1..5

Referential Integrity

Parent UPDATE: Restrict

Parent DELETE: Restrict

Settings

Deferable  
 Deferred  
 Disable  
 No Validate  
 Rely

Category: - None -

Parent Entity: SCOTT.T CUSTOMER      Child Entity: SCOTT.T BORROWING

Generate      OK      Cancel      Apply      Help

Option	Description
Parent Update	An event when the primary key of the record is going to be changed

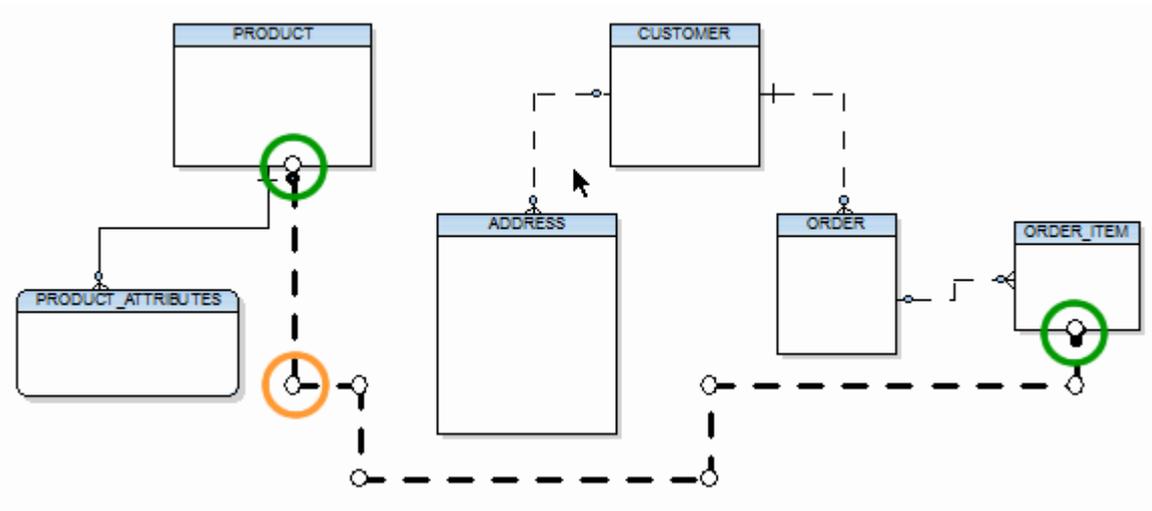
Option	Description
	<p>inside the parent entity. The rules are:</p> <p><b>None</b> No rule.</p> <p><b>Restrict</b> If the primary key is changed inside the parent entity record, and inside the child entity there are records allocated to the original parent record primary key, the change will not be made and database will report an error.</p> <p><b>Cascade</b> If the primary key is changed inside the parent entity record, and inside the child entity there are records allocated to the original parent record primary key, the database will change appropriate foreign keys inside the child entity.</p> <p><b>Set NULL</b> If the primary key is changed inside the parent entity record, and inside the child entity there are records allocated to original parent record primary key, the database will set appropriate foreign keys inside the child entity to NULL.</p> <p><b>Set Default</b> If the primary key is changed inside the parent entity record, and inside the child entity there are records allocated to the original parent record primary key, the database will set appropriate foreign keys inside the child entity to a default value.</p>
Parent Delete	<p>An event when a record inside the parent entity is being deleted. The rules are:</p> <p><b>None</b> No rule.</p> <p><b>Restrict</b> If a record inside the parent entity is being deleted, and inside the child entity there are records allocated to parent entity, the command won't be executed.</p> <p><b>Cascade</b> In case of record deletion inside the parent entity, the records inside the child entity will be deleted as well.</p> <p><b>Set NULL</b> In case of record deletion inside the parent entity, the foreign keys inside the child entity will be set to NULL.</p> <p><b>Set Default</b> In case of record deletion inside the parent entity, the foreign keys inside the child entity will be set to a default value.</p>

**i** | TIP: Define default values for referential integrity type. [Default Values](#)

## Edit Relationship Lines

Relationship lines consist of:

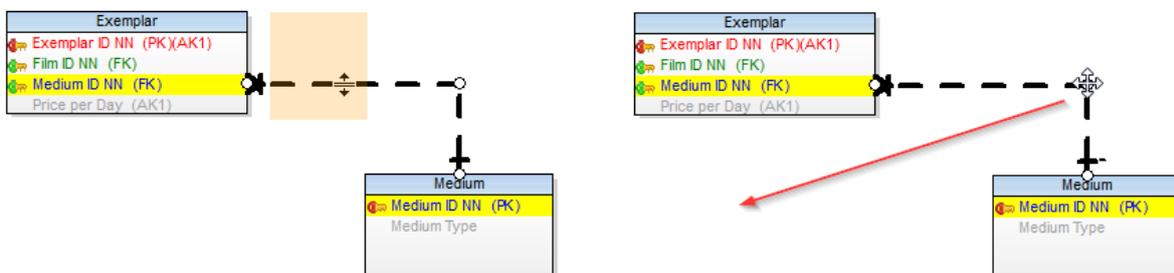
- **Lines**
  - Solid lines represent identifying relationships.
  - Dashed lines represent non-identifying relationships.
- **Anchor Points** - End points directly connected to entity boxes. You can alter them as you need.
- **Handle Points** - Other points that can be added on relationship lines via CTRL key. Handle points allow you to select a part of relationship line to move or delete.



## Move Lines, Handle Points and Anchor Points

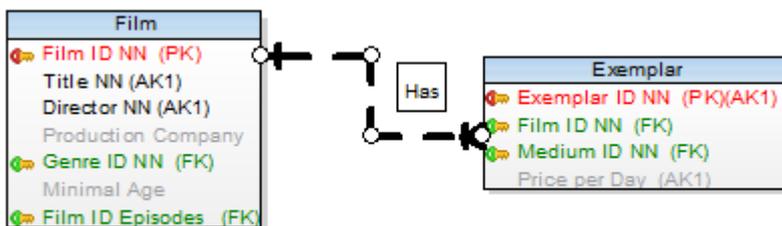
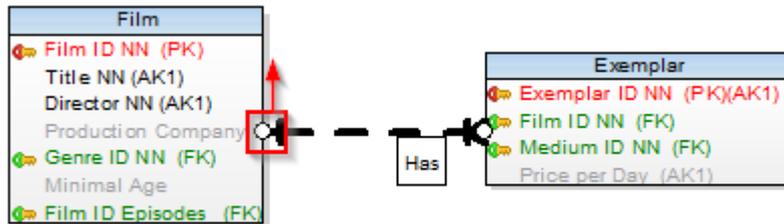
### To move part of line, handle point or anchor point

1. Select a relationship line.
2. Place mouse cursor over the line or point.
3. Use drag and drop technique to change position of the line or point.



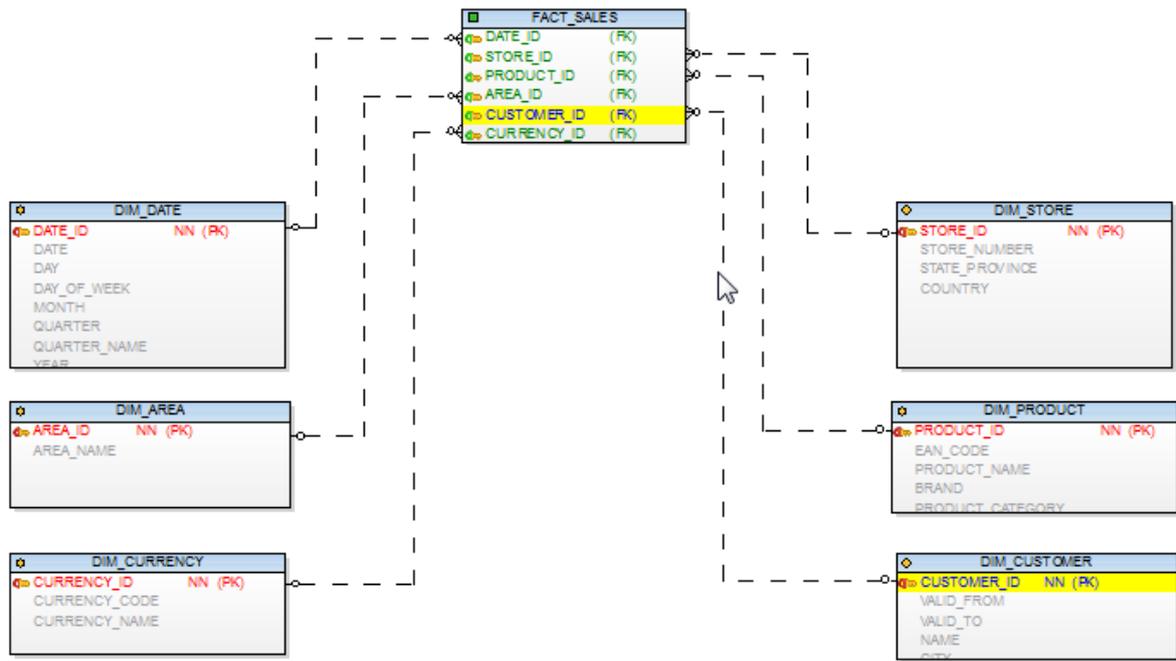
### To move single anchor point and break a line

1. Select a relationship line.
2. Place mouse cursor over the anchor point.
3. Use drag and drop technique to change position of the break point.



### Column to Column Alignment

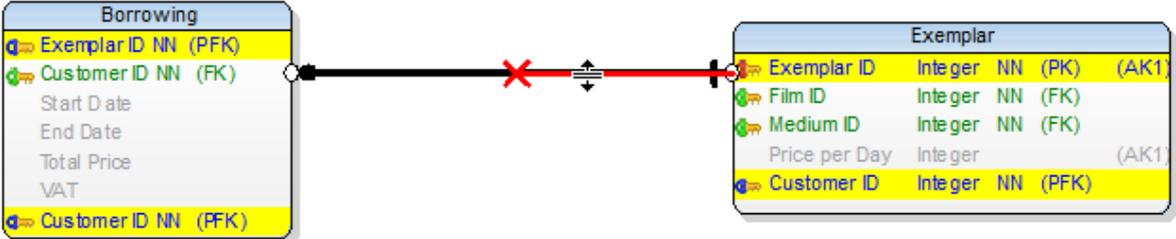
Lines can be rearranged manually and self-explanatory links from parent to child entities can be displayed in ER diagram.



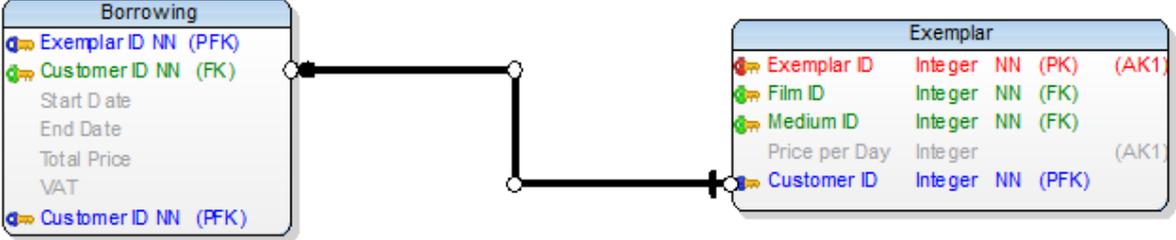
**Add Handle Points**

*To add handle point to a relationship line and break horizontal line:*

1. Click the selected relationship line.
2. Press CTRL and click the line. Red cross icon will show up. Release CTRL key.
3. Move your mouse cursor left or right and then move the highlighted part of the line up or down.

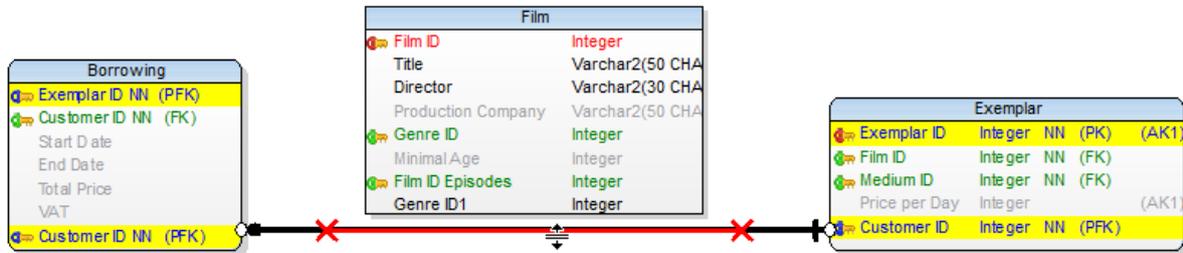


Result:

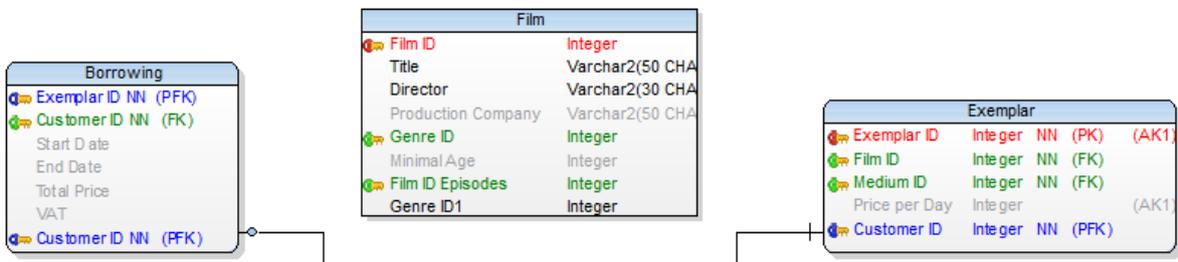


**To break horizontal line using two handle points:**

1. Click the selected relationship line.
2. Press CTRL and click the line. New red cross icon will show up.
3. Do the same for the second position.
4. Move your mouse cursor between the two handle points and then move the highlighted part of the line up or down.



Result:

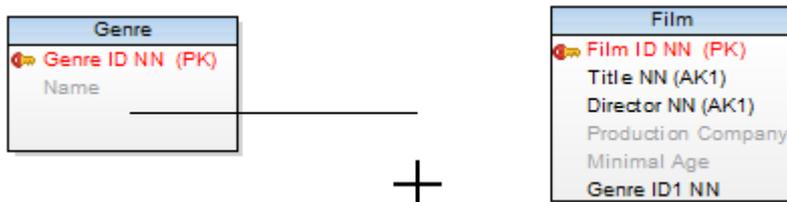


**Add Horizontal or Vertical Lines Only**

**To create horizontally or vertically straight lines**

(Entities are beside or below each other.)

1. Click the identifying or non-identifying relationship icon on the toolbar.
2. Click parent entity and press and hold SHIFT key. Wherever you move your mouse cursor, only horizontal or vertical straight lines are offered.
3. Move your mouse cursor to the desired position (over child entity)
4. Click the target entity and release SHIFT key.



**i** TIP: Using this method (via SHIFT key), you can lead the relationship as you need. E. g. you want to go around some entities in a particular way. If you just click the parent and child entity, the relationship will be created automatically and you will have to do some improvements later. To create your own track for the relationship:

1. Click the identifying or non-identifying relationship icon on the toolbar.
2. Click parent entity and press and hold SHIFT key. Wherever you move your mouse cursor, only horizontal or vertical straight lines are offered.
3. Move your mouse cursor where you need to lead the line. Click the WS where you need to break it (still holding the SHIFT key). Make as many break points as you need.
4. Finally, click the target entity and release SHIFT key.

## Move, Hide, Find Relationship Names

### ***To move the name of relationship on the Workspace***

Right-click the selected relationship in particular place and select **Move Caption Here**, **Move Caption to Parent** or **Move Caption to Child**.

### ***To hide relationship names on the Workspace***

Click  **Hide Captions** on the toolbar.

(Also right-click the Workspace | **Workspace Format** | **General** tab | select **Hide Line Captions**.)

### ***To find the relationship line by its name on the Workspace***

Click the relationship name (caption). The appropriate relationship line will be highlighted on the Workspace.

### ***To find the relationship name (caption) by its line on the Workspace***

Click the relationship line, its name (caption) will be highlighted in a frame.

## Multiple Selection/Move of Relationships

Relationships are moved together with selected group of entities. ([Select Objects](#))

To edit the selected relationships and change their format at one jump, use SHIFT for multiple selection. Then right-click any selected relationship and select **Edit** or **Format**.

## Format Relationship Lines

In Toad Data Modeler, you can change format of a particular relationship or more relationships at one jump. (Use SHIFT key for multiple selection.)

### ***To change format of all relationship lines on the Workspace***

Right-click the WS and select **Workspace Format**.

### To change format of the selected relationship(s)

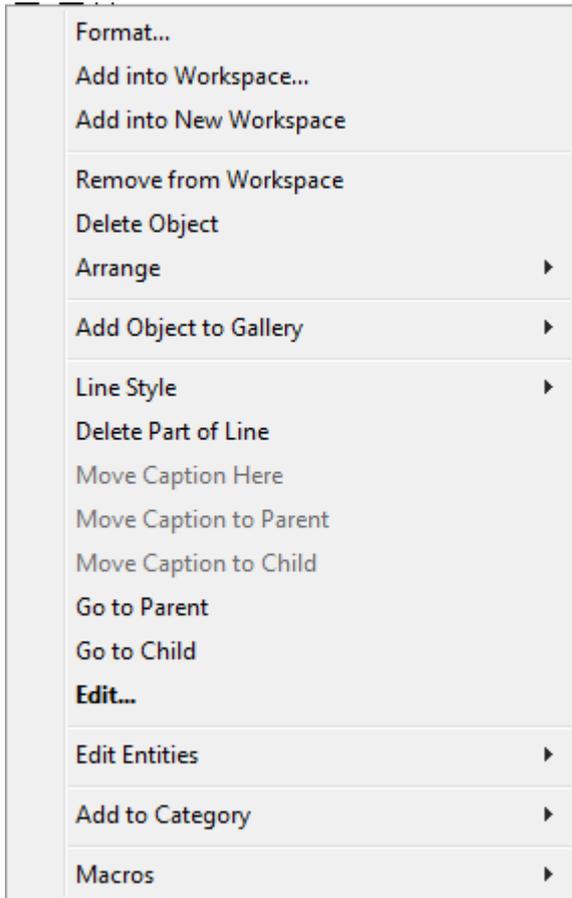
Right-click the relationship and select **Format**.

### To change format of the relationship name (caption)

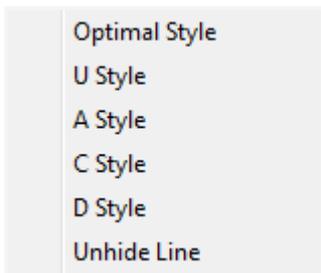
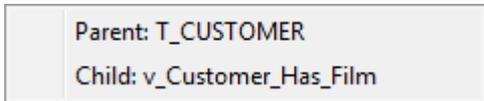
Right-click the relationship | **Format** | Click **Font Settings** and define the settings in the **Font** dialog.

**i** | TIP: Use the Inplace editor to change the relationship name in your diagram directly. [Inplace Editor](#)

## Relationship Right-Click Options

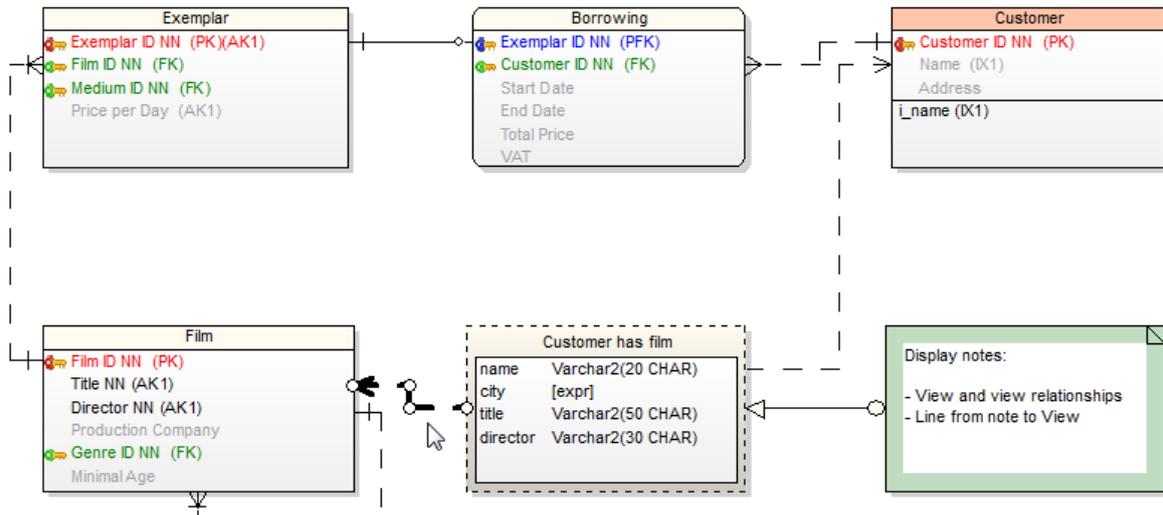


Option	Description
Format	Opens the <b>Object Format</b> dialog for the selected relationship.
Add into Workspace	Opens the <b>Workspaces</b> dialog where you can select a WS to add the relationship to. (The entities that the relationship connects will be inserted as well.)
Add into New Workspace	Creates a new Workspace in the Application Window and adds the relationship to it (including the entities that the relationship

Option	Description
	connects).
Remove from Workspace	Removes the selected shortcut from particular Workspace.
Delete Object	Deletes the selected relationship from model.
Arrange	Arranges the relationship in another layer. <a href="#">Arrange Objects in Layers</a>
Add Object to Gallery	You can add the relationship in a gallery.
Line Style	<p>There are several predefined patterns of line style. The letter in symbolizes the shape of the resulting relationship line:</p>  <p><b>Unhide Line</b> - Displays the part of relationship line hidden behind entity/view boxes.</p>
Delete Part of Line	Removes selected part of line.
Move Caption Here	Moves the caption of the relationship to the position where you pressed the right-click.
Move Caption to Parent	Moves the caption of the relationship to the parent table of the relationship.
Move Caption to Child	Moves the caption of the relationship to the child table of the relationship.
Go to Parent	Locates parent entity in ERD.
Go to Child	Locates child entity in ERD.
Edit	Opens the <b>Relationship Properties</b> dialog.
Edit Entities/Views	 <p>Opens the parent/child <b>Entity Properties</b> form.</p>
Add to Category	Add the relationship to current or new category.
Macros	Provides available macros for relationships.

## Views

Toad Data Modeler allows you to display and model views in your ER diagrams visually.



You can manage Views via:

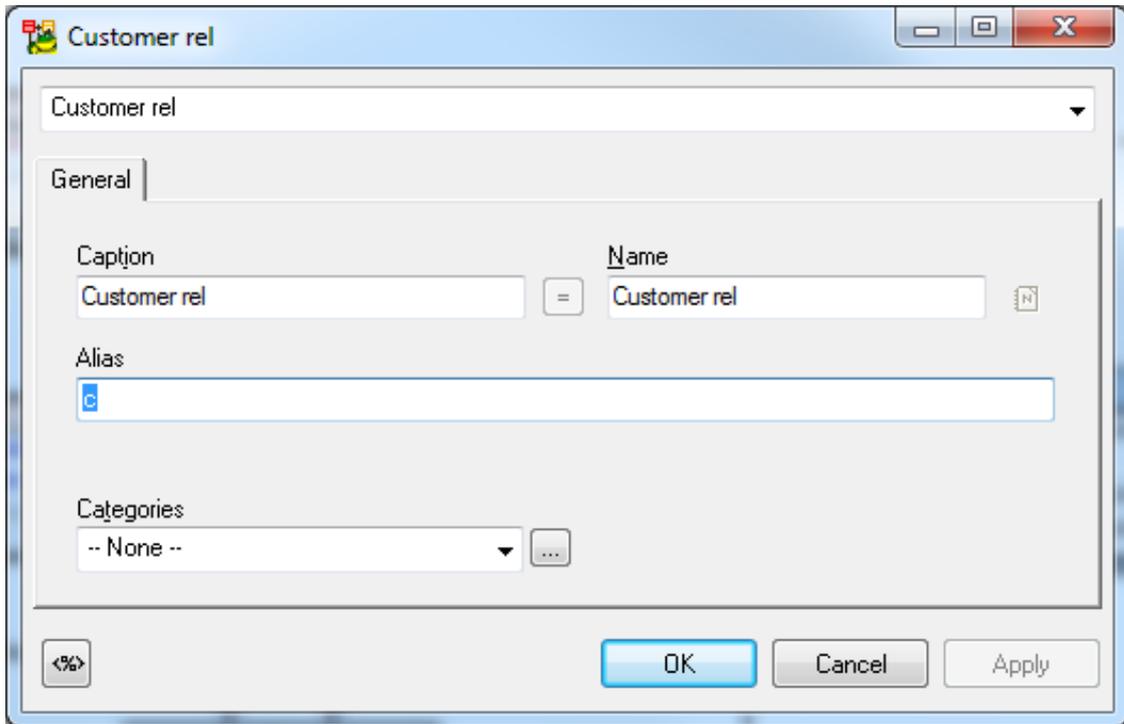
- **Model menu | Model Items | Views**
- **Model Explorer | Views folder**

View properties and options are database dependent.

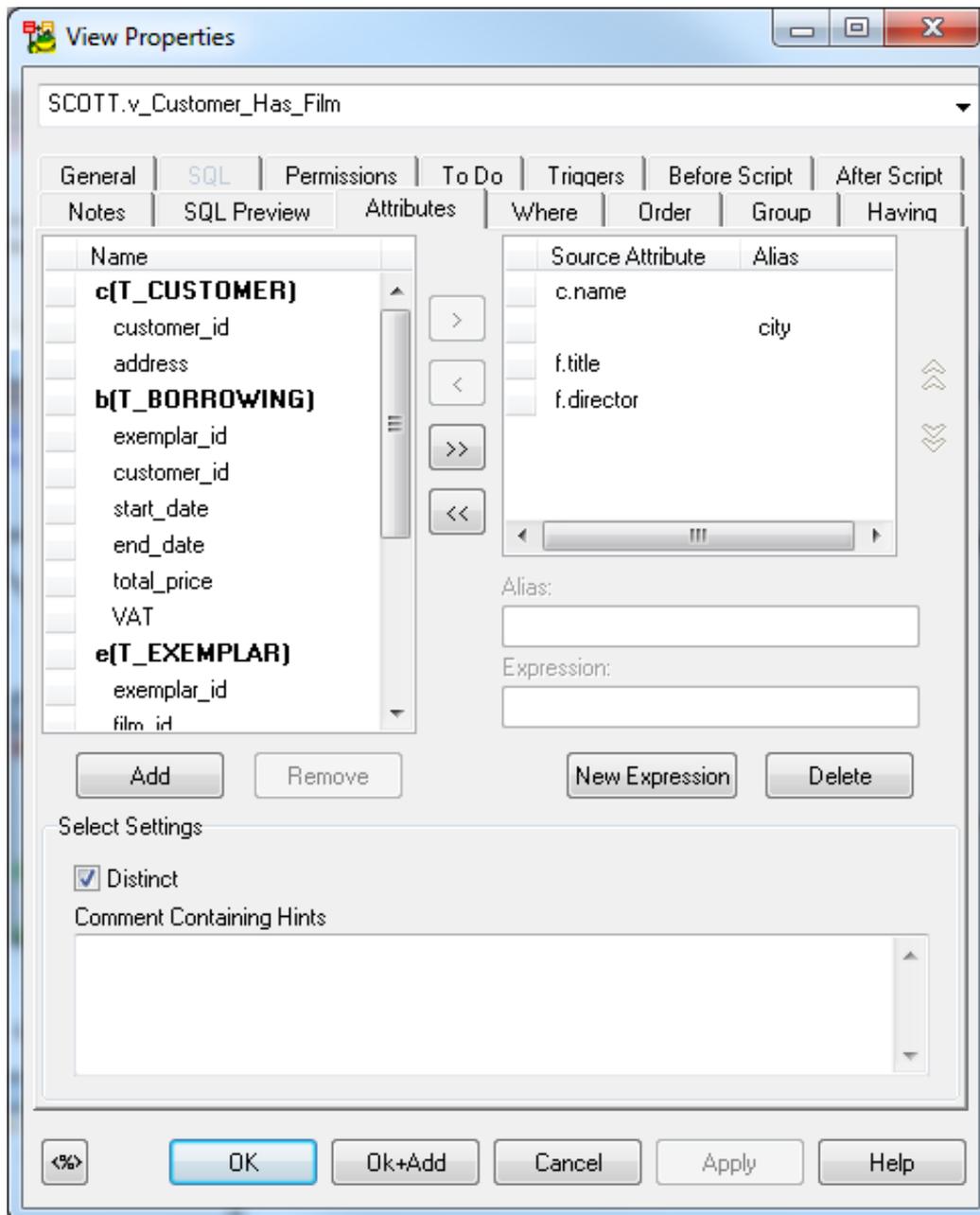
### To create a new view

1. Click  on the toolbar and then click the workspace.
2. Click  on the toolbar and draw view relationships between some existing table or view and your newly created view.

3. Double click the view relationship line to edit it. In Alias field you can specify a new alias name for the linked table.



4. Double-click the view to edit it.
5. Define properties on **General** tab and other tabs.
6. On tab **Attributes** you can select columns from the linked tables. Relationship aliases and names of entities are displayed.



7. You can use a template on **SQL** tab of the view.

**i** **NOTE:** As well as for entities and relationships, you can create shortcuts of views on the workspaces of your model. List of view shortcuts can be found in **Model Explorer** | in particular Workspace folder and in the **Views** folder | **Shortcuts**.

**i** **NOTE:** There are some limitations for modeling and reverse engineering of views in Toad Data Modeler. In the following cases views are loaded/modeled as "Select in Views as Text" without view relationships:

- When JOIN is used inside FROM statement. CROSS JOIN is the only supported type of JOIN.
- When a simple format for a name of a table is not used in FROM or when anything follows after a name of a table, e.g. functions, subquery, partition\_extension\_clause, PIVOT, UNPIVOT, DBLINK, flashback\_query\_clause, and row\_pattern\_clause.
- When "WITH common\_table\_expression", UNION, EXCEPT, INTERSECT, MINUS, model\_clause, and hierarchical\_query\_clause are used in view.
- When another definition exists between ORDER BY and the end of view, e.g. FOR clause in SQL Server.

## Materialized Views

Materialized Views are supported in the following databases: Oracle, DB 2, DB2 zOS, , Sybase SQL Anywhere 11, Teradata 13.

Toad Data Modeler allows you to display materialized views graphically in your ER diagram.

### To add a materialized view

Click  on the toolbar and then click anywhere on the work area.

or

**Model Explorer** | Right-click the **Materialized Views** folder | **Add Materialized View**.

### To edit a materialized view

Double-click the materialized view on the Workspace.

or

Edit the materialized view in **Model Explorer** | **Materialized Views** folder | double-click the selected materialized view (or right-click | **Edit**).

Option	Description
Object Navigator Box	All materialized views of your model are listed here. The combo-box allows you to edit them, one by one from one place. Use <b>Apply</b> to confirm all the changes you make.
<b>General Tab</b>	<b>Description</b>
Caption	Logical materialized view name
Name	Physical materialized view name
Schema	Schema selection box
Categories	Category selection box

Option	Description
Generate	Select it to generate the materialized view in final SQL (DDL) script.
Generate SQL only	Select it to generate only the SQL code written on tab <b>SQL</b> . The rest of items will be ignored.
<b>SQL Tab</b>	Write SQL code of materialized view subquery on this tab. <a href="#">About Templates</a>
<b>Permissions Tab</b>	Here you can assign Users or User Groups permissions to the materialized view.
<b>To Do Tab</b>	On this tab, you can write some tasks on the selected materialized view.  Note: To see all To Do tasks, select <b>Model   To Do</b> .
<b>Before Script Tab</b>	Whatever you write here, it will be generated before the materialized view definition.
<b>After Script Tab</b>	Whatever you write here, it will be generated after the materialized view definition.
<b>Notes Tab</b>	Tab for notes on particular materialized view. The text written here will display in a pop up dialog when you point your mouse cursor at the materialized view shortcut in your ER diagram.
<b>SQL Preview Tab</b>	Click  at the bottom of this tab to see the part of SQL code for the materialized view.
<b>Refresh Tab</b>	Use this tab to specify the default methods, modes and times for the database to refresh the materialized view.
<b>Physical Properties Tab</b>	Define storage characteristics of materialized view on this tab.
<b>Materialized Views Properties Tab</b>	Define other materialized view characteristics on this tab.
<b>Create Index Tab</b>	Create index sentences are written on this tab.

## Procedures

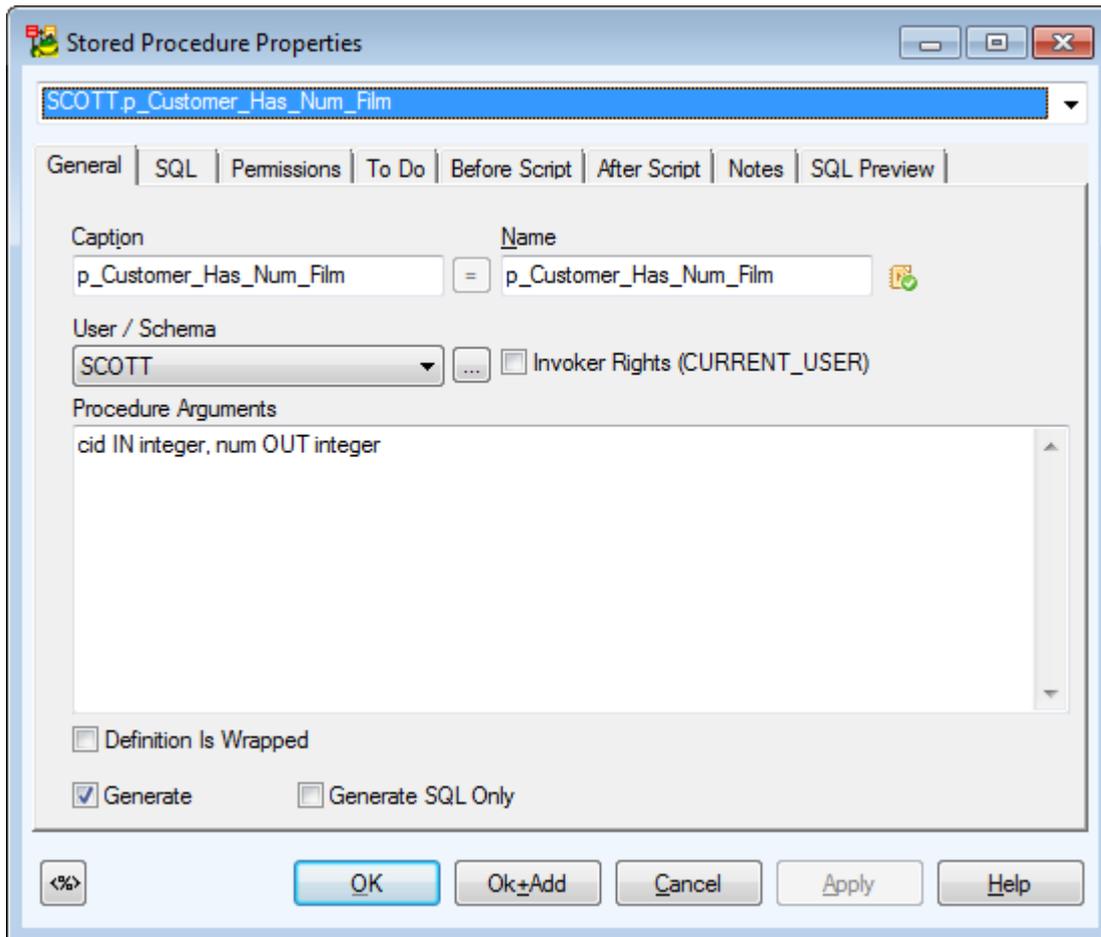
### *To add a procedure*

Right-click the **Procedures** item in **Model Explorer | Add Procedure**.

### *To edit a procedure*

Double-click the selected procedure in **Model Explorer | Procedures** (or right-click **Edit**).

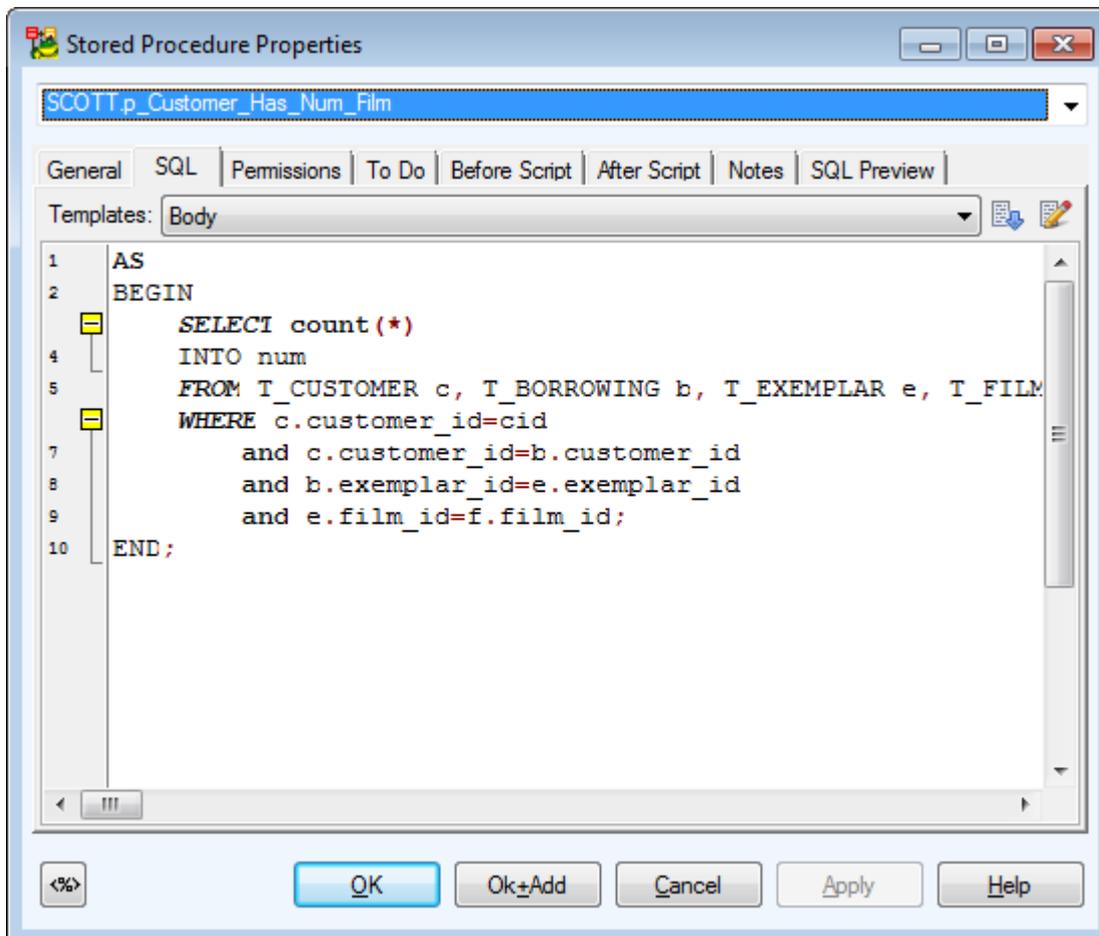
**Example: The Procedure Properties dialog (Oracle 10g)**



General Tab	Description
Caption	Logical procedure name
Name	Physical procedure name
Schema	Schema selection box
Procedure Arguments	Define procedure arguments here.
Invoker Rights	Database dependent option. See particular database reference for more information.
Definition is Wrapped	Database dependent option. See particular database reference for more information.
Generate	Select it to generate the procedure in final SQL (DDL) script.
Generate SQL only	Select it to generate only the SQL code written on tab <b>SQL</b> . The rest of items will be ignored.

<b>SQL Tab</b>	SQL code. (See the example in the following screenshot.) <a href="#">About Templates</a>
<b>Permissions Tab</b>	On this tab, you can manage permissions to particular procedure.
<b>To Do Tab</b>	Here you can write some tasks on the selected procedure. <b>Note:</b> To see all To Do tasks, select <b>Model   To Do</b> .
<b>Before Script Tab</b>	Whatever you write here, it will be generated before the Store Procedure definition.
<b>After Script Tab</b>	Whatever you write here, it will be generated after the Store Procedure definition.
<b>Notes Tab</b>	Tab for notes on particular procedure.

**Example: The Procedure Properties dialog | SQL tab (Oracle 10g)**



# Compare Procedures in Sync & Convert Wizard

When the **Sync & Convert Wizard** shows differences between procedures (on page **Select Items**), you can double-click the SQL item of procedures to display details about differences between them.

## Functions

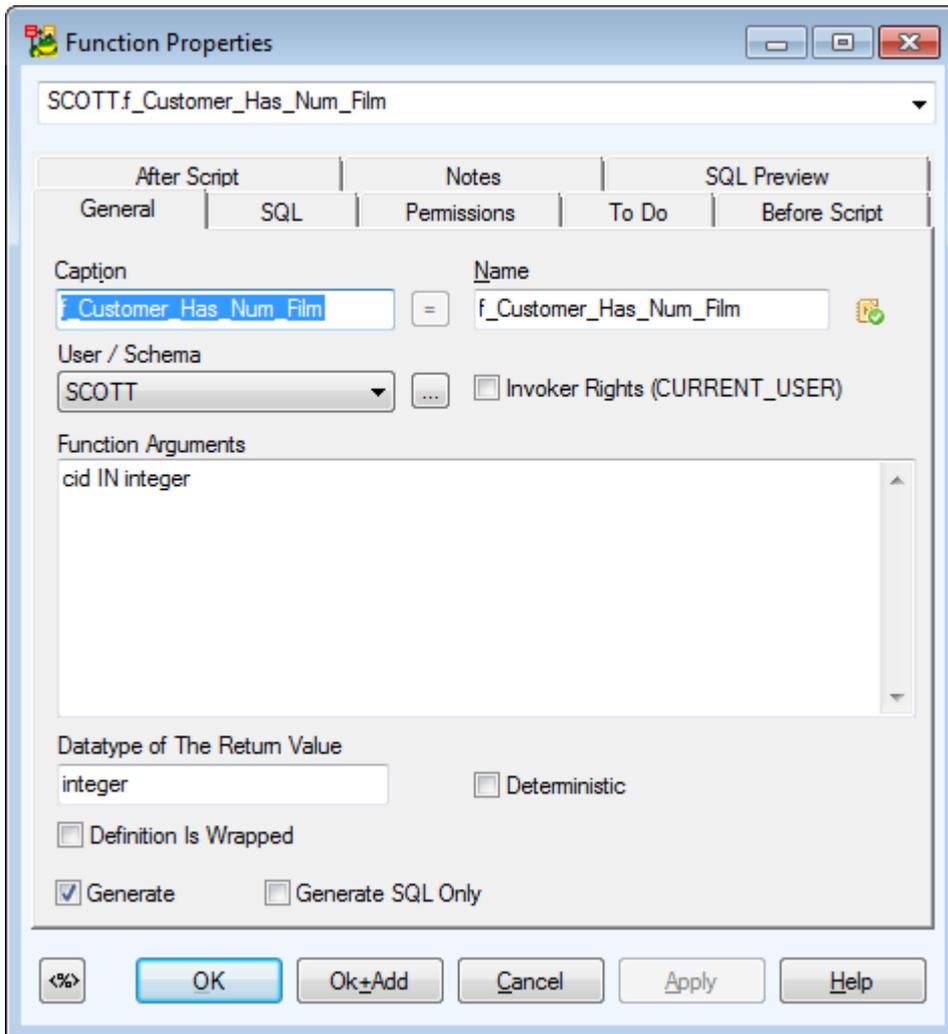
### To add a function

Right-click the **Functions** item in **Model Explorer | Add Function**.

### To edit a function

Double-click the selected function in **Model Explorer | Functions** (or right-click **Edit**).

**Example:** The **Function Properties** dialog (Oracle 10g)



### General Tab

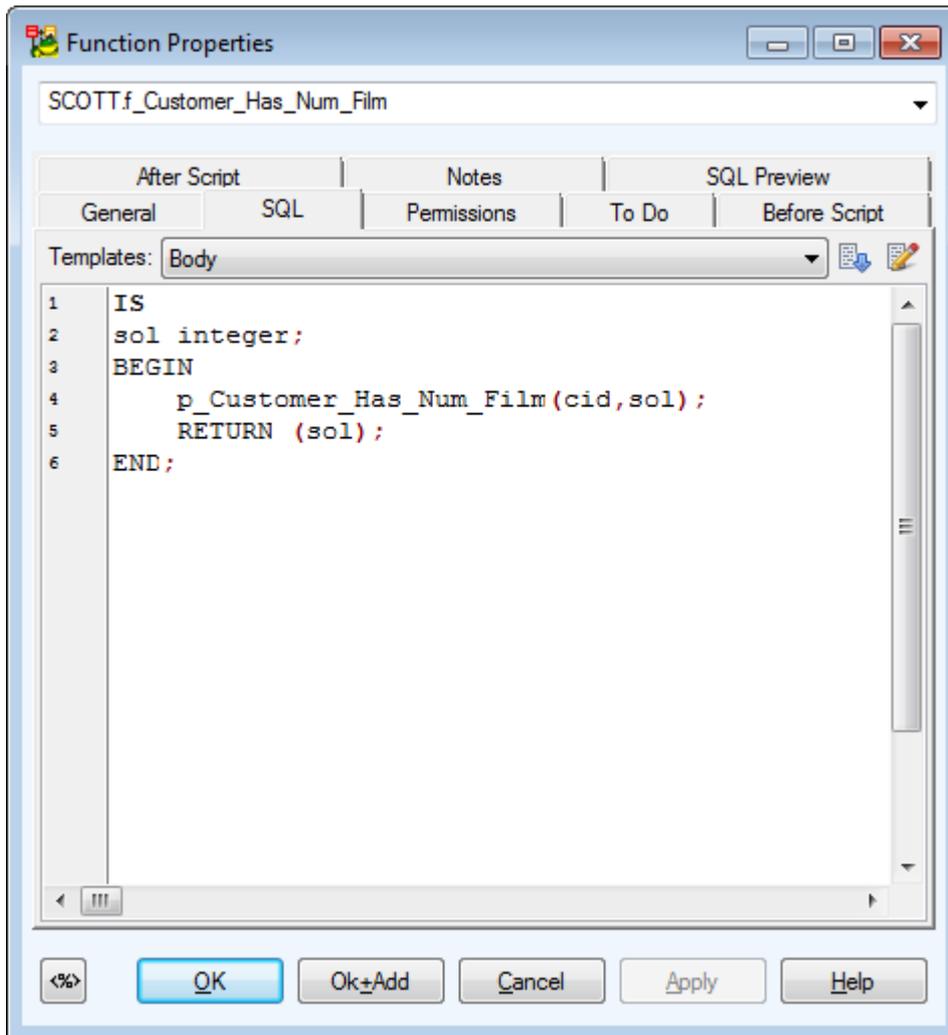
### Description

Caption

Logical function name

Name	Physical function name
Schema	Schema selection box
Invoker Rights (CURRENT_USER)	Database dependent option. See particular database reference for more information.
Function Arguments	Write function arguments here.
Datatype of the Return Value	Define data type of the return value.
Deterministic	Database dependent option. See particular database reference for more information.
Definition is Wrapped	Database dependent option. See particular database reference for more information.
Generate	Select it to generate the function in final SQL (DDL) script.
Generate SQL only	Select it to generate only the content of the <b>SQL</b> tab. All other items will be ignored.
<b>SQL Tab</b>	SQL code. (See the example in the following screenshot.) <a href="#">About Templates</a>
<b>To Do Tab</b>	On this tab, you can write some tasks on the selected function.
<b>Permissions Tab</b>	On this tab, you can assign a User or User Group permissions for selected function.
<b>Before Script Tab</b>	Whatever you write here, it will be generated after the function definition.
<b>After Script Tab</b>	Whatever you write here, it will be generated before the function definition.
<b>Notes Tab</b>	Tab for notes on particular function.

**Example:** The **Function Properties** dialog | **SQL** tab (Oracle 10g)



## Defaults

### To add a default

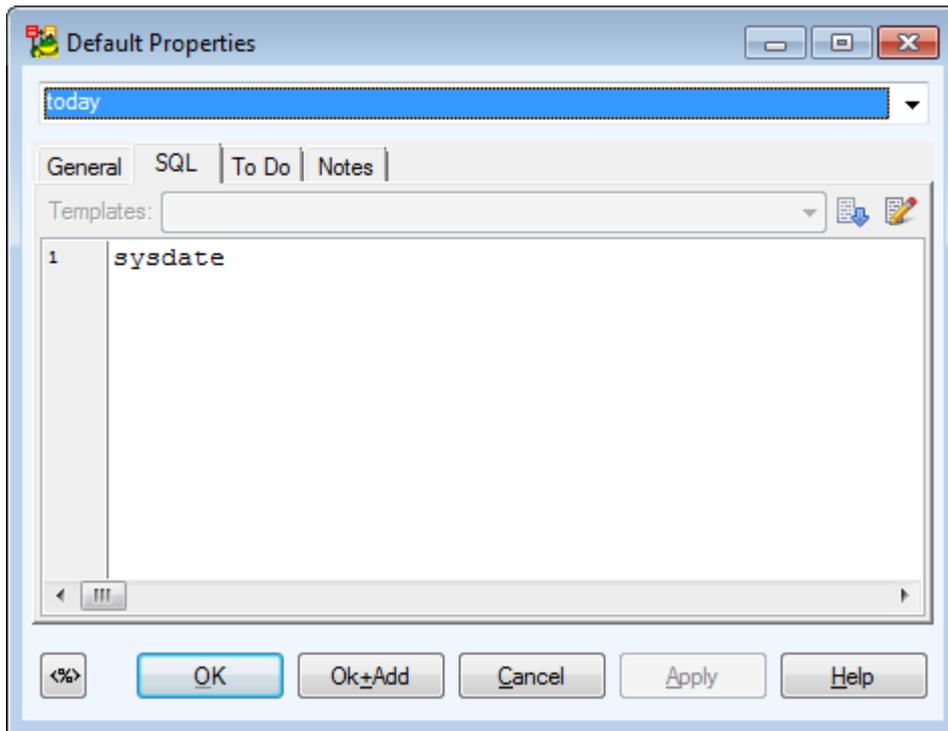
Select **Model | Model Items | Defaults** and click **Add** in the **Defaults** dialog.

### To edit a default

Select **Model | Model Items | Defaults** and double-click the selected default or click **Edit**.

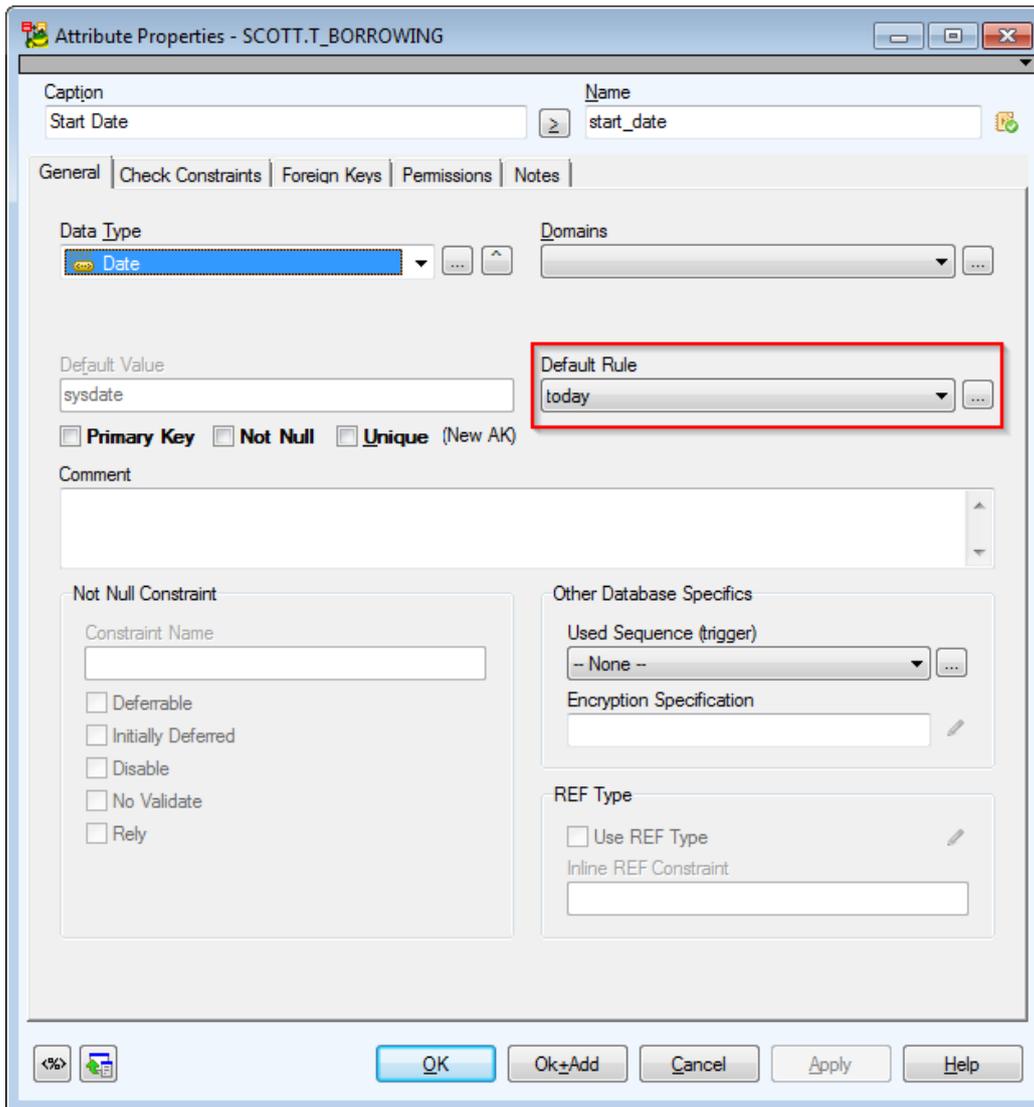
**i** Note: You can also edit/rename/delete defaults in **Model Explorer | Defaults** folder | Right-click the selected default.

**Example:** The **Default Properties** dialog (Oracle 10g)



### **To select a default for attributes**

1. Open the **Attribute Properties** dialog of the selected attribute | **General** tab.
2. From the **Default Rule** box, select a default or write your own default to the **Default** box.



## Default box versus Default Rule combo-box

**Default box** - Set a default for particular attribute.

or

**Default Rule** - Select a default rule from the combo-box or click the button on the right to access the **Defaults** dialog (also accessible via **Model** menu | **Model Items** | **Defaults**). There, you can define a new default rule. Then, if you want to change the default rule, you can simply change it via the **Defaults** dialog. The change will automatically apply in all attributes with this default rule.

**i** Note: You can define defaults for domains too.

## Check Constraint Rules

### *To add a check constraint rule*

Select **Model | Model Items | Check Constraint Rules** and click **Add** in the **Check Constraint Rules** dialog.

### *To edit a check constraint rule*

Select **Model | Model Items | Check Constraint Rules** and double-click the selected check constraint rule or click **Edit**.

**i** | Note: You can also edit/rename/delete check constraint rules in **Model Explorer | Check ConstraintRules** folder | Right-click the selected check constraint rule.

### *To assign a check constraint rule to attribute*

1. Edit attribute and click the **Check Constraints** tab.
2. **Add** a new check constraint, confirm **Apply** and click **Edit** to open its **Properties** dialog.
3. In the **Check Constraint Rule** box, select a rule for the attribute.

**i** | Note: You can assign rules to attributes and domains.

## Schemas

### *To add a schema*

Select **Model | Model Items | Schemas** and click **Add** in the **Schemas** dialog.

### *To edit a schema*

Select **Model | Model Items | Schemas** and double-click the selected schema or click **Edit**.

**i** | Note: You can also edit/rename/delete schemas in **Model Explorer | Schemas** folder | Right-click the selected schema.

## Synonyms

### *To add a synonym*

Select **Model | Model Items | Synonyms** and click **Add** in the **Synonyms** dialog.

### *To edit a synonym*

Select **Model | Model Items | Synonyms** and double-click the selected synonym or click **Edit**.

**i** | Note: You can also edit/rename/delete synonyms in **Model Explorer | Synonyms** folder | Right-click the selected synonym.

## Users

Toad Data Modeler allows you to define Users and assign them to User Groups. Later you can assign the Users and User Groups permissions to particular objects.

### To add a user

Click  on the toolbar and click **Add** in the **Users** dialog.

or

In **Model Explorer** | Right-click the **Users** folder | **Add User**.

### To edit a user

Select **Model** | **Model Items** | **Users** | double-click the selected user or click **Edit**.

or

In **Model Explorer** | **Users** folder | double-click the selected user or right-click | **Edit**.

Option	Description
Object Navigator Box	All users of your model are listed here. The combo-box allows you to edit them, one by one from one place. Use <b>Apply</b> to confirm all the changes you make.

General Tab	Description
Caption	Logical user name
Name	Physical user name
Password	User's password

#### Membership Tab

On this tab, you can assign a User to User Group.



Note: One user can be assigned to more User Groups.

Select a User Group and click appropriate button:



Adds selected item(s).



Puts back selected item(s).



Adds all items.



Puts back all items.

#### To Do

On this tab, you can write some tasks on particular user.

### To assign users to user groups

**Example:** All admins should be assigned to user group *Administrators*.

## Method A - via the Users dialog

1. In the **Users** dialog, select the *SCOTT* user and click **Edit**.
2. In the **User Properties** dialog, click the **Membership** tab.
3. Select the *Administrators* user group and click the **Add** arrow button to shift the selected group to the window **Selected**.  
(The User Groups have already been defined in **Model | Model Items | User Groups**.)

**i** TIP: If you confirm **Apply**, the **User Properties** dialog will remain opened, and you can comfortably assign other users to user groups. - Simply select another user from the object navigator box at the top.

## Method B - via the User Groups dialog

1. Click  on the toolbar.
2. From the **User Groups** dialog, select *Administrators* user group and click **Edit**.
3. In the **User Group Properties** dialog, click the **Members** tab.
4. Select *SCOTT* and click the **Add** arrow button to shift the selected user to the window **Selected**.

### User Groups

Toad Data Modeler allows you to define **Users** and assign them to **User Groups**. You can then assign User and User Groups permissions to objects.

#### To add a user group

Click  on the **Users Toolbar** and click **Add** in the displayed dialog.

or

In **Model Explorer** | Right-click the **User Groups** folder | **Add User Group**.

#### To edit a user group

Go to **Model Menu | Model Items | User Groups** | double-click the selected user group or click **Edit**.

or

In **Model Explorer** | **User Groups** folder | double-click the selected user group or right-click | **Edit**.

Option	Description
Object Navigator Box	All user groups of your model are listed here. The combo-box allows you to edit them, one by one from one place. Use <b>Apply</b> to confirm all the changes you make.
General Tab	Description
Caption	Logical user group name

Option	Description
Name	Physical user group name
<b>Membership Tab</b>	<b>Description</b>
Available	Available user groups
Selected	Selected user groups
<b>Members Tab</b>	<b>Description</b>
User Groups section	You can create another group in already existing group. (This is possible for some databases.) From the list of existing groups, select a group and click the <b>Add</b> arrow button.
Users section	Here, you can assign users to a group. <b>Note:</b> You can assign users to a user group also in the <b>Users</b> dialog.
<b>To Do Tab</b>	On this tab, you can write some tasks on the selected user group. <b>Note:</b> To see all To Do tasks, select <b>Model   To Do</b> .



Note:

1. To copy user groups, use CTRL + Drag&Drop techniques.
2. To move user groups, use Drag&Drop techniques.  
You can copy and move your user groups within a model and between models of the same and different databases:
  - In **User Groups** dialog (**Model Menu | Model Items | User Groups**)
  - In **Model Explorer | User Groups** folder
  - Between **Model Explorer** and **User Groups** dialog
3. To delete user groups, go to:
  - **Model Menu | Model Items | User Groups** | Select a user group and click **Delete**.
  - **Model Explorer | User Groups** folder | Right-click and select **Delete Item**.

## Select Target Database

The first step to create a model in Toad Data Modeler is to choose your target database.

### Scenario

Creating new **Oracle 10g** physical model named *Videorental*.

1. Click  on **Main Toolbar** (or press CTRL+N).
2. Click the **Physical Data Model** tab and select the target database - *Oracle 10g*.
3. Write *Videorental* into the **Model Name** textbox.

- 
4. Confirm by clicking **OK**.

Result:

- The model will be shown in **Application View**.
- The **All Items** workspace displays automatically.
- The status bar in of **Application Window** displays the database name. (*DB: Oracle 10g* in our case.)
- Most of the menus and toolbars become activated.



Note:

**Database Name** - The target database for which the model is created (e.g. Oracle 10g). The database name can be found at the bottom of the Application Window or in displayed pop-up hint when you hover your mouse cursor on the model name in the **Application View**.

**Model Name** - A model can be saved to more files. In Toad Data Modeler, Model Name should be understood as a title of a document that can be saved to several files of a different name.

**File Name** - A name of the file where the model is saved. File Name is defined after you select **Save Model** or **Save Model as**.

[Rename](#)

## Create Entities

In Toad Data Modeler, there are several ways how to create entities - on the Workspace, via **Model Explorer** and in the **Entities** dialog (**Model | Model Items | Entities**). Create entities directly on the Workspace.

---

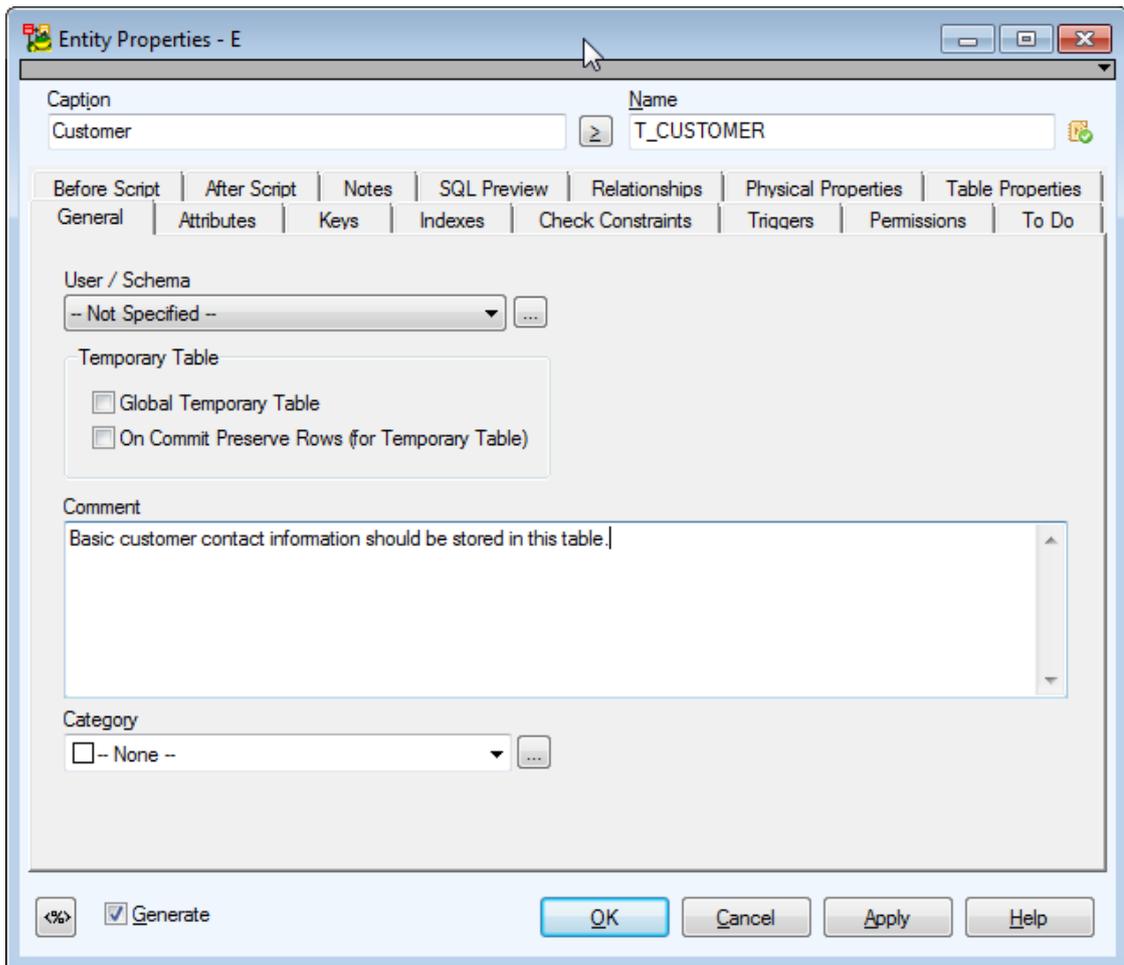
### Scenario

Create entity *Customer* on the Workspace in your *Videorental* model.

---

1. Click  on the toolbar (also CTRL+E) and then click anywhere on the Workspace.
2. Double-click the entity to edit it.

3. Define the entity caption and name



Caption     Logical entity name - *Customer*.

Name        Physical entity name - *T\_CUSTOMER*.

4. Define other properties on tab **General** and other tabs (e.g. **Notes**, **Comments** etc.). To save the changes simultaneously and leave the form open, click **Apply**.

## Create Attributes

You can create attributes in:

- Entity right-click menu on Workspace
- **Entity Properties** dialog
- **Attribute Properties** dialog
- Model Explorer

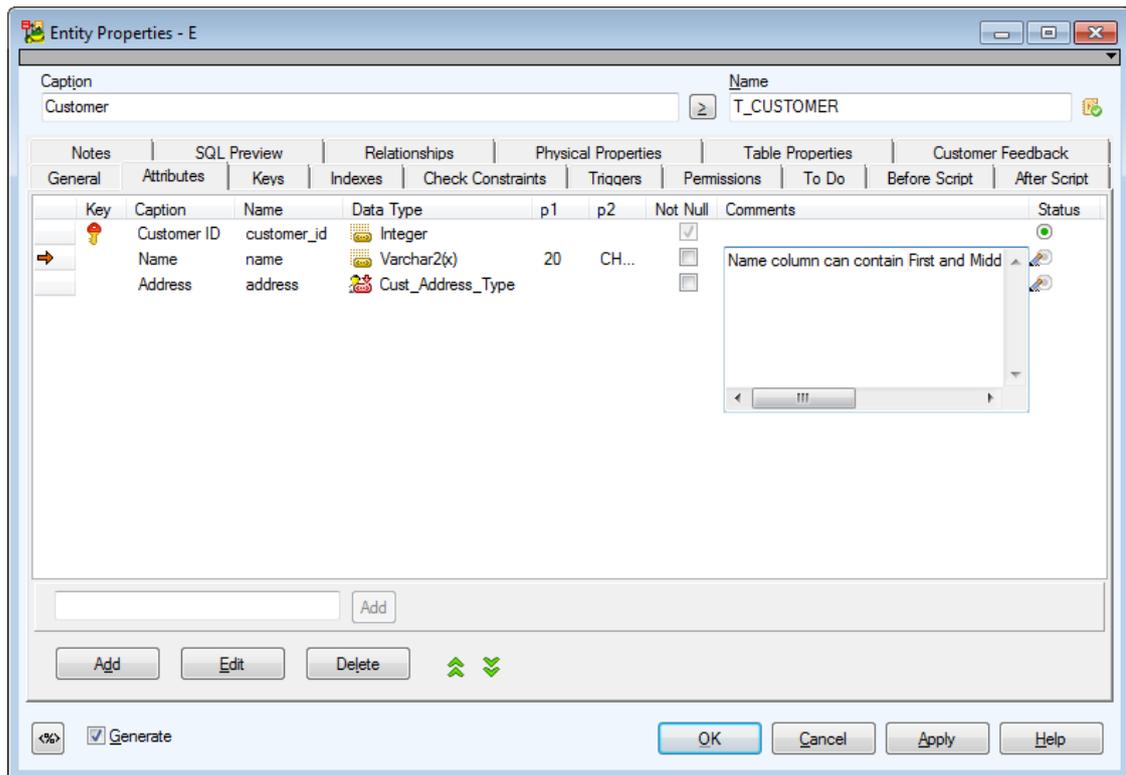
### To create an attribute on the Workspace

Right-click an entity on the Workspace and select **Add | Attribute**. The **Attribute Properties** dialog opens.

**i** Note: Select the **Primary Key (Unique)** checkbox to set the attribute as PK or unique attribute.

### To create an attribute in the Entity Properties form

1. Double-click an entity on Workspace.
2. In the **Entity Properties** form, switch to the **Attributes** tab and click **Add**.



### Entity Properties attribute columns

Column/Option	Description
Key	Graphical representation of keys of a particular attribute
Caption	Logical attribute name
Name	Physical attribute name
Data Type	Data Type of an attribute

Column/Option	Description
	<p><b>i</b> TIP: You can set a default data type for newly created attributes. See <b>Settings Menu   Options   Physical Model   *database platform and version*   Default Data Type</b> combo-box. (The selected data type will be also applied to Dictionary Types and Domains.)</p>
p1	Parameter 1. Only some of the available data types have this parameter. Defines properties of the selected data type, e.g. length in case of the Char data type.
p2	Parameter 2. Only some of the available data types have two parameters. E.g. the Decimal data type has two parameters, which define precision and scale.
Not Null	When checked, the attribute cannot be empty.
Comments	Comments or descriptions related to the attribute
Status	Shows status of attributes in grid. <a href="#">Status of Items in Grids</a>

#### Buttons:



- opens the **Application Variables** form

**Smaller Add button**- adds new attributes quickly, just enter attribute name and then click **Add**.

**Bigger Add button** - adds an attribute

**Edit** - opens **Attribute Properties** of the selected entity

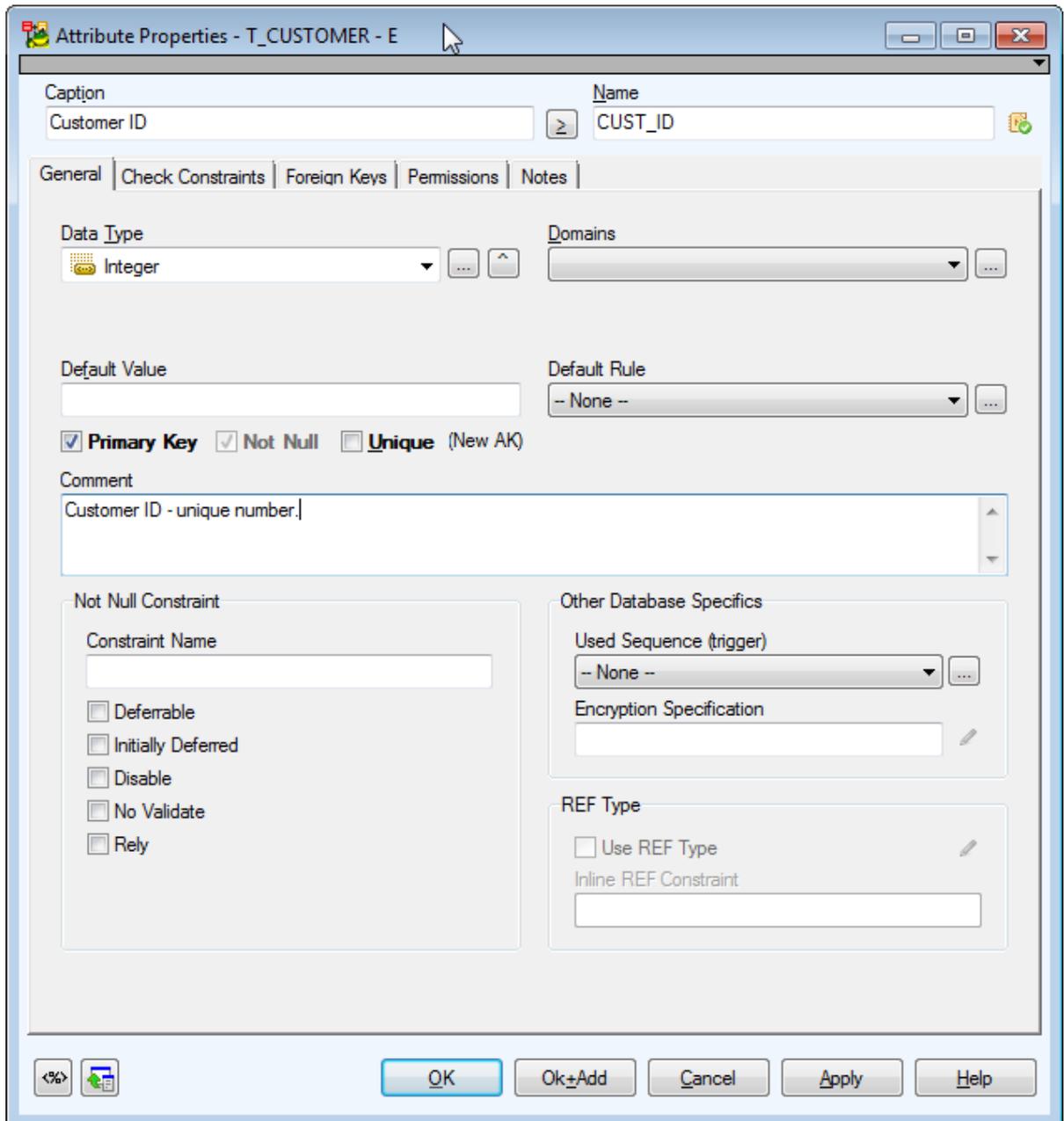
**Delete** - deletes the selected attribute



- moves the selected attribute up or down in the list

#### **To create another attribute in the Attribute Properties form**

Click **OK+Add**.



### **To create an attribute in Model Explorer**

Unfold the **Entities** folder | Unfold the specific entity folder | Right-click the **Attributes** folder | **Add Attribute**.

## **Create Keys**

An entity can have a primary key and many alternate keys. The keys are stored in the **Keys** tab in the **Entity Properties** form.

### To create a key

- Right-click an entity on the Workspace and select **Add | Key**. The **Key Properties** dialog opens.

### To create a PK (unique) attribute

1. Right-click an entity on the Workspace and select **Add | Attribute**. The **Attribute Properties** dialog opens.
2. On tab **General**, select the **Primary Key (Unique)** checkbox.

### To assign an attribute to key

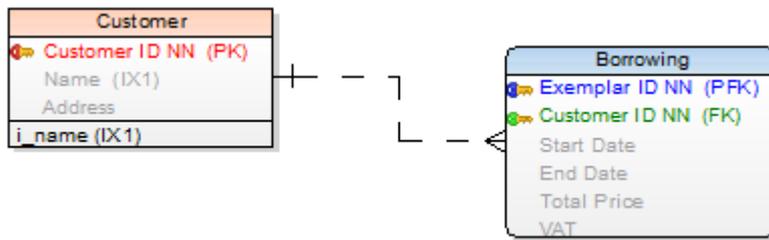
- In the **Entity Properties** dialog, **Attributes** tab, double-click the empty space in the **Key** column next to the selected attribute.

or

- In the **Key Properties** dialog | **Attributes** tab.

**i** Note:

- Primary keys are graphically marked by red key by default.



- A primary key can be added to more attributes.
- It is possible to delete primary keys. It is not possible to delete alternate keys of unique attributes.

## NotNull Property for PK and AK Attributes

Toad Data Modeler allows you to check/uncheck the NotNull property for PK and AK attributes. This is possible only if the **Allow Null Attributes in Keys** checkbox is selected (unchecked by default).

### To select this checkbox

Select **Settings | Options | Physical Model** and select the checkbox.

If this checkbox is not selected, Toad Data Modeler doesn't allow you to uncheck the NotNull checkbox in key attributes.

The availability of this option differs by database:

Database	Allow NULL in PK	Allow NULL in AK
DB2 z/OS v. 11	n/a	n/a
DB2 v. 8, 9.x, 10.x	n/a	n/a
Greenplum	n/a	available

Database	Allow NULL in PK	Allow NULL in AK
Ingres 9.3, 10	n/a	n/a
Microsoft Access	available	available
Microsoft Azure SQL Database	n/a	available
SQL Server 2012	n/a	available
SQL Server 2014	n/a	available
MySQL 5.5 and higher versions	n/a	available
Oracle	available	available
PostgreSQL	n/a	available
SQLite 3.7	available	available
Sybase ASE, Sybase IQ	n/a	n/a
Sybase SQL Anywhere	n/a	n/a
Teradata	n/a	n/a

If the **Allow Null Attributes in Keys** checkbox is selected:

- When you assign an attribute to a key (PK, AK), the **Not Null** checkbox of the attribute will be checked. (But you are able to uncheck it)
- During propagation of the key via the identifying relationship, the Not Null property in child attribute is inherited from the parent attribute. The only exception is when a database doesn't support Null value in primary key, which would be created via the propagation.
- During Model Conversion, different settings of the Not Null property and its support in different databases are taken into consideration.
- For databases that support Null value in child attribute, the **Mandatory Parent** checkbox is selected in the **Relationship Properties** dialog and should behave coherently to Null value in child attribute as well as it behaves for non-identifying relationships.
- For databases that support Null value in child attribute, the settings of the **Synchronize NotNull with Mandatory Parent** option work the same way for PFK as for FK.

See [Synchronization of NotNull and Mandatory Parent](#) for more information.

## Create Relationships

1. Select a relationship type - click the appropriate relationship icon on the toolbar:

-  Identifying relationship (also CTRL+R)
-  Non-identifying relationship
-  M:N relationship

2. Move your mouse cursor over the work area. (The cursor changes its appearance.)
3. Click the first entity (parent) and then the target entity (child).



TIP: Hide relationship names:

1. Right-click the Workspace and select **Workspace Format**.
2. In the **Workspace Format** dialog | **General** tab, select **Hide Line Captions** checkbox.

### ***To create self-relationship for non-identifying relationship***

1. Click the Non-identifying relationship icon on the toolbar .
2. Move your mouse cursor over the work area.
3. Double-click the selected entity on the Workspace.



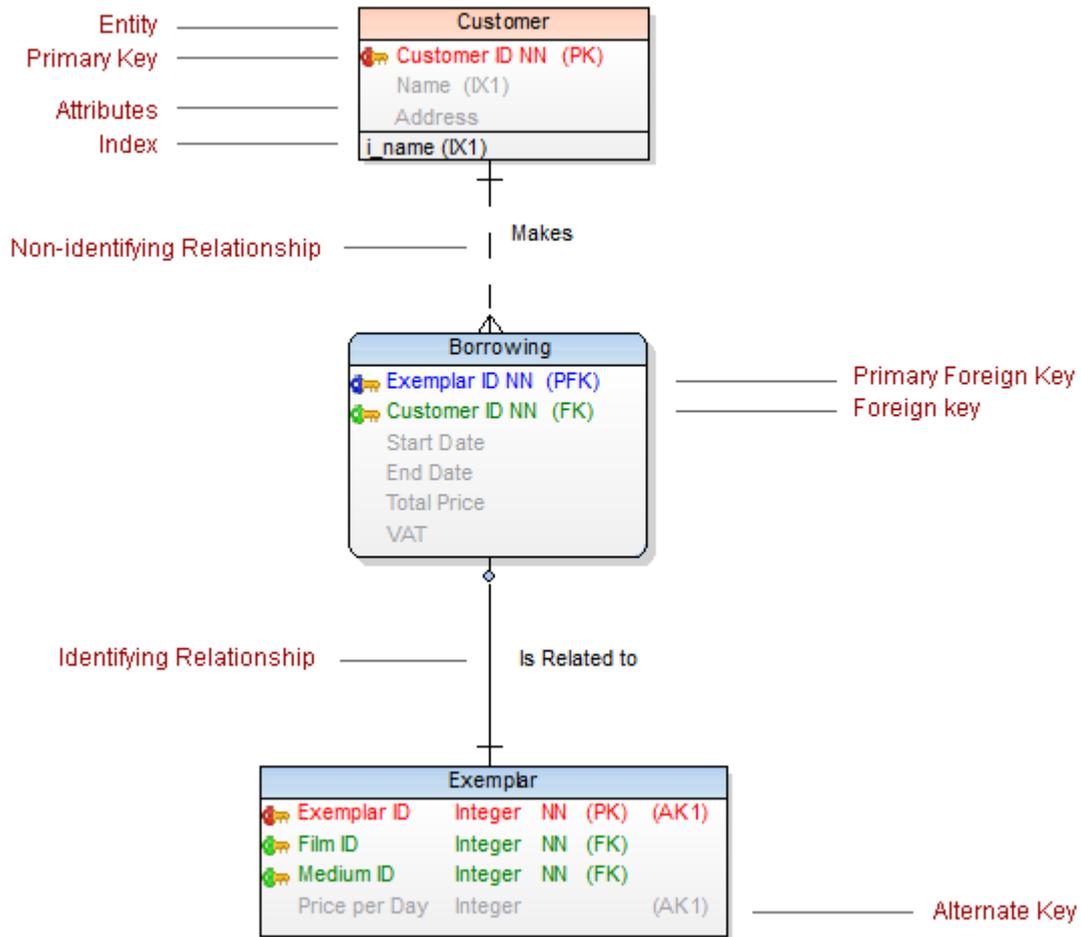
TIP: Before you create a self-relationship, select **Settings | Options | Physical Model | Self Relation Attribute Name/Caption** and define a name for propagated attributes there (e.g. via prefix, suffix, application variable.)

### ***To add multiple relationships***

1. Press SHIFT and click the **Relationship** icon. See the blue frame in the icon now. .
2. Create as many relationships in you model as you need.
3. Right-click the work area (or click the **Relationship** icon again) to turn this function off.

# Notation and Cardinality

## IE Notation

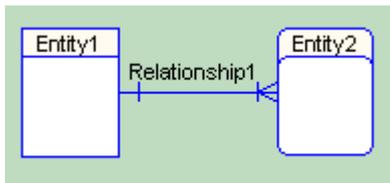


## Cardinality

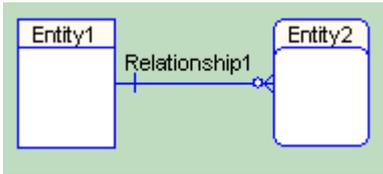
One-to-many relationship is represented by this symbol:

One-to-one relationship is represented by this symbol:

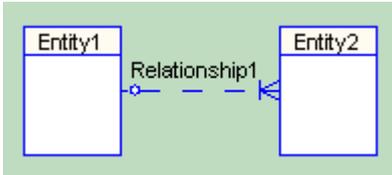
**Parent:** mandatory **Child:** mandatory



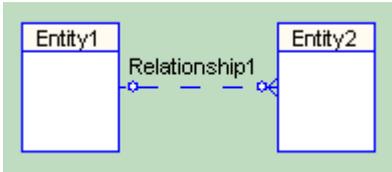
**Parent:** mandatory **Child:** optional



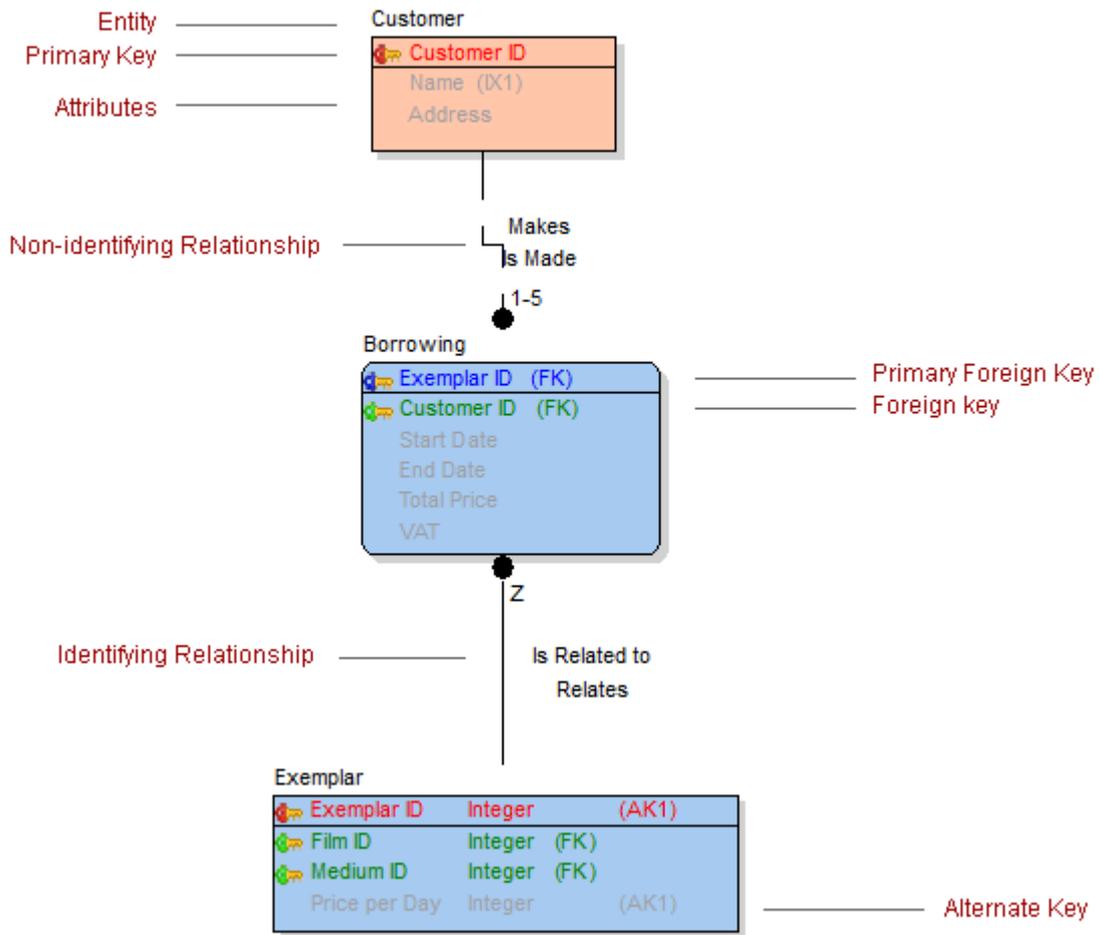
**Parent:** optional **Child:** mandatory



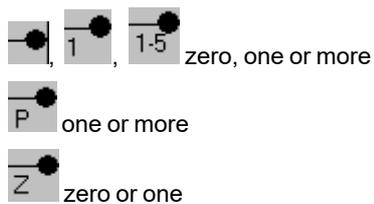
**Parent:** optional **Child:** optional



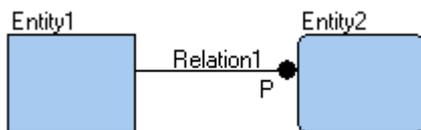
## IDF1X Notation



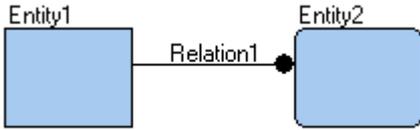
## Cardinality



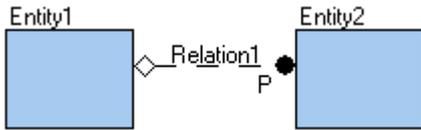
**Parent:** mandatory **Child:** mandatory



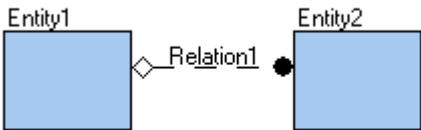
**Parent:** mandatory **Child:** optional



**Parent:** optional **Child:** mandatory



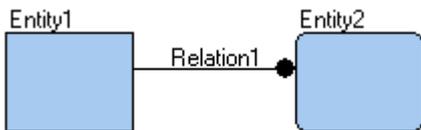
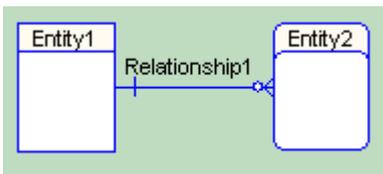
**Parent:** optional **Child:** optional



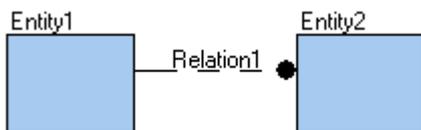
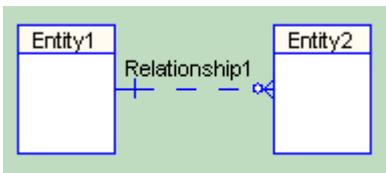
See [Synchronization of Not Null and Mandatory Parent](#) for more information.

## Relationship Types

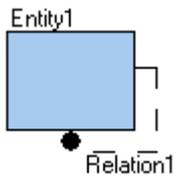
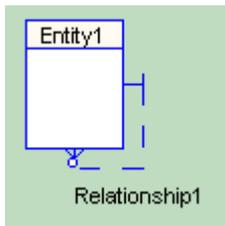
### Identifying Relationship



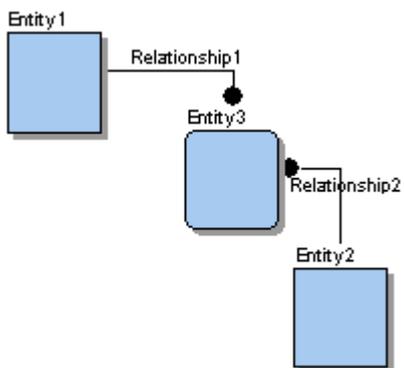
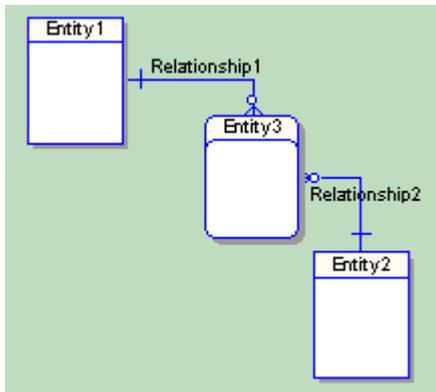
### Non-identifying Relationship



## Non-identifying Self-relationship



## M:N Relationship



# Optional/Mandatory Parent/Child

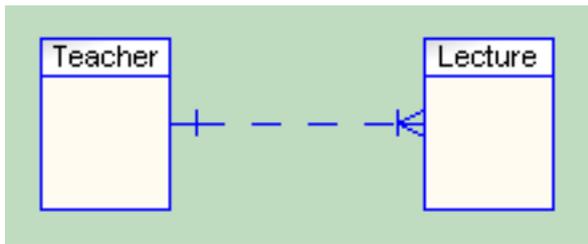
## To set Parent/Child entity as Mandatory/Optional

1. Double-click a relationship on workspace to open **Relationship Properties**.
2. Switch to the **General tab** and check/uncheck **Mandatory Parent/Child** checkbox in the **Cardinality** section.

Parent: **Mandatory**

Child: **Mandatory**

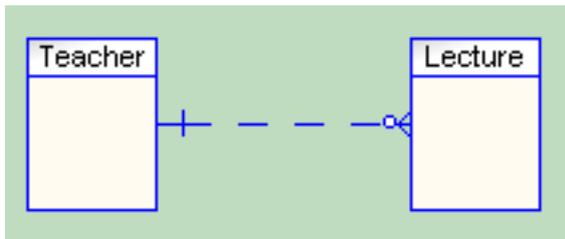
A teacher **MUST** teach a lecture (the record related to lecture is mandatory), a lecture **MUST** be taught by a teacher (the record related to teacher is also mandatory. Teacher is Mandatory.)



Parent: **Mandatory**

Child: **Optional**

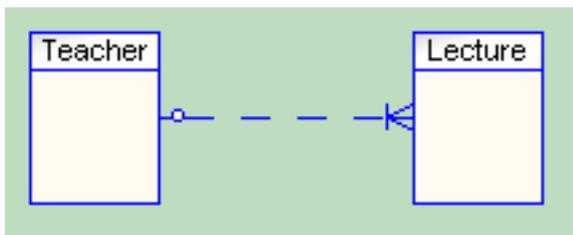
A teacher **MAY** teach a lecture (the record related to lecture is Optional), a lecture **MUST** be taught by a teacher (Teacher is Mandatory.)



Parent: **Optional**

Child: **Mandatory**

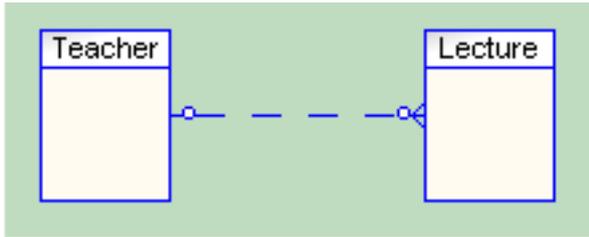
A teacher **MUST** teach a lecture (lecture is Mandatory), a lecture **MAY** be taught by a teacher (teacher is Optional).



Parent: **Optional**

Child: **Optional**

A teacher **MAY** teach a lecture (lecture is Optional), a lecture **MAY** be taught by a teacher (teacher is Optional).



The information can be found on **Relationship Properties** form, see graphical representation at bottom of the form.

Relationship Properties

Caption: Makes      Name: makes

General | To Do | Notes | SQL Preview | Index to Foreign Key

Relationship Type

Identifying       Non-Identifying

Parent Key: pk\_T\_CUSTOMER

Parent	Child
Customer ID	customer_id

Referential Integrity

Parent UPDATE: Restrict

Parent DELETE: Restrict

Settings

Deferrable  
 Deferred  
 Disable  
 No Validate  
 Rely

Cardinality

Mandatory Parent      Cardinality: 5  
 Mandatory Child      1..1 to 1..5

Category: - None -

Parent Entity: SCOTT.T.CUSTOMER      Child Entity: SCOTT.T.BORROWING

Generate      OK      Cancel      Apply      Help

## Set up Referential Integrity Rules

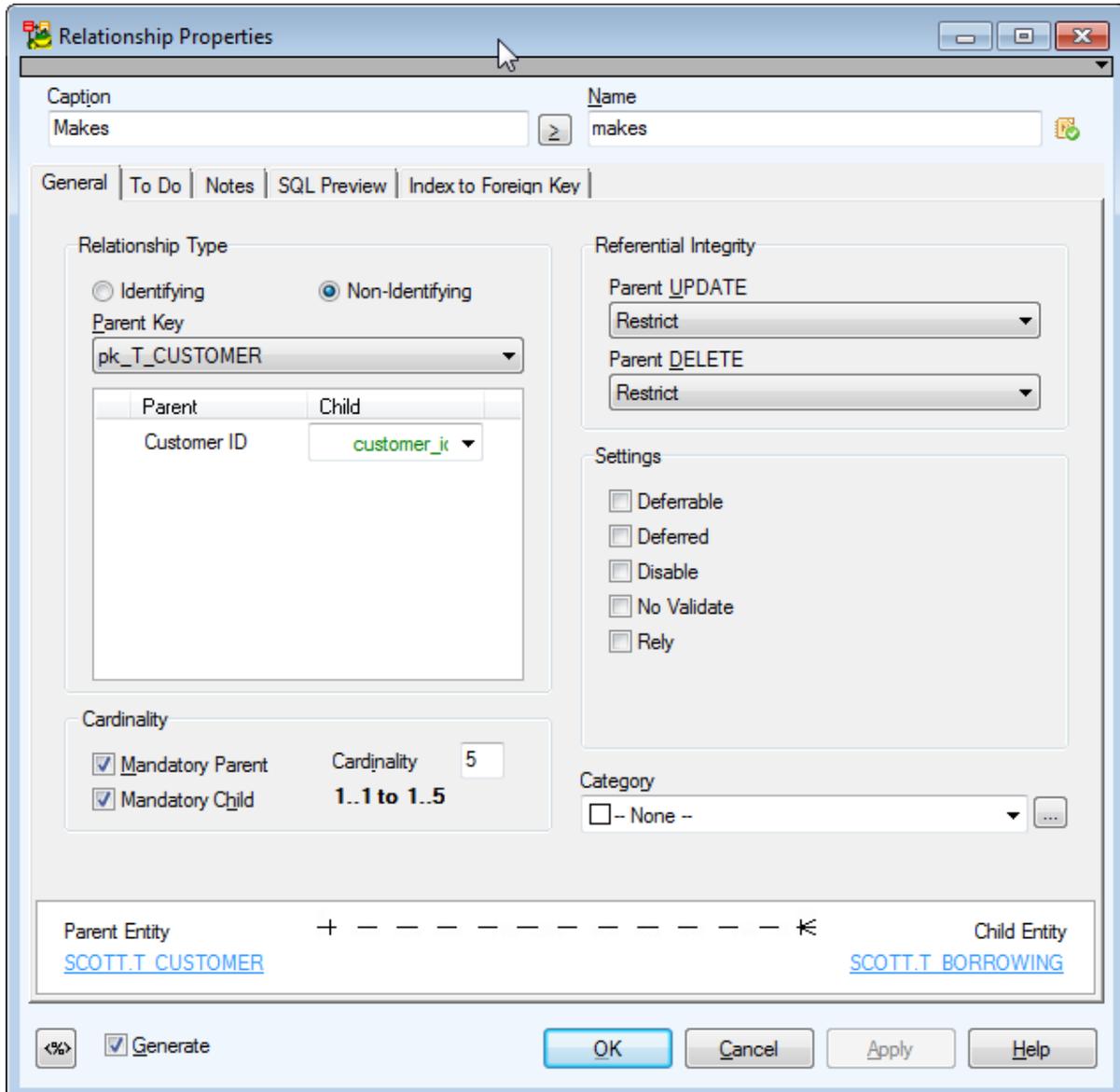
### To set up referential integrity rules for Parent entities

Open the **Relationship Properties** form and see the Referential Integrity group box.

Here, you can change the None rule to:

- Restrict
- Cascade
- Set Null
- Set Default

The default rule can be re-configured.



**i** Note: Some databases do not support Referential Integrity (RI) rules.

## Connect Parent and Child Entities

Toad Data Modeler allows you to connect Parent and Child entities through the following unique values:

- Key (Primary or Alternate Key) of parent entity
- Unique Index of parent entity

### To change the linking method

1. Open the **Relationship Properties** form and see the **Relationship Type** group box.
2. From the **Parent Key** box, select the method. If neither Unique item, nor Alternate Key exists, only the Primary key is available.

Relationship Properties

Caption: Is Rated      Name: is\_rated

General | To Do | Notes | SQL Preview | Index to Foreign Key

Relationship Type

Identifying       Non-Identifying

Parent Key: title-director

Parent	Child
Title	title
Director	director

Cardinality

Mandatory Parent      Cardinality: N

Mandatory Child      1..1 to 1..n

Referential Integrity

Parent UPDATE: Restrict

Parent DELETE: Restrict

Settings

Deferrable

Deferred

Disable

No Validate

Rely

Category: -- None --

Parent Entity: SCOTT.T.FILM      Child Entity: SCOTT.T.CUSTOMER.RATING

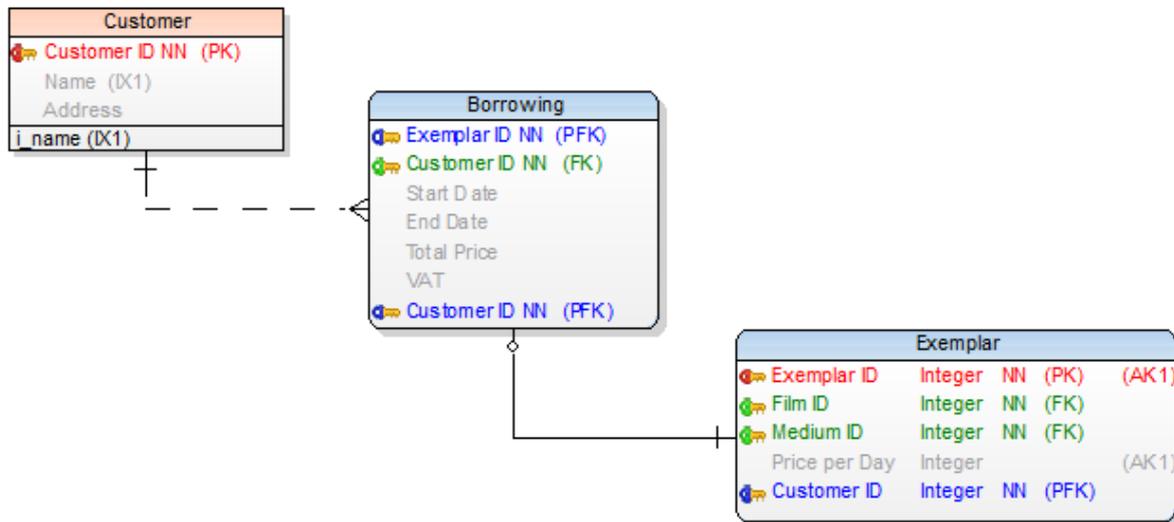
<>  Generate      OK      Cancel      Apply      Help

**i** Note: Some databases do not support Alternate Keys.

# Understanding Foreign Keys

There are two types of foreign keys in Toad Data Modeler:

- Primary Foreign keys (PFK, graphically marked by blue key).
- Foreign keys (FK, graphically marked by green key).



## PFK

As soon as you create an identifying relationship, Toad Data Modeler automatically adds a copy of the primary key of the Parent table to the Child table - **Primary Foreign Key (PFK)**. (The foreign key in Child table is a part of the primary key.)

## FK

As soon as you create a non-identifying relationship, Toad Data Modeler automatically adds a copy of the primary key of the Parent table to the Child table - **Foreign key (FK)**. (This foreign key in Child table is not a part of the primary key.)

In other words, foreign keys (PFKs and FKs) are imported from parent entities to child entities automatically when you create relationship. Foreign keys cannot exist without relationships. Therefore, you are not allowed to delete FK or PFK from Child tables either. To delete them, you have to delete appropriate relationship.

In Toad Data Modeler, information on foreign keys can be found in:

- The **Attribute Properties** dialog | **Foreign Keys** tab where you can see details on particular FK.
- The **Relationship Properties** dialog where you set linking method between parent and child entities.

## Foreign Keys Mapping

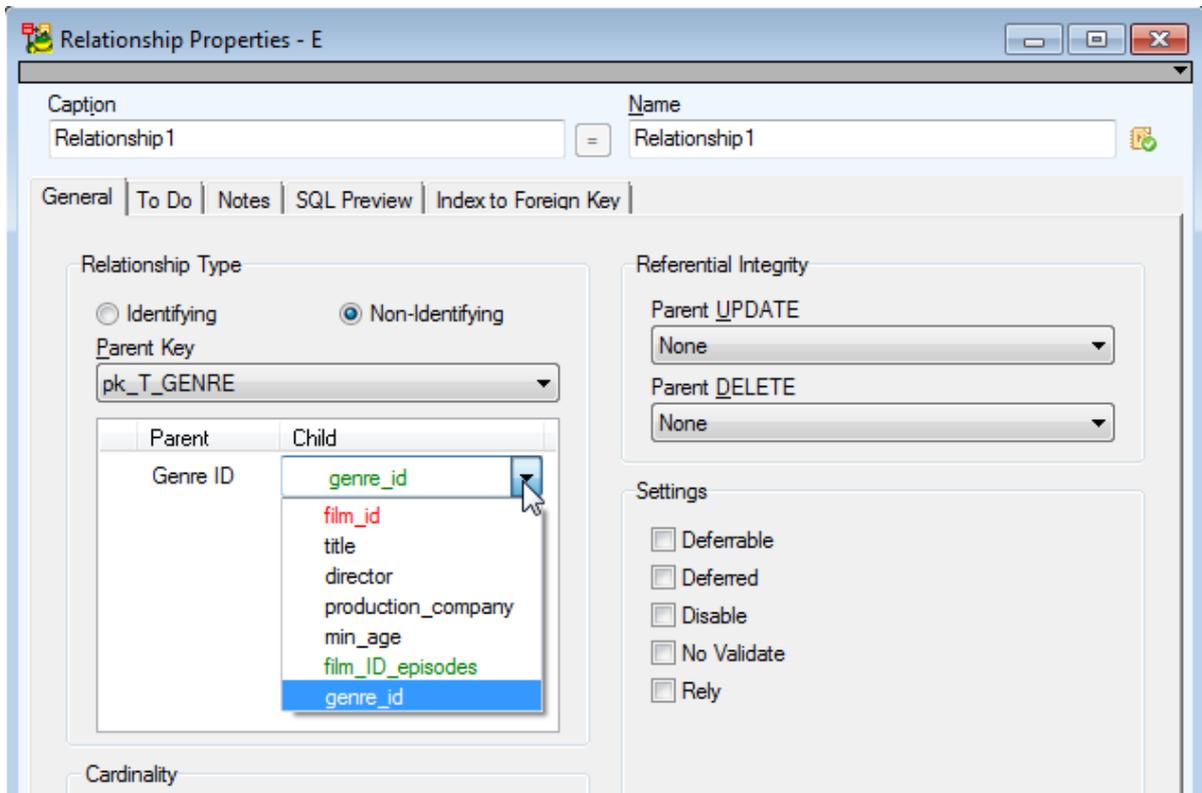
In Toad Data Modeler, you can control your foreign keys. Foreign keys mapping feature allows you to create Compound Keys, use existing keys etc.

### Example:

You have reverse engineered a model without relationships. In Toad Data Modeler, a new model has been created. However, information about foreign keys has been lost as in Toad Data Modeler stands that foreign keys cannot exist without relationships. Now you need to create relationships in Toad Data Modeler.

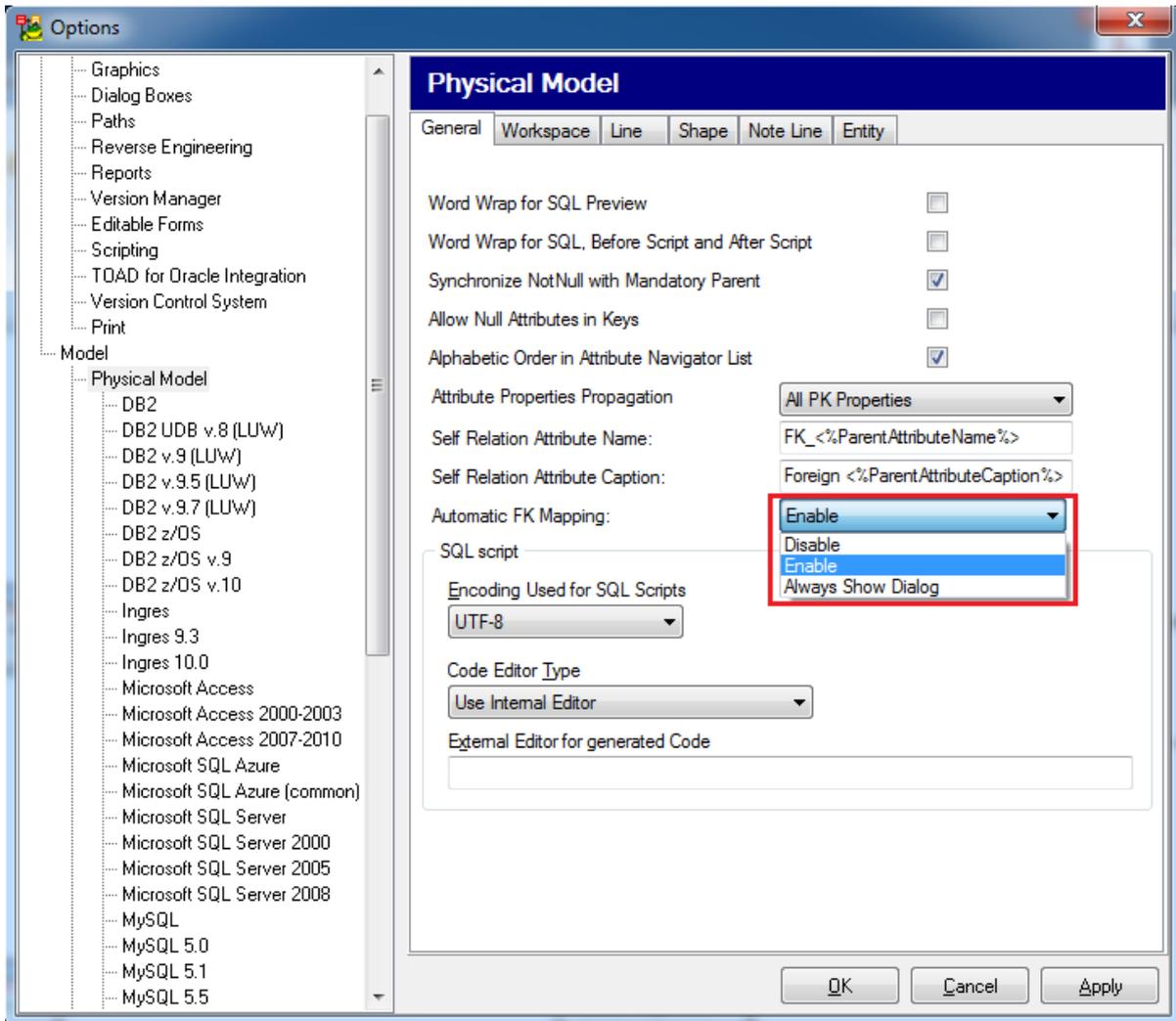
As soon as you create a new relationship, a new foreign key is added to the child entity automatically or existing attribute is mapped to primary key automatically.

In case you need to choose different column for the relationship, edit the relationship and choose the new column on tab **General**.

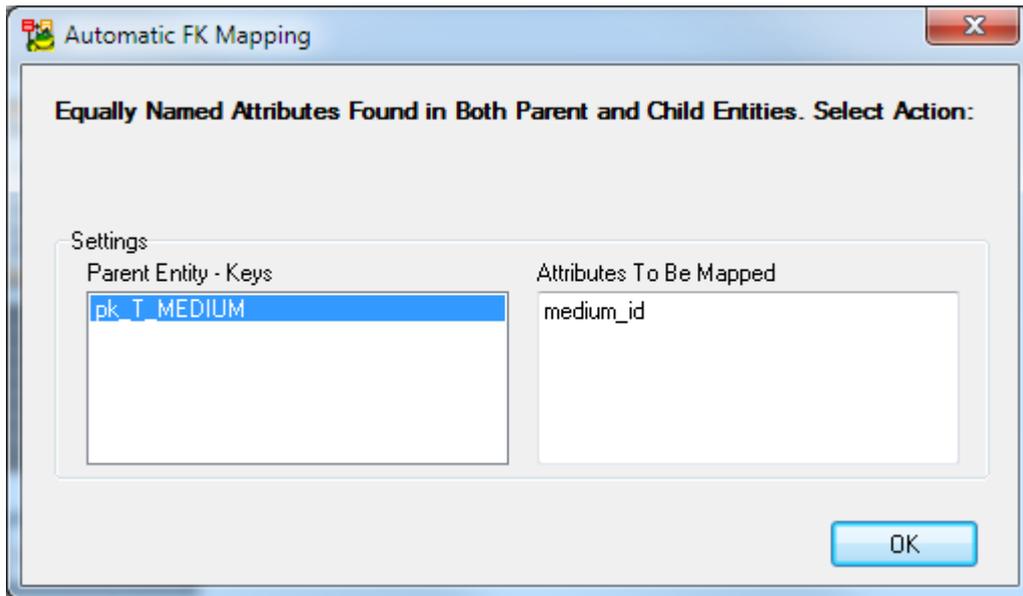


### Automatic FK Mapping

You can also set **Automatic FK Mapping** in **Settings | Options | Physical Model**.



- Disable—FK mapping is OFF
- Enable—Toad Data Modeler searches for matching attributes. If a single option is found, it gets mapped. If multiple options are found, a dialog appears.
- Always Show Dialog—The dialog opens even if only a single option is found.



## Parent Attributes (Rolenames)

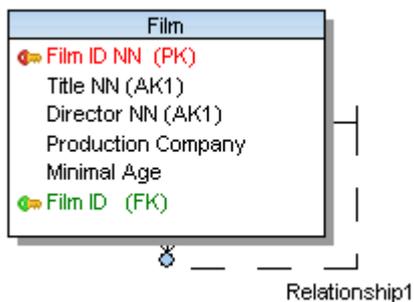
You can change the attribute name e. g. in the following cases:

- When you create a self-relationship.
- Whenever you want to change a Foreign Key attribute name.

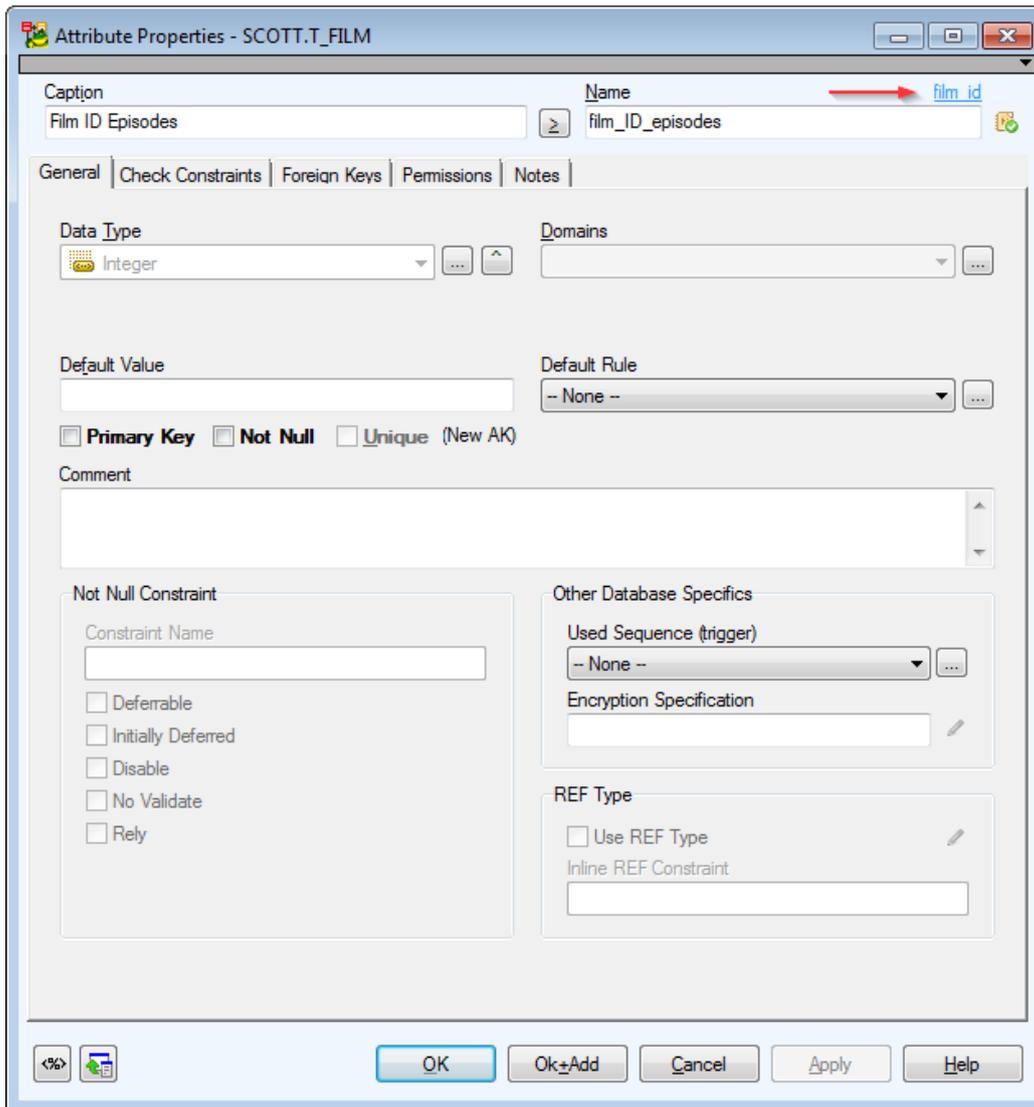
### Scenario:

You have just added a self-relationship to the *Film* entity.

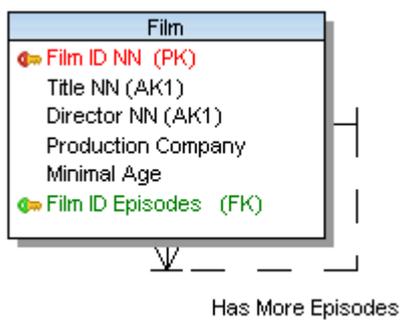
1. A copy of your identifying key attribute has been created in the entity - *Film ID*. Now you have two *Film ID* attributes in the *Film* entity.



2. Edit the newly created Film ID attribute, define a new **Name**. The link to parent attribute is displayed on top right-hand corner of the form.



3. Confirm **OK** and see the change on the Workspace.



## About Advanced Options

Toad Data Modeler supports several database systems. As there are differences among databases, there are only applicable to specific databases.

For example:

- Microsoft SQL Server 2000 and higher only support Dictionary Types, which are called User-Defined Data Types in database itself
- PostgreSQL databases do not support Users, only User Groups. Instead of Users, Roles are used for managing permissions.
- and others...

Database platform and version specific information are listed in the respective topics in the **Databases** chapter.

## Alternate Keys

You can create a relationship between a Parent and Child entities using:

- Key (Primary or Alternate Key) of parent entity
- Unique Index of parent entity

Alternate keys are used when you want to link two entities using two attributes. These two attributes make one unique item.

---

### Scenario

You would like to create a composite alternate key which contains *Title* and *Director* attributes in the *Film* entity.

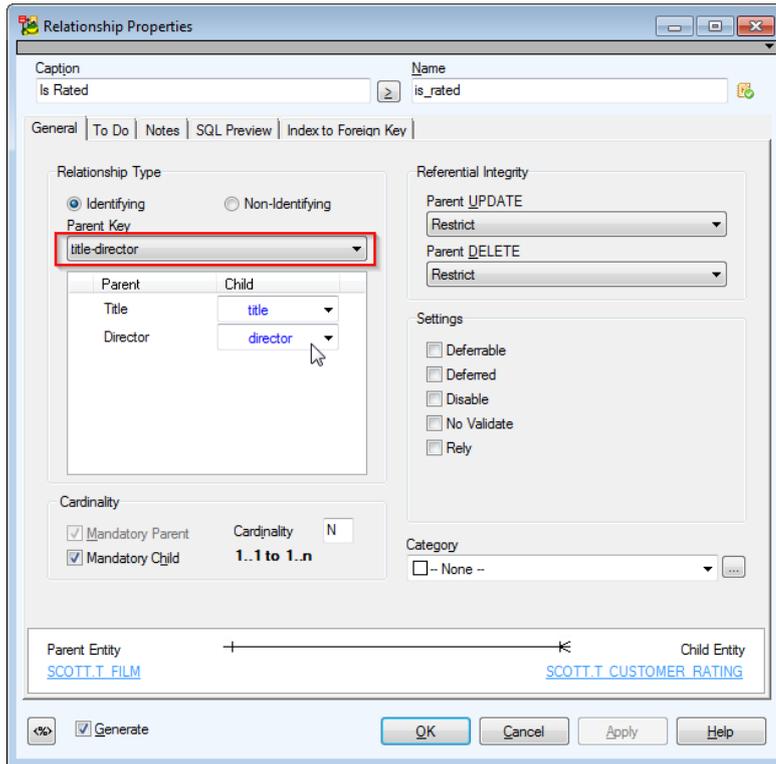
### Solution

1. Edit the *Film* entity.
2. In the **Entity Properties** form click the **Keys** tab. Here, primary keys and alternate keys are stored, and you can manage them easily using the Add, Edit and Delete buttons. Now click **Add** to create a new key and confirm by clicking **Apply**.
3. Edit the new key.
4. Define its properties on tab **General** and then switch to the **Attributes** tab.
5. From the window **Available**, select attributes *Title* and *Director* and click the **Add** arrow button to move them to the **Selected** window.
6. Set properties in other tabs as you wish (e.g. Notes) and confirm.

**Scenario:** After creating the alternate key, you would like to use it for linking entities.

### To select the alternate key for relationship

1. Double-click the relationship.
2. From the **Parent Key** box, select the alternate composite key *title-director*.



## Indexes

### To create an index

- Right-click an entity on Workspace and choose **Add | Index**. The **Index Properties** dialog opens.

or

- In the **Entity Properties** form, select the **Indexes** tab and click **Add**.

### To edit indexes

- In the **Entity Properties** dialog | **Indexes** tab, double-click the index or **Edit**.

Option	Description
	Above the Object Navigator Dropdown Menu, you can see name of entity that the index belongs to. Click the button on top right-hand corner to open the parent form (Entity Properties form).
Object Navigator Dropdown Menu	All indexes of selected entity are listed here. The box allows you to edit indexes quickly and comfortably from one place. Use buttons to change order of indexes. Use Ctrl + Up to move index upwards or Ctrl + Down to move it downwards.

Option	Description
<b>General Tab</b>	<b>Description</b>
Caption	Logical attribute name.
Name	Physical attribute name.
Schema	Schema selection.
Unique	Select this option to set the index as unique. Via unique indexes, you can link entities together. See <a href="#">Select Parent Key for Relationship</a> for more information on available linking methods.
Bitmap Index	Database dependent item (Oracle). Select this checkbox to define the index as bitmap index.
Generate	Select it to generate the index in final SQL (DDL) script. (It is selected by default.)

**i** Note: Other options on the **General** tab vary according to the database platform you're using. Options specific to your database can be found in the "Databases" chapter.

Items Tab	Option
Available	A list of all attributes of the entity.
Selected	Attribute(s) that have been assigned to the index.
<b>Notes Tab</b>	Tab for notes on the index.

Index Properties Tab	Description
Tablespace	Select a tablespace or click the button on the right to define a new tablespace.
Indextype	Index type definition
ODCI Parameters	ODCI parameters definition

### To display indexes on the Workspace

Right-click the WS, select **Workspace Format | Entity** tab and select the **Display Indexes** checkbox. See how indexes are displayed:

T_CUSTOMER
customer_id NN (PK)
name NN (IX1,IX2)
address NN (IX2)
i_name (IX1)
i_name_address (IX2)

**i** Note: Even when the indexes are not displayed, you can see which attribute belongs to which index (e.g. attribute *Name* is assigned to indexes *i\_name* and *i\_name\_address* - (IX1,IX2))

### To delete an index

In the **Entity Properties** dialog select the **Indexes** tab, choose the index and click **Delete**.

## Check Constraints

Check constraints can be created in the **Check Constraints** tab in **Entity Properties** form (for multiple column check constraints) or in the **Attribute Properties** dialog (for single column check constraint).

### To add a check constraint

In **Entity Properties** form, select the **Check Constraints** tab and click **Add**.

### To edit a check constraint

In **Entity Properties** form | **Check Constraints** tab, double-click the selected check constraint or press **Edit** .

General Tab	Description
Caption	Logical check constraint name
Name	Physical check constraint name
Check Constraint Rule	Select rule or click the button on the right to define a new rule.
Generate	Select it to generate the check constraint in final SQL (DDL) script (selected by default).
SQL Tab	Write SQL script for the check constraint here. See <a href="#">About Templates</a> for more information.
Notes Tab	Space for your notes on the check constraint.

**i** Note:

- To copy a check constraint, press CTRL and drag the constraint over the **Check Constraints** folder of a target entity in **Model Explorer**.
- To move a check constraint, drag it over the **Check Constraints** folder of a target entity in **Model Explorer**.
- To delete a check constraint, select it and click **Delete** in the **Check Constraints** tab of the **Entity Properties** form.

# Triggers

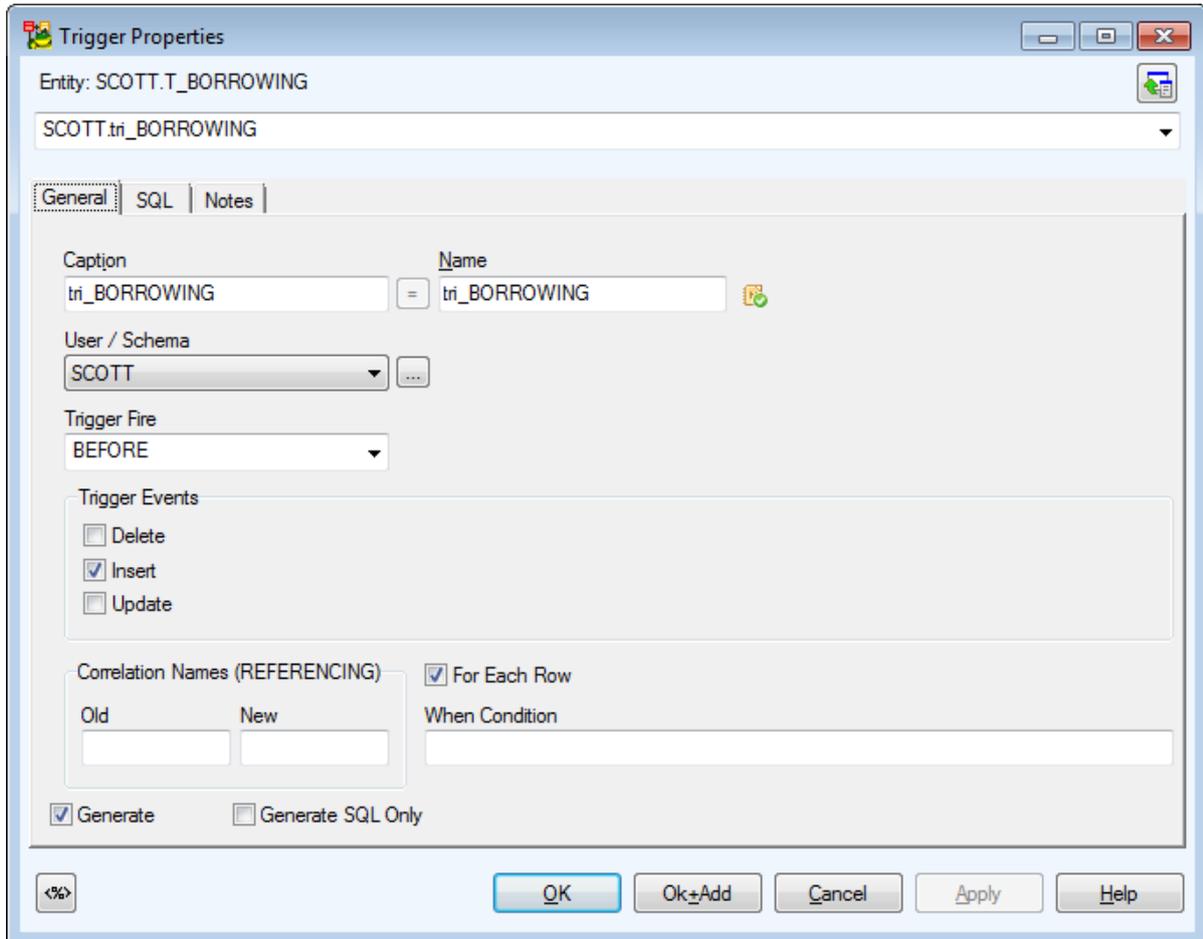
## To add a trigger

In **Entity Properties** form, select the **Triggers** tab and click **Add**.

## To edit a trigger

In **Entity Properties** form, **Triggers** tab, double-click the selected trigger or press **Edit**.

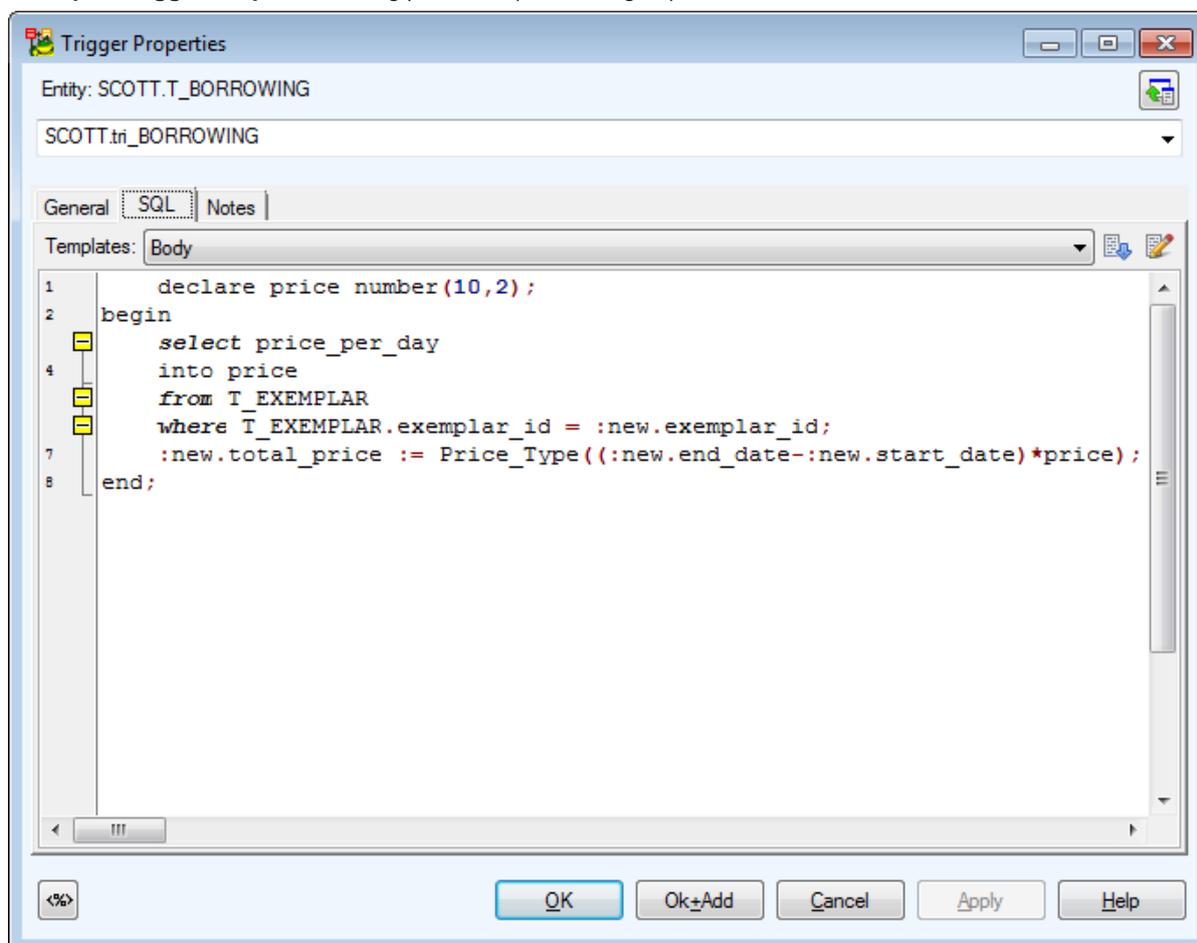
**Example: Trigger Properties dialog (Oracle 10g db)**



Option	Description
	Above the Object Navigator Box, you can see name of entity the trigger belongs to. If you click the button in top right-hand corner, the parent form will open ( <b>Entity Properties</b> in this case).
General Tab	Description
Caption	Logical trigger name
Name	Physical trigger name

Option	Description
Schema	Schema selection box
Trigger Fire	<b>Before, After</b> (database dependent) - select a trigger fire.
Trigger Events	<b>Delete, Insert, Update</b> - select a trigger event.
Generate	Select to generate the trigger in final SQL (DDL) script (selected by default.)
Generate SQL Only	Select to generate the SQL code written in tab <b>SQL</b> only.
<b>SQL Tab</b>	Write SQL script for the trigger here. <a href="#">About Templates</a>
<b>Notes Tab</b>	Space for your notes on the trigger.

**Example: Trigger Properties dialog | SQL tab (Oracle 10g db)**

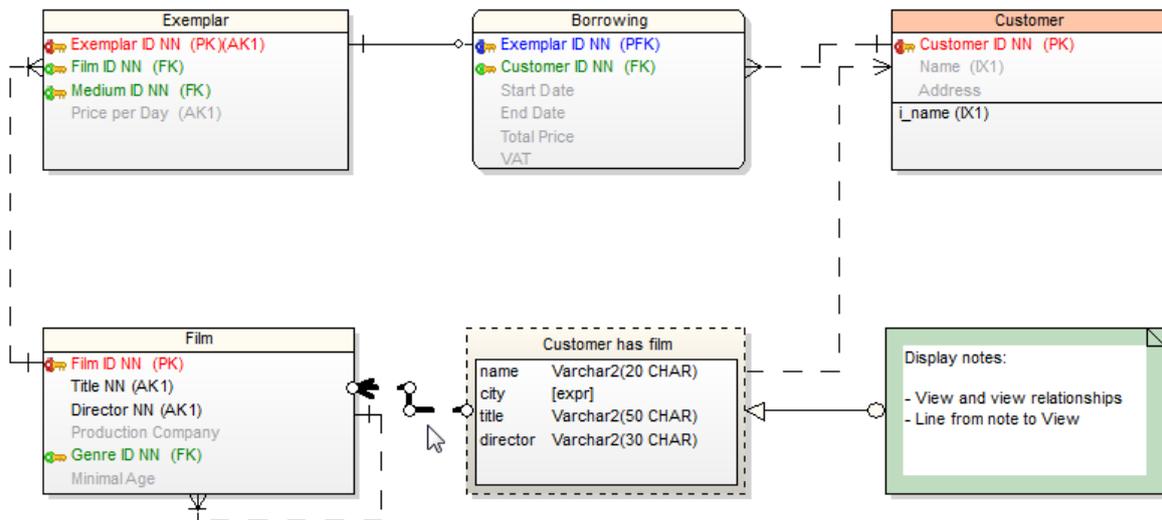


**i** Note:

- To copy a trigger, press CTRL and drag the trigger over the **Triggers** folder of a target entity in **Model Explorer**.
- To move a trigger, drag it over the **Triggers** tab (folder) of a target entity in **Model Explorer**.
- To delete a trigger, select it and click **Delete** on the **Triggers** tab in the **Entity Properties** form.

## Views

Toad Data Modeler allows you to display and model views in your ER diagrams visually.



You can manage Views via:

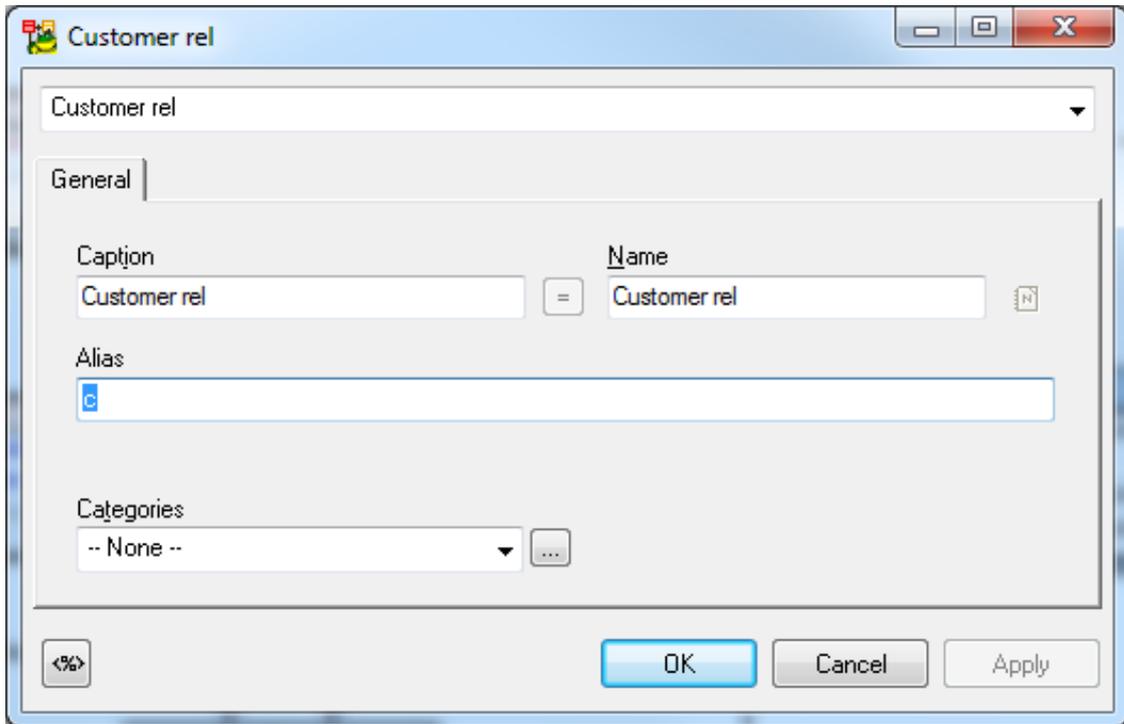
- **Model** menu | **Model Items** | **Views**
- **Model Explorer** | **Views** folder

View properties and options are database dependent.

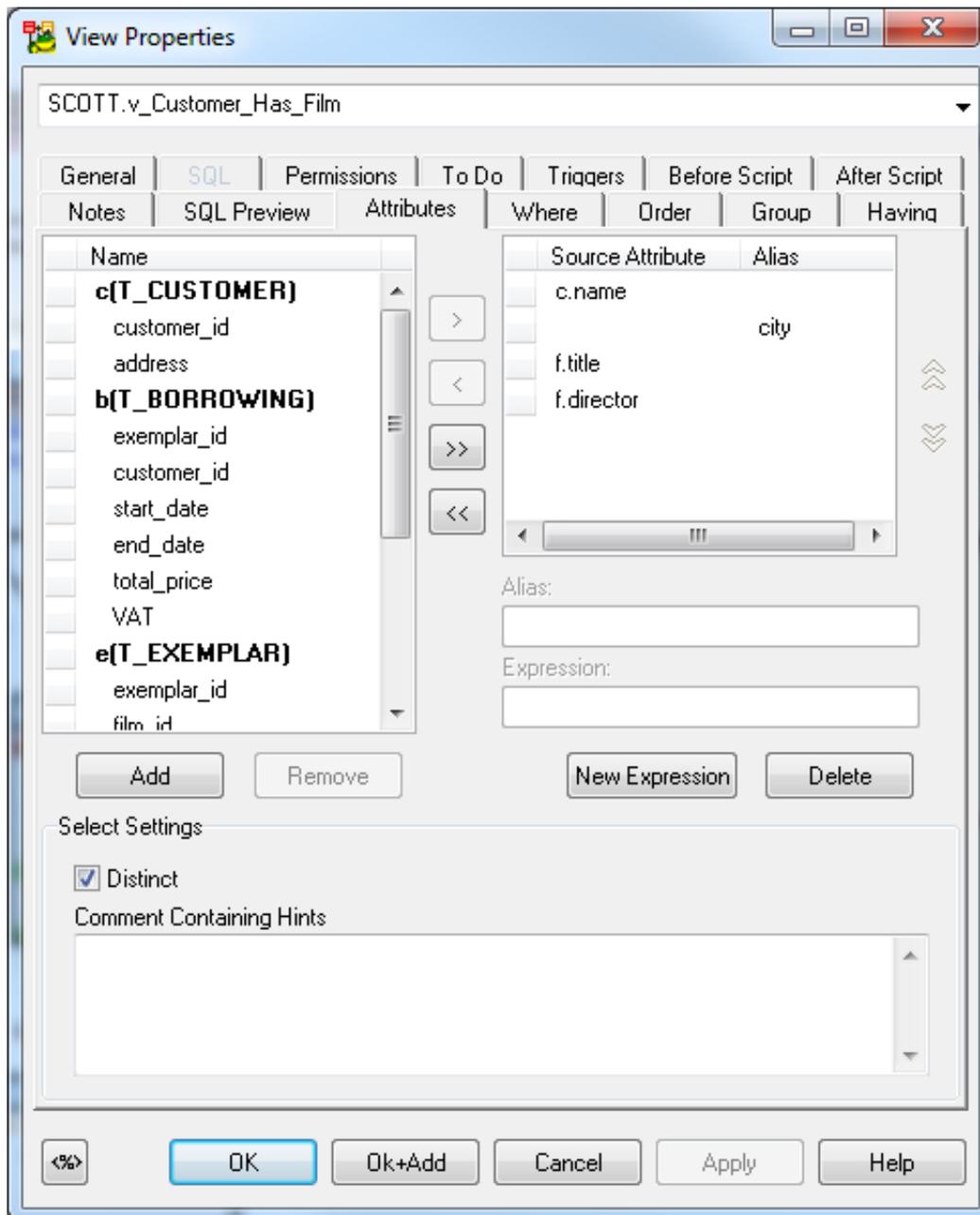
### To create a new view

1. Click  on the toolbar and then click the workspace.
2. Click  on the toolbar and draw view relationships between some existing table or view and your newly created view.

3. Double click the view relationship line to edit it. In Alias field you can specify a new alias name for the linked table.



4. Double-click the view to edit it.
5. Define properties on **General** tab and other tabs.
6. On tab **Attributes** you can select columns from the linked tables. Relationship aliases and names of entities are displayed.



7. You can use a template on **SQL** tab of the view.

**i** **NOTE:** As well as for entities and relationships, you can create shortcuts of views on the workspaces of your model. List of view shortcuts can be found in **Model Explorer** | in particular Workspace folder and in the **Views** folder | **Shortcuts**.

**i** **NOTE:** There are some limitations for modeling and reverse engineering of views in Toad Data Modeler. In the following cases views are loaded/modeled as "Select in Views as Text" without view relationships:

- When JOIN is used inside FROM statement. CROSS JOIN is the only supported type of JOIN.
- When a simple format for a name of a table is not used in FROM or when anything follows after a name of a table, e.g. functions, subquery, partition\_extension\_clause, PIVOT, UNPIVOT, DBLINK, flashback\_query\_clause, and row\_pattern\_clause.
- When "WITH common\_table\_expression", UNION, EXCEPT, INTERSECT, MINUS, model\_clause, and hierarchical\_query\_clause are used in view.
- When another definition exists between ORDER BY and the end of view, e.g. FOR clause in SQL Server.

## Materialized Views

Materialized Views are supported in the following databases: Oracle, DB 2, DB2 zOS, , Sybase SQL Anywhere 11, Teradata 13.

Toad Data Modeler allows you to display materialized views graphically in your ER diagram.

### To add a materialized view

Click  on the toolbar and then click anywhere on the work area.

or

**Model Explorer** | Right-click the **Materialized Views** folder | **Add Materialized View**.

### To edit a materialized view

Double-click the materialized view on the Workspace.

or

Edit the materialized view in **Model Explorer** | **Materialized Views** folder | double-click the selected materialized view (or right-click | **Edit**).

Option	Description
Object Navigator Box	All materialized views of your model are listed here. The combo-box allows you to edit them, one by one from one place. Use <b>Apply</b> to confirm all the changes you make.
<b>General Tab</b>	<b>Description</b>
Caption	Logical materialized view name
Name	Physical materialized view name
Schema	Schema selection box

Option	Description
Categories	Category selection box
Generate	Select it to generate the materialized view in final SQL (DDL) script.
Generate SQL only	Select it to generate only the SQL code written on tab <b>SQL</b> . The rest of items will be ignored.
<b>SQL Tab</b>	Write SQL code of materialized view subquery on this tab. <a href="#">About Templates</a>
<b>Permissions Tab</b>	Here you can assign Users or User Groups permissions to the materialized view.
<b>To Do Tab</b>	On this tab, you can write some tasks on the selected materialized view.  Note: To see all To Do tasks, select <b>Model   To Do</b> .
<b>Before Script Tab</b>	Whatever you write here, it will be generated before the materialized view definition.
<b>After Script Tab</b>	Whatever you write here, it will be generated after the materialized view definition.
<b>Notes Tab</b>	Tab for notes on particular materialized view. The text written here will display in a pop up dialog when you point your mouse cursor at the materialized view shortcut in your ER diagram.
<b>SQL Preview Tab</b>	Click  at the bottom of this tab to see the part of SQL code for the materialized view.
<b>Refresh Tab</b>	Use this tab to specify the default methods, modes and times for the database to refresh the materialized view.
<b>Physical Properties Tab</b>	Define storage characteristics of materialized view on this tab.
<b>Materialized Views Properties Tab</b>	Define other materialized view characteristics on this tab.
<b>Create Index Tab</b>	Create index sentences are written on this tab.

## Procedures

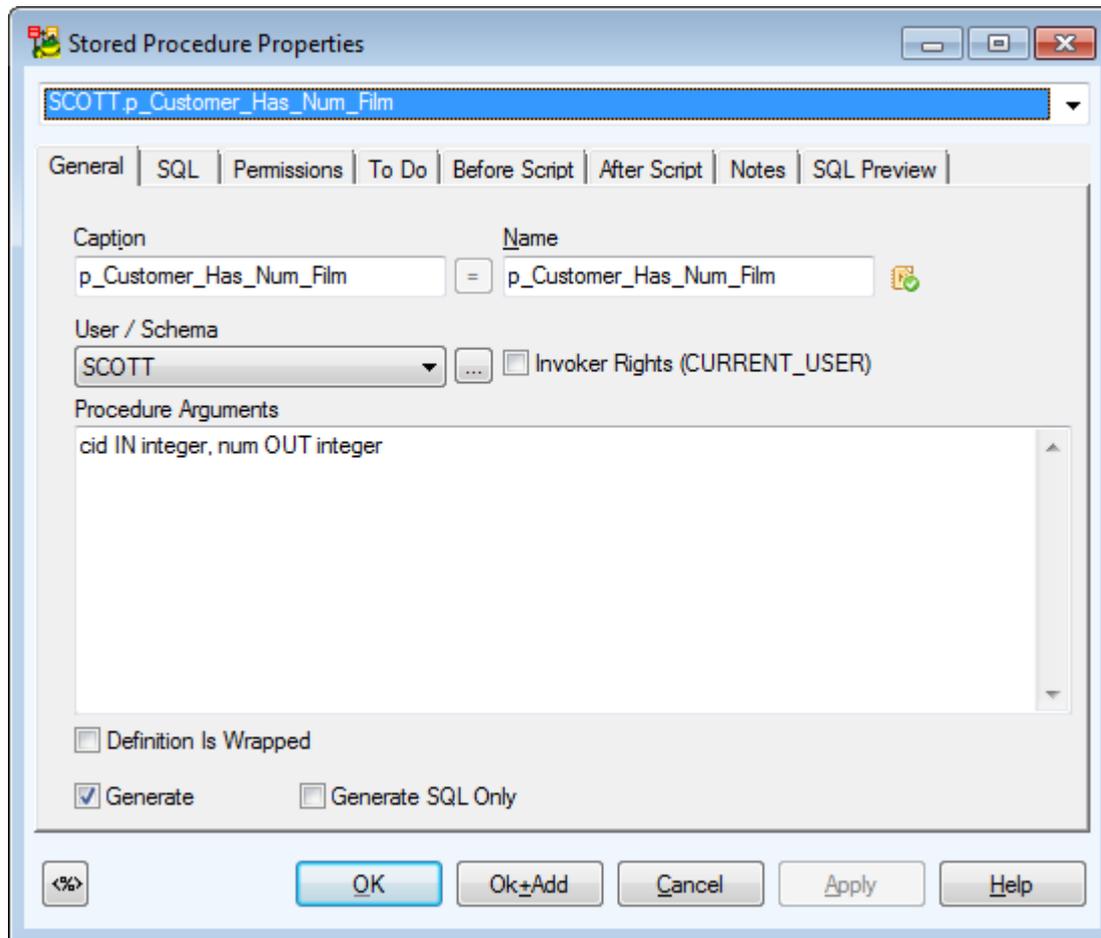
### *To add a procedure*

Right-click the **Procedures** item in **Model Explorer | Add Procedure**.

### To edit a procedure

Double-click the selected procedure in **Model Explorer | Procedures** (or right-click **Edit**).

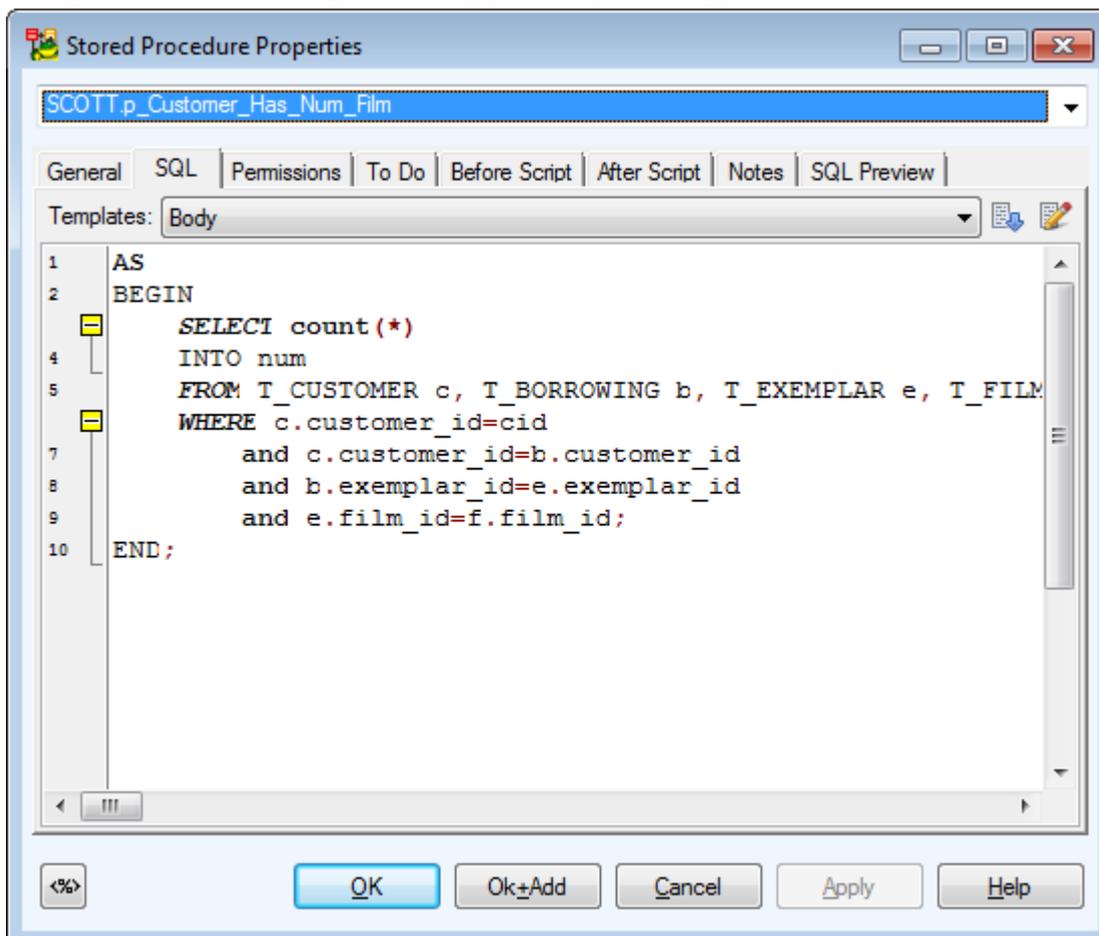
**Example:** The **Procedure Properties** dialog (Oracle 10g)



General Tab	Description
Caption	Logical procedure name
Name	Physical procedure name
Schema	Schema selection box
Procedure Arguments	Define procedure arguments here.
Invoker Rights	Database dependent option. See particular database reference for more information.
Definition is Wrapped	Database dependent option. See particular database reference for more information.

<b>Generate</b>	Select it to generate the procedure in final SQL (DDL) script.
<b>Generate SQL only</b>	Select it to generate only the SQL code written on tab <b>SQL</b> . The rest of items will be ignored.
<b>SQL Tab</b>	SQL code. (See the example in the following screenshot.) <a href="#">About Templates</a>
<b>Permissions Tab</b>	On this tab, you can manage permissions to particular procedure.
<b>To Do Tab</b>	Here you can write some tasks on the selected procedure. <b>Note:</b> To see all To Do tasks, select <b>Model   To Do</b> .
<b>Before Script Tab</b>	Whatever you write here, it will be generated before the Store Procedure definition.
<b>After Script Tab</b>	Whatever you write here, it will be generated after the Store Procedure definition.
<b>Notes Tab</b>	Tab for notes on particular procedure.

**Example:** The **Procedure Properties** dialog | **SQL** tab (Oracle 10g)



# Compare Procedures in Sync & Convert Wizard

When the **Sync & Convert Wizard** shows differences between procedures (on page **Select Items**), you can double-click the SQL item of procedures to display details about differences between them.

## Functions

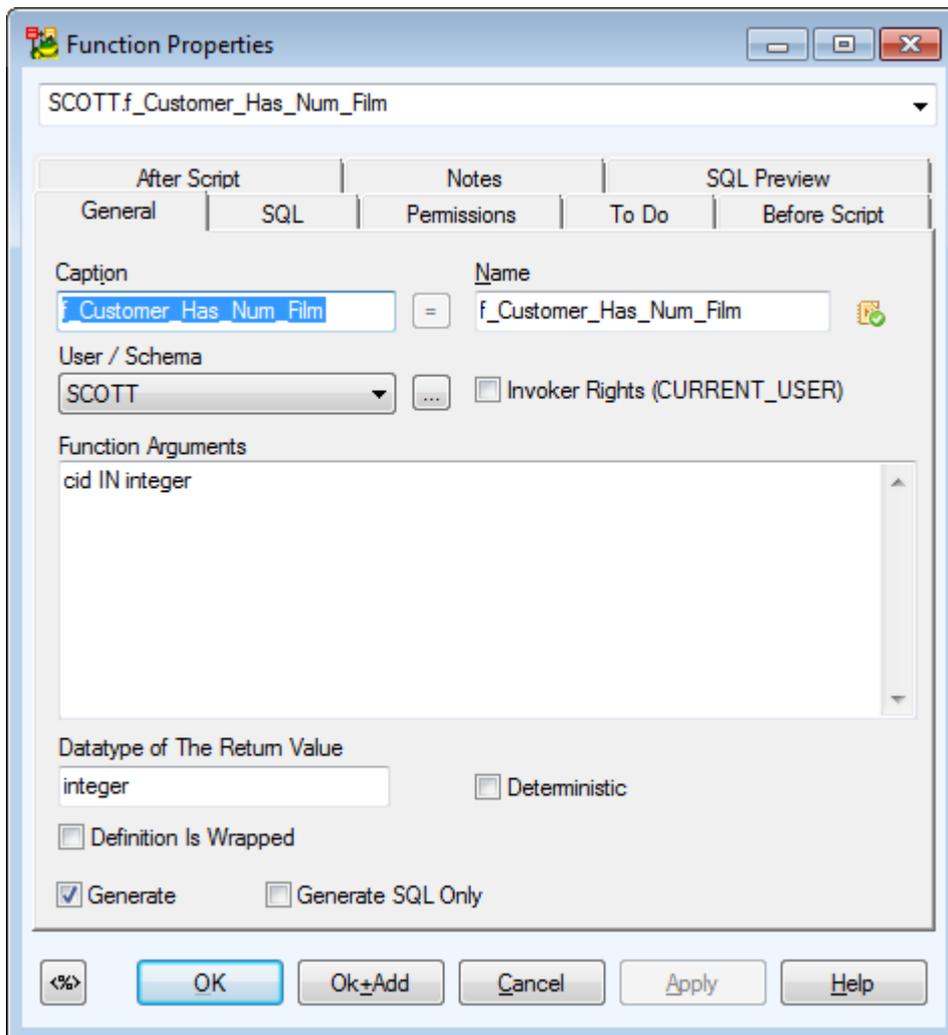
### To add a function

Right-click the **Functions** item in **Model Explorer** | **Add Function**.

### To edit a function

Double-click the selected function in **Model Explorer** | **Functions** (or right-click **Edit**).

**Example:** The **Function Properties** dialog (Oracle 10g)

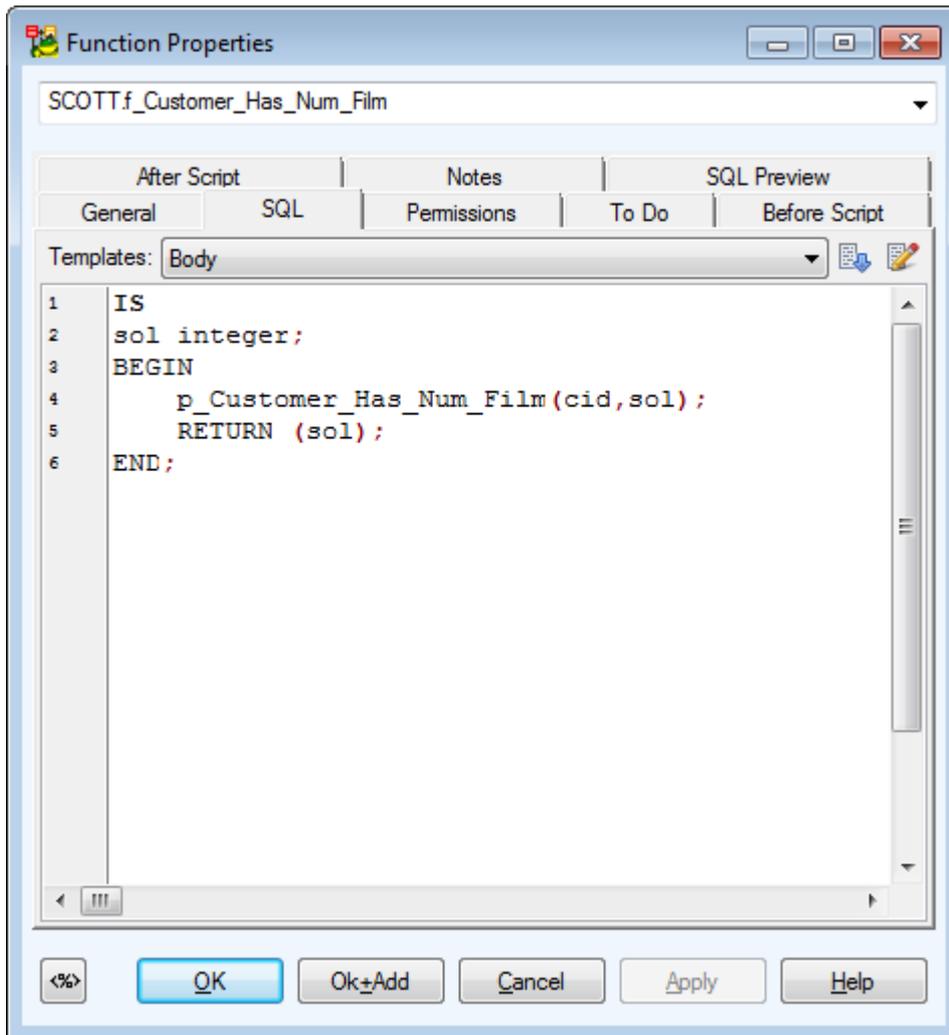


**General Tab**

**Description**

Caption	Logical function name
Name	Physical function name
Schema	Schema selection box
Invoker Rights (CURRENT_USER)	Database dependent option. See particular database reference for more information.
Function Arguments	Write function arguments here.
Datatype of the Return Value	Define data type of the return value.
Deterministic	Database dependent option. See particular database reference for more information.
Definition is Wrapped	Database dependent option. See particular database reference for more information.
Generate	Select it to generate the function in final SQL (DDL) script.
Generate SQL only	Select it to generate only the content of the <b>SQL</b> tab. All other items will be ignored.
<b>SQL Tab</b>	SQL code. (See the example in the following screenshot.) <a href="#">About Templates</a>
<b>To Do Tab</b>	On this tab, you can write some tasks on the selected function.
<b>Permissions Tab</b>	On this tab, you can assign a User or User Group permissions for selected function.
<b>Before Script Tab</b>	Whatever you write here, it will be generated after the function definition.
<b>After Script Tab</b>	Whatever you write here, it will be generated before the function definition.
<b>Notes Tab</b>	Tab for notes on particular function.

**Example:** The **Function Properties** dialog | **SQL** tab (Oracle 10g)



## Schemas

### To add a schema

Select **Model** | **Model Items** | **Schemas** and click **Add** in the **Schemas** dialog.

### To edit a schema

Select **Model** | **Model Items** | **Schemas** and double-click the selected schema or click **Edit**.

**i** Note: You can also edit/rename/delete schemas in **Model Explorer** | **Schemas** folder | Right-click the selected schema.

## Users

Toad Data Modeler allows you to define Users and assign them to User Groups. Later you can assign the Users and User Groups permissions to particular objects.

### To add a user

Click  on the toolbar and click **Add** in the **Users** dialog.

or

In **Model Explorer** | Right-click the **Users** folder | **Add User**.

### To edit a user

Select **Model** | **Model Items** | **Users** | double-click the selected user or click **Edit**.

or

In **Model Explorer** | **Users** folder | double-click the selected user or right-click | **Edit**.

Option	Description
Object Navigator Box	All users of your model are listed here. The combo-box allows you to edit them, one by one from one place. Use <b>Apply</b> to confirm all the changes you make.

General Tab	Description
Caption	Logical user name
Name	Physical user name
Password	User's password

#### Membership Tab

On this tab, you can assign a User to User Group.

 Note: One user can be assigned to more User Groups.

Select a User Group and click appropriate button:

 Adds selected item(s).

 Puts back selected item(s).

 Adds all items.

 Puts back all items.

**To Do** On this tab, you can write some tasks on particular user.

### To assign users to user groups

**Example:** All admins should be assigned to user group *Administrators*.

## Method A - via the Users dialog

1. In the **Users** dialog, select the *SCOTT* user and click **Edit**.
2. In the **User Properties** dialog, click the **Membership** tab.
3. Select the *Administrators* user group and click the **Add** arrow button to shift the selected group to the window **Selected**.  
(The User Groups have already been defined in **Model | Model Items | User Groups**.)

**i** TIP: If you confirm **Apply**, the **User Properties** dialog will remain opened, and you can comfortably assign other users to user groups. - Simply select another user from the object navigator box at the top.

## Method B - via the User Groups dialog

1. Click  on the toolbar.
2. From the **User Groups** dialog, select *Administrators* user group and click **Edit**.
3. In the **User Group Properties** dialog, click the **Members** tab.
4. Select *SCOTT* and click the **Add** arrow button to shift the selected user to the window **Selected**.

# User Groups

Toad Data Modeler allows you to define **Users** and assign them to **User Groups**. You can then assign User and User Groups permissions to objects.

### To add a user group

Click  on the **Users Toolbar** and click **Add** in the displayed dialog.

or

In **Model Explorer** | Right-click the **User Groups** folder | **Add User Group**.

### To edit a user group

Go to **Model Menu | Model Items | User Groups** | double-click the selected user group or click **Edit**.

or

In **Model Explorer** | **User Groups** folder | double-click the selected user group or right-click | **Edit**.

Option	Description
Object Navigator Box	All user groups of your model are listed here. The combo-box allows you to edit them, one by one from one place. Use <b>Apply</b> to confirm all the changes you make.
<b>General Tab</b>	<b>Description</b>
Caption	Logical user group name
Name	Physical user group name
<b>Membership Tab</b>	<b>Description</b>

Option	Description
Available	Available user groups
Selected	Selected user groups
Members Tab	Description
User Groups section	You can create another group in already existing group. (This is possible for some databases.) From the list of existing groups, select a group and click the <b>Add</b> arrow button.
Users section	Here, you can assign users to a group. <b>Note:</b> You can assign users to a user group also in the <b>Users</b> dialog.
To Do Tab	On this tab, you can write some tasks on the selected user group. <b>Note:</b> To see all To Do tasks, select <b>Model   To Do</b> .

**i** Note:

1. To copy user groups, use CTRL + Drag&Drop techniques.
2. To move user groups, use Drag&Drop techniques.  
You can copy and move your user groups within a model and between models of the same and different databases:
  - In **User Groups** dialog (**Model Menu | Model Items | User Groups**)
  - In **Model Explorer | User Groups** folder
  - Between **Model Explorer** and **User Groups** dialog
3. To delete user groups, go to:
  - **Model Menu | Model Items | User Groups** | Select a user group and click **Delete**.
  - **Model Explorer | User Groups** folder | Right-click and select **Delete Item**.

## Permissions

In Toad Data Modeler, you can assign permissions to the following objects:

- Entity
- Attribute
- User Data Type
- View
- Procedure
- Schema
- Users and User Groups.

This list is dependent on your current database platform and version. For example, some databases do not support assigning permissions to Users.

For every object, different permissions can be set (SELECT, INSERT, UPDATE etc.), depending on current database platform.

Options for permissions are described in the following example. Permissions for attributes, user data types etc. are set in the **Properties** dialog of particular object | **Permissions** tab (e.g. **Attribute Properties** | **Properties**).

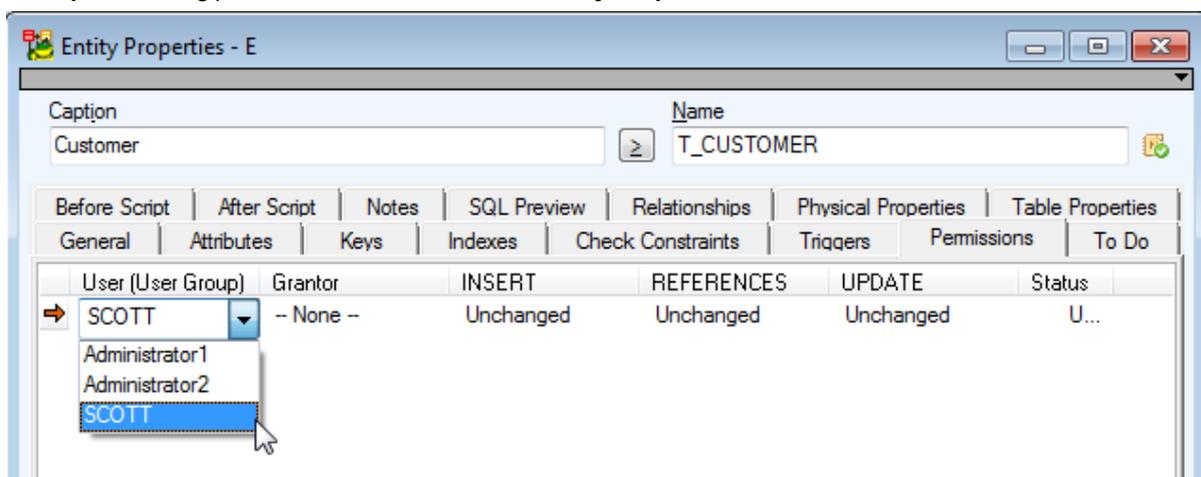
## Add Permissions

To be able to add a Permission, **it's necessary to define a User or User Group**. In case you forget, Toad Data Modeler allows you to define them directly from any **Permissions** tab of a particular object - click **Users** or **User Groups**.

### To add a permission for entity

In the **Entity Properties** form, select the **Permissions** tab and click **Add**.

**Example:** Creating permission for user *SCOTT* in **Entity Properties** form:



If you want to change the permission User or User Group, select the permission **User (User Group)** column, **press F2** and choose from the list.

**i** TIP: This kind of editing properties is usable anywhere in Toad Data Modeler. For more information, see [Inplace Editor](#).

### Permissions Tab Description

User (User Group)	Name of user (group) that the permission has been assigned to.
Grantor	Name of user (group) that assigns the permission.
Permissions:	SELECT, INSERT, UPDATE, DELETE, RULE, REFERENCES, TRIGGER

### To edit a permission

In the **Entity Properties** form | **Permissions** tab, double-click the selected permission or press **Edit**.

### General Tab Description

Permissions	List of all available permissions to a specific object.
Status	Shows if the particular permission has been assigned or not. <b>Unchanged</b> - No change has been made. <b>Grant</b> - Permission has been granted. <b>Deny</b> - Permission has been denied. (E.g. in Microsoft SQL 2005 models.)
with Grant Option	<b>Yes/No</b> - Determines if the permission User (Group) can assign the permission to another User (Group).

## About Universal Data Model

Universal (Generic Relational) Data Model allows you to model entity relationship diagrams without the necessity to select target database platform and database specific items.

## Specifics of Universal (Generic Relational) Data Model

- In Universal Data Models PK attributes migrate from parent to child entities as PFK or FK attributes.
- It is necessary to convert model to target database platform if you decide to generate SQL for your Universal Data .Model
- Reports can be generated in various formats.
- Reverse engineering via ADO/ODBC is available.

## About Logical Data Modeling

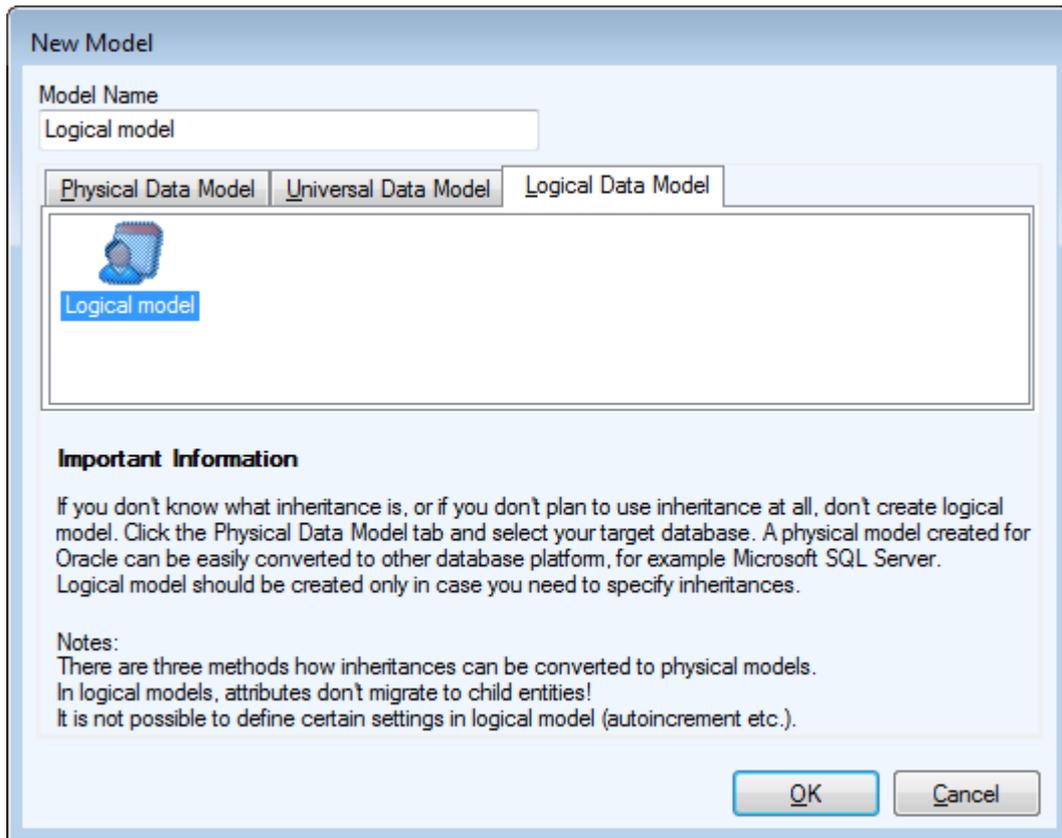
Toad Data Modeler allows you to design and maintain a logical model giving a complete picture of the business area. Logical model is independent of the database platform and is much simpler than physical model. It uses objects such as inheritance, valid values or M:N relationships. From the Logical ER (LER) diagram, you can build a Physical ER (PER) diagram of the selected database platform (LER to PER conversion).

### **i** Note:

- This topic contains information on objects and functions that are specific for logical modeling. General information and other features applicable in Logical model as well as in Physical model are contained and described in other sections of this Help file. See e.g. the "Model Objects" chapter, "Model Explorer" topic etc.
- See the sample logical model *Employee* that is included in the installation package for Toad Data Modeler. Default location is: C:\Program Files\Quest Software\Toad Data Modeler\Samples.

### **To create a logical model**

Select **File | New | Model | Logical Data Model** tab.



## Benefits of Logical Data Model

Logical Data Model allows you to model inheritances in entity relationship diagrams. Universal Data Model and Physical Data Model do not support this feature.

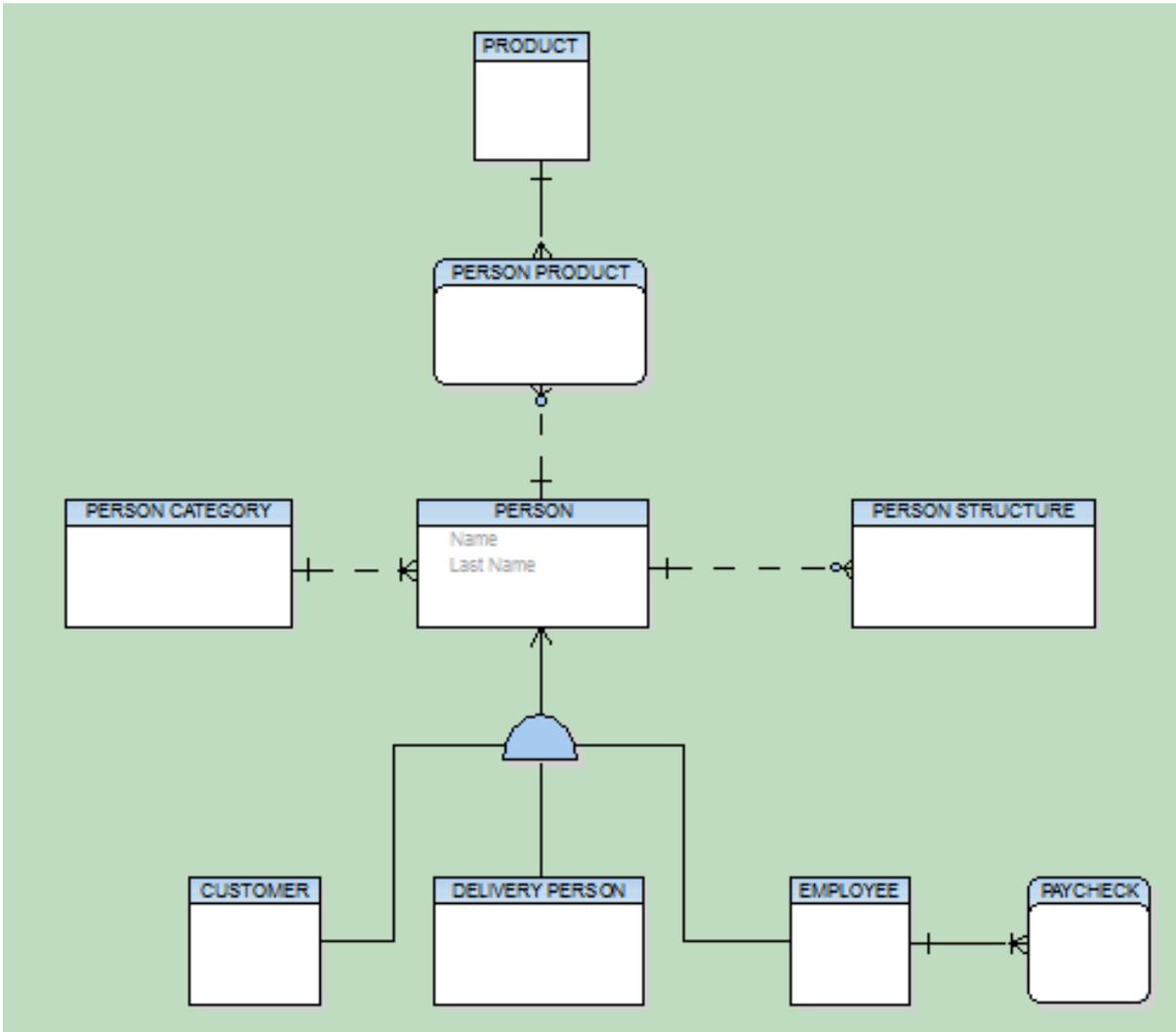
## Specifics of Logical Data Model

- There are three methods how inheritance can be resolved when converting to physical model.
- Attributes do not migrate to child entities.
- It is not possible to define database specific items in Logical Model, for example sequences/autoincrements etc.
- You can define Valid Values in logical model (will be converted to physical model).

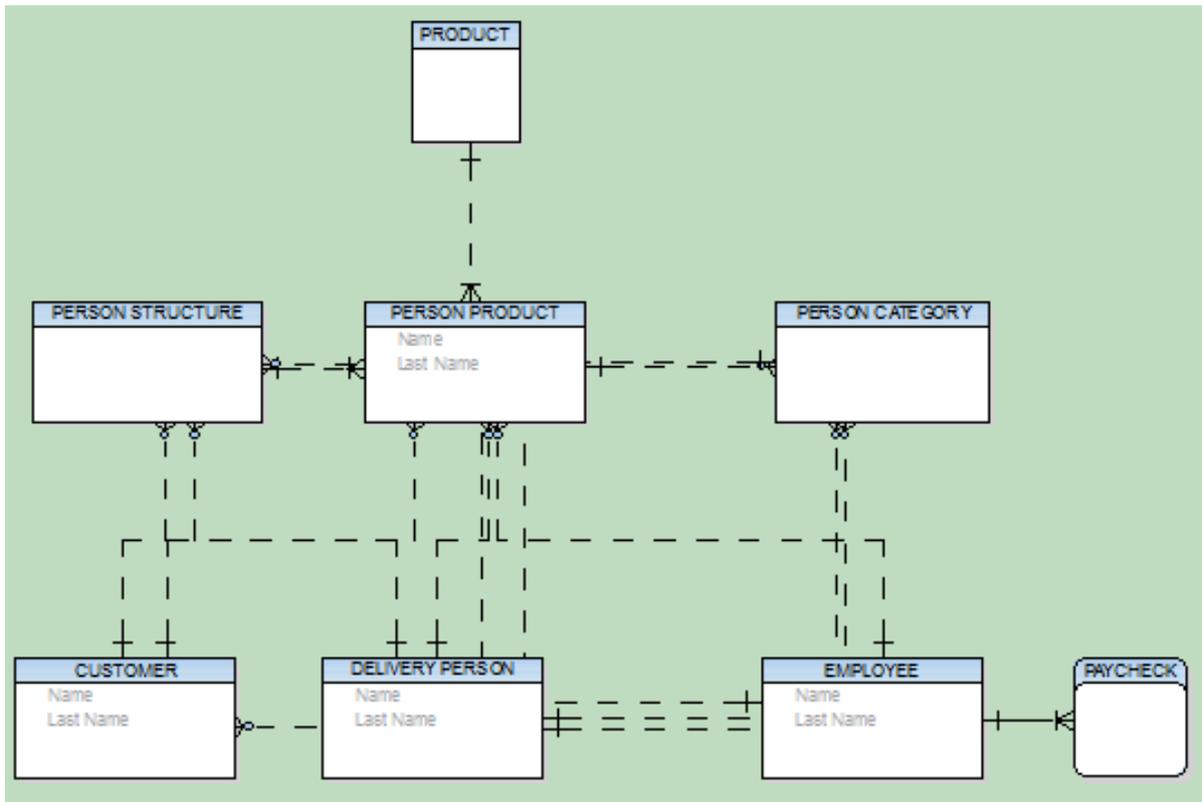
## Benefits of Super and Sub Types

In Logical Model you can define database structure in various ways. See the pictures below. Both of them show a structure modeled in Logical Model and both the models will result in the identical output when converted to Physical Model. The difference is that Model A uses Super and Sub Types while Model B doesn't use inheritance at all.

# Model A - Utilizing Super and Sub Types



## Model B - Lacking Super and Sub Types



This example shows:

1. That you can create logical models in different ways and achieve the same result after conversion to physical model.
2. That you can be more productive when using inheritance. For example, you only need to change the Last Name attribute once in the first model. Without using inheritance in model B, you have to change it four times.
3. How much “readable” the first model is compared to the second one.
4. How important is to select appropriate inheritance resolution when converting your logical model to physical.
5. That creating logical models without inheritance has minimal benefits compared to using inheritance.

## Disadvantage

The main disadvantage of logical modeling is that direct synchronization with existing database **is not possible**. Only physical models of specific database platforms and versions may be synchronized with an existing database. Therefore, if you want to synchronize your logical model, you need to convert it to the physical model first.

## Objects in Logical Model

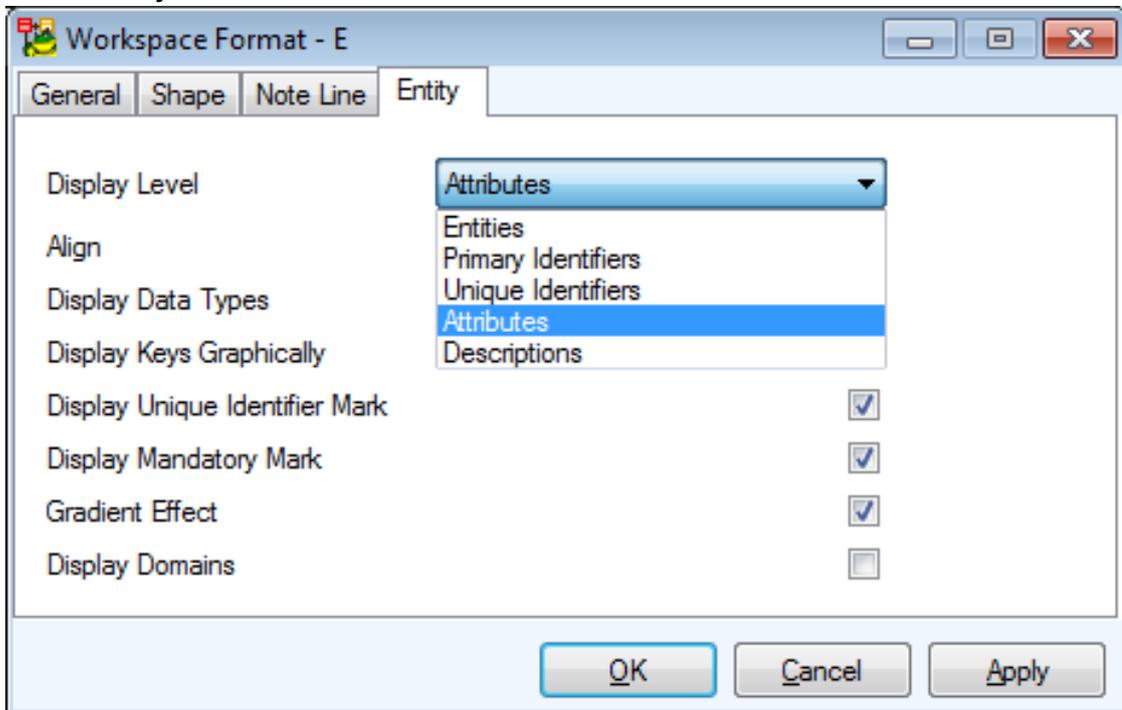
- Entity
- Relationship (Identifying and non-identifying)
- Inheritance
- Stamp
- Category (Caption of Categories)

All these objects are available in the **Objects** menu | **Add New** or via icons in the Designer toolbar.

**i** Note: Besides these objects you can add also other graphical shapes to your ER diagram. [2-D Shapes](#)

## Format Logical Objects

1. Right-click the Workspace in your logical model and select **Workspace Format**.
2. Click the **Entity** tab.



**Display Level** - Select items you need to display in your logical model - Entities, Primary Identifiers, Unique Identifiers, Attributes, Descriptions.

See the **Display Level** box on the toolbar (also **View** menu | **Display Level**).

### **To define colors for attributes on your Workspace**

Select **Settings** | **Options** | **Model** section | **Logical Model** | **Entity** tab | **Attribute Colors** area.

Logical model

General

Shape

Note Line

Entity

Verification

**Default**

Display Level	Attributes ▼
Align	<input checked="" type="checkbox"/>
Display Data Types	<input type="checkbox"/>
Display Keys Graphically	<input checked="" type="checkbox"/>
Display Unique Identifier Mark	<input checked="" type="checkbox"/>
Display Mandatory Mark	<input checked="" type="checkbox"/>
Gradient Effect	<input checked="" type="checkbox"/>
Display Domains	<input type="checkbox"/>

**Attribute Colors**

Primary Unique Identifier	<div style="display: flex; align-items: center;"> <div style="width: 15px; height: 15px; background-color: red; margin-right: 5px;"></div> <span>Red</span> <span style="margin-left: 10px;">▼</span> </div>
Mandatory Attribute	<div style="display: flex; align-items: center;"> <div style="width: 15px; height: 15px; background-color: black; margin-right: 5px;"></div> <span>Black</span> <span style="margin-left: 10px;">▼</span> </div>
Optional Attribute	<div style="display: flex; align-items: center;"> <div style="width: 15px; height: 15px; background-color: gray; margin-right: 5px;"></div> <span>Medium Gray</span> <span style="margin-left: 10px;">▼</span> </div>

## Edit Entities

- Double-click the entity on the Workspace.

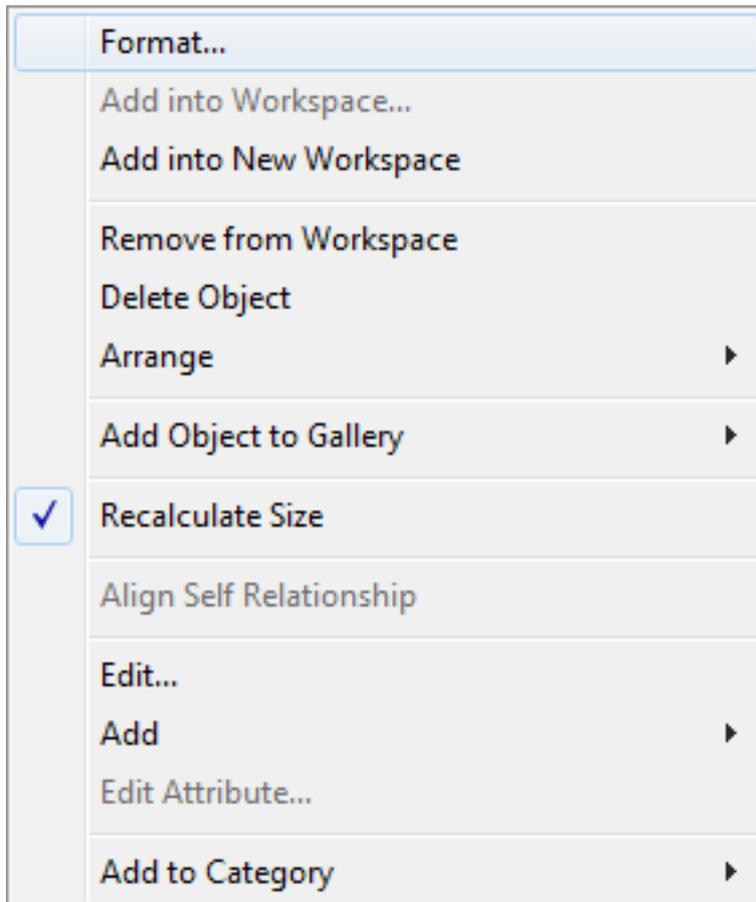
or

- Edit the entity in **Model Explorer** | **Entities** folder | double-click the selected entity (or right-click | **Edit**).

Option	Description
Object Navigator Dropdown Menu	All entities of your logical model are listed here. It allows you to edit entities quickly and comfortably from one place.  <b>Tip:</b> After you finish editing an entity, click <b>Apply</b> to confirm changes and select another entity from the Object Navigator box.
General Tab	Description
Caption	Logical name of entity
Name	Physical name of entity
Category	Category selection box. To see/add/edit categories of your model,

Option	Description
	click the small button on the right.
Size	Definition of supposed table size. Toad Data Modeler also converts defined values to physical model.
Nature	Definition of table nature (Logical information). Select existing value from the box or write a new value.
Logical Only	Select this checkbox not to convert the entity to physical model during LER to PER conversion.
<b>Attributes Tab</b>	On this tab, you can add, edit and delete attributes of the entity.
<b>Unique Identifiers Tab</b>	<p>On this tab, you can manage unique identifiers. A unique identifier for each entity is created by default.</p> <p><b>Note:</b> Working with attributes and unique identifiers in logical model is similar to working with attributes and primary keys in physical model. More details about the operations can be found in appropriate sections of the "Physical Data Model" chapter.</p>
<b>Description Tab</b>	You can enter the object description and technical description here.
<b>To Do Tab</b>	<p>You can enter To Do tasks related to the object here.</p> <p> Note: To display all To Do tasks, select <b>To Do</b> from <b>Model menu</b>.</p>

## Entity Right-Click Options in Logical Model



Option	Description
Format...	Opens the <b>Object Format</b> dialog for selected entity.
Add into Workspace...	Opens the <b>Workspaces</b> dialog where you can select a WS to add the entity to.
Add into New Workspace	Creates a new Workspace in the Application Window and adds the entity to it.
Remove from Workspace	Removes the selected shortcut from particular Workspace.
Delete Object	Deletes the selected entity from model.
Arrange	Arranges the entity in another layer. <a href="#">Arrange Objects in Layers</a>
Add Object to Gallery	Adds object to new or selected Gallery.

Option	Description
Recalculate Size	Adjusts the entity size to the length of its attributes.
Align Self Relationship	Aligns self relationship.
Edit	Opens the <b>Entity Properties</b> form.
Add	Adds new object (Attribute, Unique Identifier Attribute or empty Unique Identifier)
Edit Attribute...	Opens the Attribute Properties form. This option is active only if attribute is selected in the diagram.
Add to Category	Adds object to selected Category.

## Create Attributes

### To create an attribute in LER model

- Double-click an entity on the Workspace to open the **Entity Properties** form | **Attributes** tab | **Add**.  
or
- **Model Explorer** | **Entities** folder | Unfold the selected entity. | Right-click the **Attributes** item. | **Add Attribute**.

### Entity Properties attribute columns:

Option/Column	Description
Ident.	Graphical representation of unique identifier of particular attributes
Caption	Logical attribute name
Name	Physical attribute name
Data Type	Data Type of an attribute
Mandatory	The Mandatory selection box
Status	<a href="#">Status of Items in Grids</a>

## Buttons:



- opens the **Application Variables** form

**Add** - adds an attribute

**Edit** - opens the **Attribute Properties** dialog

**Delete** - deletes selected attribute



- changes position of selected attribute - up and down

## Edit Attributes

- In the **Entity Properties** dialog | **Attributes** tab, double-click the selected attribute.

or

- Find the attribute in **Model Explorer** | *EntityName* folder | **Attributes** | Double-click the selected attribute.

### Option

### Description



Above the Object Navigator Dropdown Menu, you can see name of entity that the attribute belongs to. Click the button on top right-hand corner to open the parent form (Entity Properties form).

Object Navigator  
Dropdown Menu

All attributes of selected entity are listed here. The box allows you to edit attributes quickly and comfortably from one place.

Option	Description
<b>General Tab</b>	<b>Description</b>
Caption	Logical attribute name.
Name	Physical attribute name.
Data Type	Data Type selection box.
Domain	<p>Domain selection box.</p> <p><b>i</b> Note: It's not possible to assign a data type and a domain to an attribute at the same time. If you select a domain, appropriate data type will be set automatically from the domain.</p>
Default Rule	Default selection box. Defaults are converted to physical model also.
Logical Only	Select this checkbox to not convert the attribute to physical model during LER to PER conversion.
Mandatory	Mandatory items will be converted to Not Null items in physical model.
<b>Rules Tab</b>	A tab where you assign rules to the attribute. Rules are used for the generation of Rules in physical model and later for Check Constraints in the generated SQL code (generated from physical model). In Rules, check constraints for multiple columns can be defined, for example.
<b>Valid Values Tab</b>	<p>Valid values can be defined for the following data types:</p> <ul style="list-style-type: none"> <li>• Bigint</li> <li>• Float</li> <li>• Integer</li> <li>• Char</li> <li>• VarChar</li> </ul> <p>Valid values are used for generation of simple check constraints.</p> <p>Valid values can be defined as Enumeration (for VarChar data type, for Example) or as a Range (for Integer data type.)</p> <p>As soon as you add a new Enumeration, the in-place editor in the Enumeration box will activate automatically.</p> <p><b>i</b> Note: In Toad Data Modeler, valid values in LER model = check constraints in PER model (feel free to check out the conversion).</p>

Option	Description
	If the <b>Valid Values</b> tab is not available (depends on the selected data type), you can use rules (see the <b>Rules</b> tab).
<b>Description Tab</b>	You can enter the object description here.
<b>To Do Tab</b>	You can enter To Do tasks related to the object here. <div style="border-left: 1px solid black; padding-left: 5px; margin-left: 15px;"> <p><b>i</b> Note: To display all To Do tasks, select <b>To Do</b> from <b>Model menu</b>.</p> </div>

## Create Unique Identifier

### To create a unique identifier

In the **Entity Properties** dialog, tab **Attributes**, double-click the space in the **Ident.** column next to the selected attribute.

### To create a new key

1. In the **Entity Properties** form | **Unique Identifiers** tab | click **Add**.
2. Edit the new unique identifier and on tab **Attributes**, assign an attribute to it.



Important: Primary keys are graphically marked by red key.

Keys do NOT migrate in Logical models, only in Physical models. In Logical model, only logical information is available and the relationships have only a logical meaning (therefore keys do not migrate). In Physical model the information about foreign keys is necessary, that's why after converting your Logical model to Physical model, FKs will migrate properly. This process is done automatically.

## Edit Unique Identifiers

### To add or edit the unique identifier

- Open the **Entity Properties** dialog | **Unique Identifiers** tab | double-click the unique identifier.

or

- Find the unique identifier in **Model Explorer** | *EntityName* folder | **Unique Identifiers** | Double-click or right-click it and select **Edit**.

### To assign unique identifier to an attribute

1. Click the **Attributes** tab in the **Unique Identifier Properties** dialog and select an attribute (attributes).



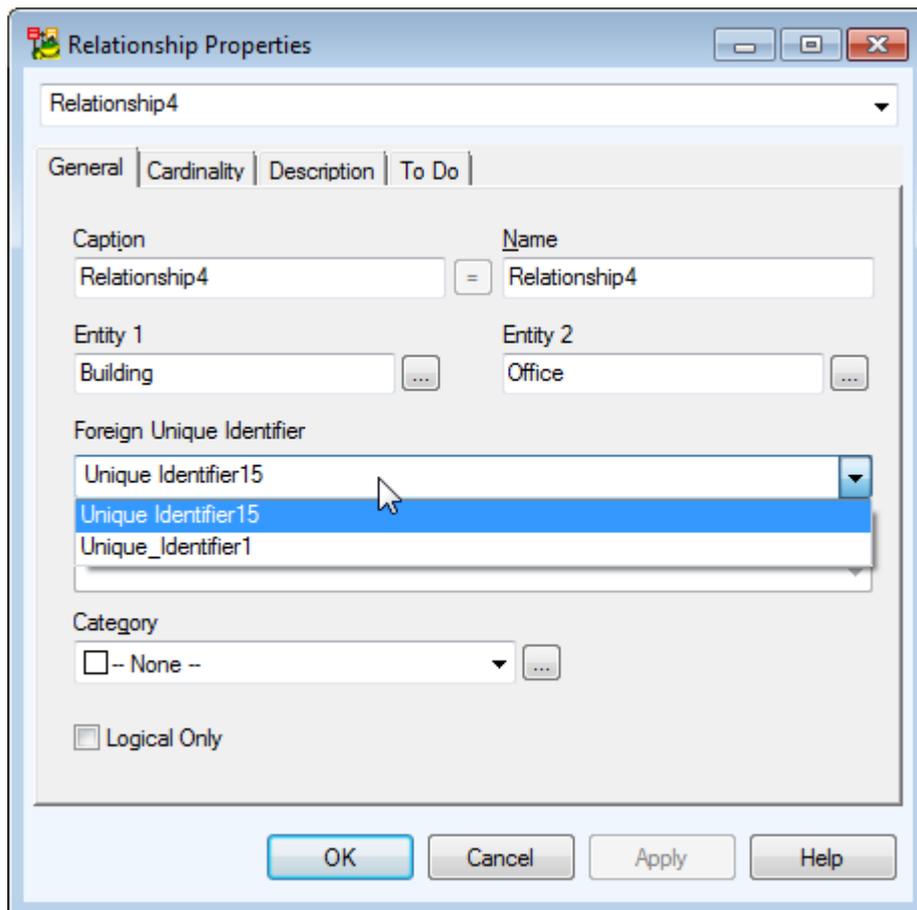
2. Click , to shift the selected attribute(s) to the section **Selected**.
3. Confirm by clicking **OK**.

## Select Linking Method

### To select a unique identifier before the LER - PER model conversion

1. Edit the selected relationship and click the **General** tab.
2. From the **Foreign Unique Identifier** box, select the identifier that will be used during LER to PER conversion.

**i** Note: For inversed relationships, you can select the identifier from the **Opposite Foreign Unique Identifier** dropdown menu.



## Edit Relationships

Logical model supports the following relationship types:

- Identifying relationship
- Non-identifying relationship

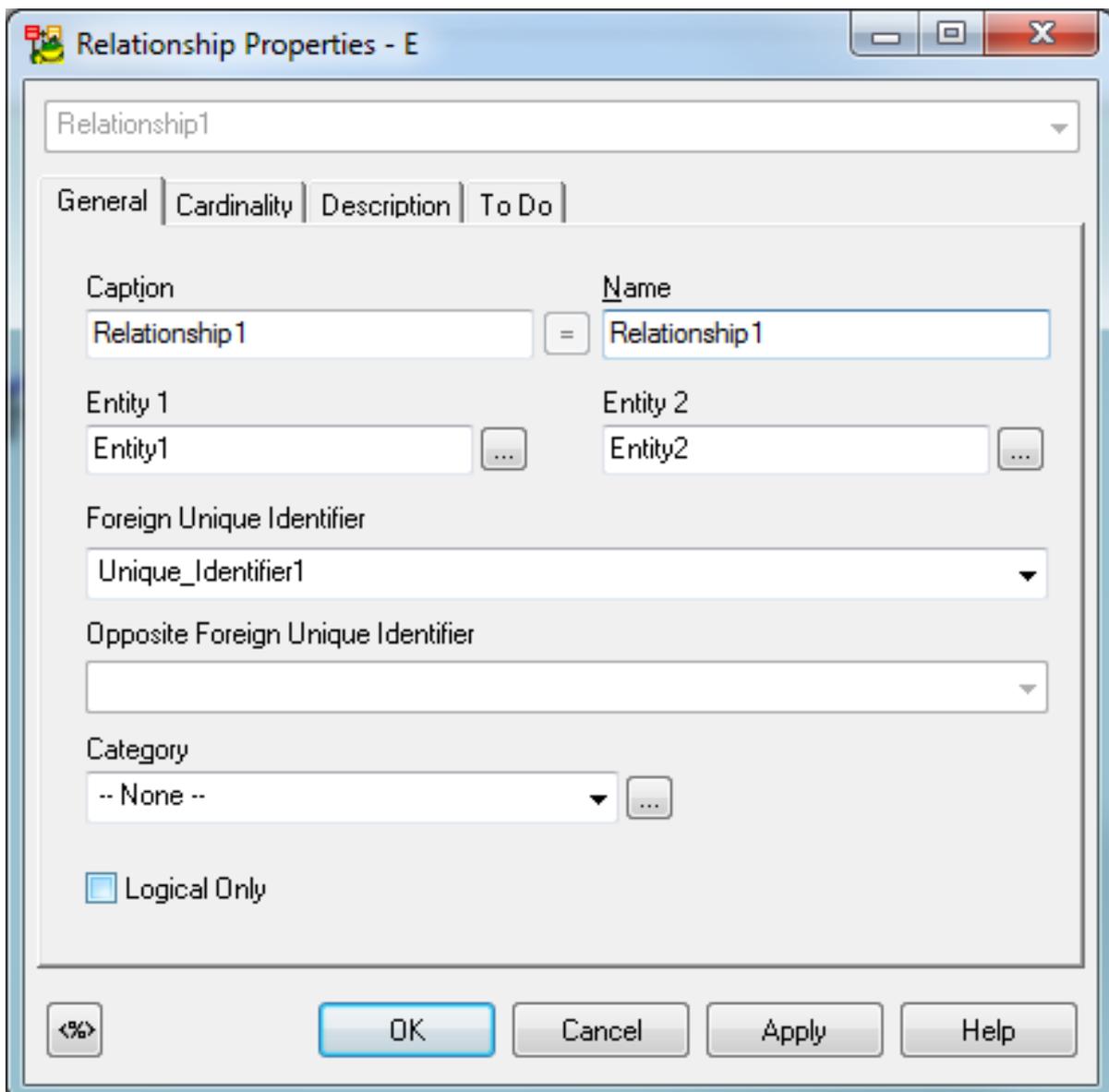
- Self-relationship (identifying as well as non-identifying)
- M:N relationship [M:N Relationships](#)
  - Note: M:N relationships are created by adding an identifying/non-identifying relationship and then changing its cardinality to **Many - Many**.

### To edit a relationship

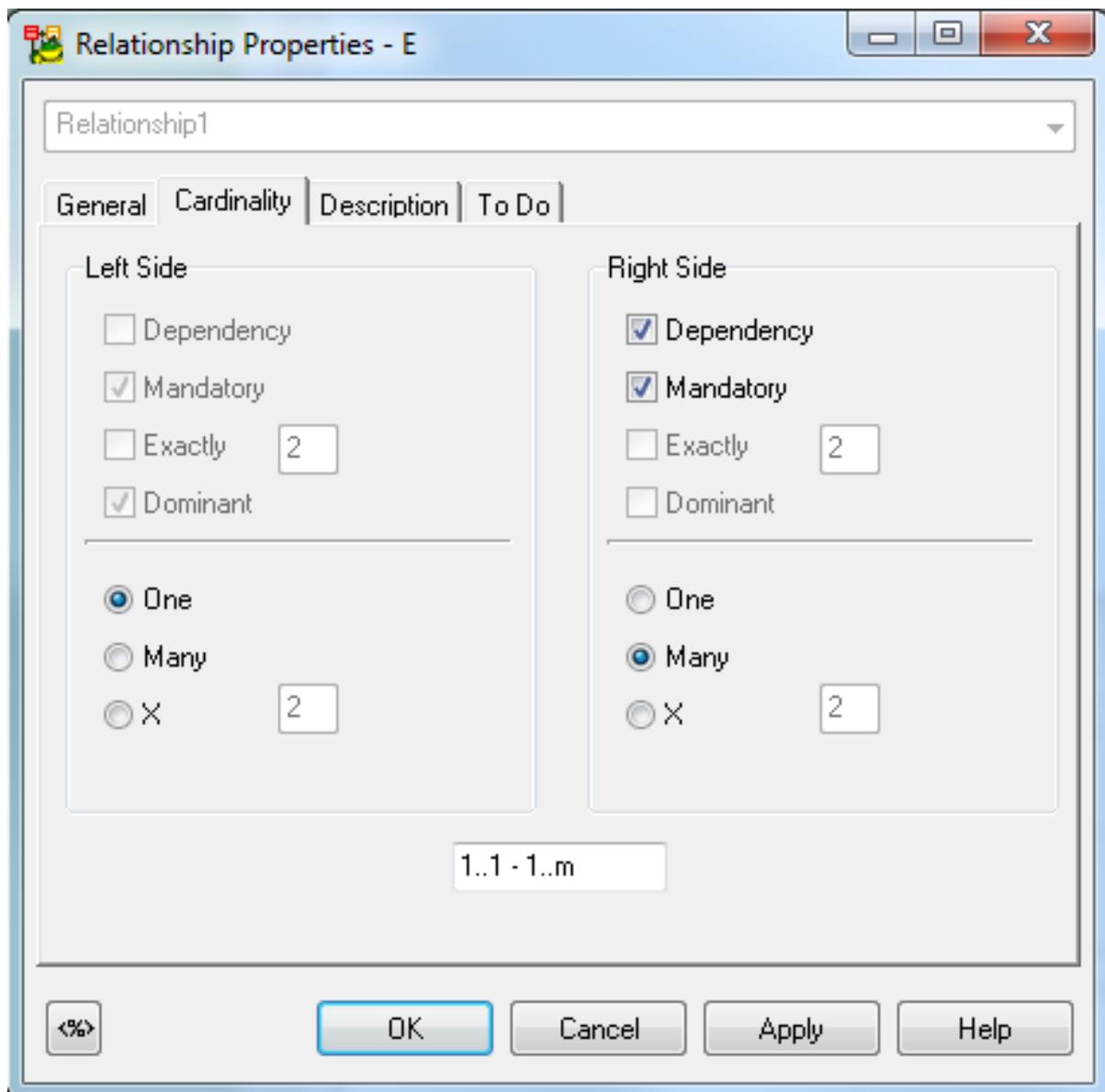
- Double-click the relationship on the Workspace.

or

- Edit the relationship in **Model Explorer** | **Relationships** folder | double-click the selected relationship (or right-click | **Edit**).



Option	Description
Object Navigator Dropdown Menu	<p>All relationships of your logical model are listed here. Edit relationships quickly and comfortably from one place.</p> <p><b>Tip:</b> After you finish editing a relationship, click <b>Apply</b> to confirm changes and select another relationship from the Object Navigator Dropdown Menu.</p>
General Tab	Description
Caption	Logical name of relationship
Name	Physical name of relationship
Foreign Unique Identifier	<p>Select the linking method for the relationship. According to your selection, the LER model will be converted to PER model.</p> <p><a href="#">Migration of Keys</a></p>
Opposite Foreign Unique Identifier	Select the linking method for the inversed relationship.
Logical only	Select this checkbox to not convert the relationship during LER to PER conversion.



**Cardinality Tab**

Define the cardinality.

Left Side:

**Dependency** - definition of dependency

**Mandatory** - definition of mandatory item on the left side

**Exactly** - definition of the lower cardinality range limit

**Dominant** - definition of a would-be parent for conversion to PER on the left side (if active and not defined otherwise)

**One** - definition of one to many or many to many relationship

**Many** - definition of one to many or many to many relationship

**X** - definition of the higher cardinality range limit

Right Side:

**Dependency** - definition of dependency

**Mandatory** - definition of mandatory item on the right side

**Exactly** - definition of the lower cardinality range limit

**Dominant** - definition of a would-be parent for conversion to PER on the right side (if active and not defined otherwise)

**One** - definition of one to many or many to many relationship

**Many** - definition of one to many or many to many relationship

**X** - definition of the higher cardinality range limit

---

**Description Tab** You can enter the object description here.

---

**To Do Tab** You can enter To Do tasks related to the object here.

**i** Note: To display all To Do tasks, select **To Do** from **Model menu**.

**i** Important: Keys do NOT migrate in Logical models, only in Physical models. In Logical model, only logical information is available and the relationships have only a logical meaning (therefore keys do not migrate). In Physical model the information about foreign keys is necessary, that's why after converting your Logical model to Physical model, FKs will migrate properly. This process is done automatically.

## Create Inheritances

1. Click  on the toolbar (also CTRL+I)
2. Move your mouse cursor over the work area and click the parent entity and then the target entity (child).

**i** TIP:

- While creating an inheritance, you can add handle points at the same time. Click the parent entity and then click the work area as many times as many handle points you want to create. Finally click the child entity.
- Create several Children of the existing inheritance:

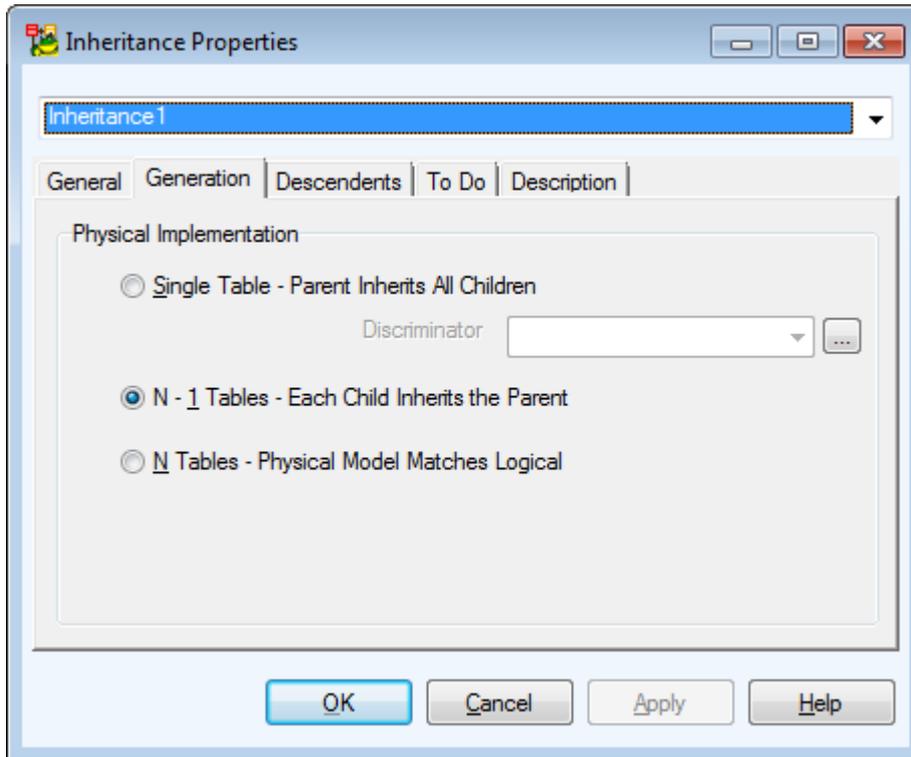
1. Click the Inheritance icon  on the toolbar.
2. Click the inheritance in the model.
3. Click another entity in the model.

## Edit Inheritances

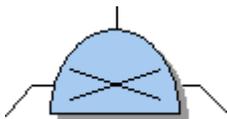
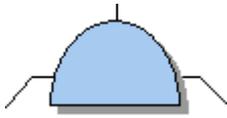
- Double-click the inheritance on the Workspace.

or

- Edit the inheritance in **Model Explorer | Inheritances** folder | double-click the selected inheritance (or right-click | **Edit**).



Option	Description
Object Navigator Dropdown Menu	All inheritances in your model are listed here. Use it to switch between multiple objects and edit them easily.
<b>General Tab</b>	<b>Description</b>
Caption	Logical name of inheritance
Name	Physical name of inheritance
Parent	Name of the parent entity
Logical Only	Select this checkbox to not convert the inheritance during LER to PER conversion.
Exclusive	<p>Definition of exclusive inheritance.</p> <p>For exclusive inheritances, Toad Data Modeler generates triggers that will perform a check whether a correct record in siblings exists or not, and decide whether a record can be added to table or not etc.</p> <p>Exclusive inheritances are displayed with cross in the middle of the graphics:</p>

Option	Description
	 <p>Standard inheritances don't have the cross inside the graphics:</p> 
Complete	Logical information only. The information says that all records must be complete.
<b>Generation Tab</b>	On this tab, select how do you want to resolve the inheritance during conversion from LER to PER. <a href="#">Inheritance</a>
<b>Descendents Tab</b>	<b>Description</b>
Name	Name of descendent
Discriminator Valid Value	Valid values of Discriminator
Edit Discriminator	Opens the <b>Valid Values</b> dialog for the selected Discriminator.
<b>To Do Tab</b>	<p>You can enter To Do tasks related to the object here.</p> <p><b>i</b>   Note: To display all To Do tasks, select <b>To Do</b> from <b>Model menu</b>.</p>
<b>Description Tab</b>	You can enter the object description here.

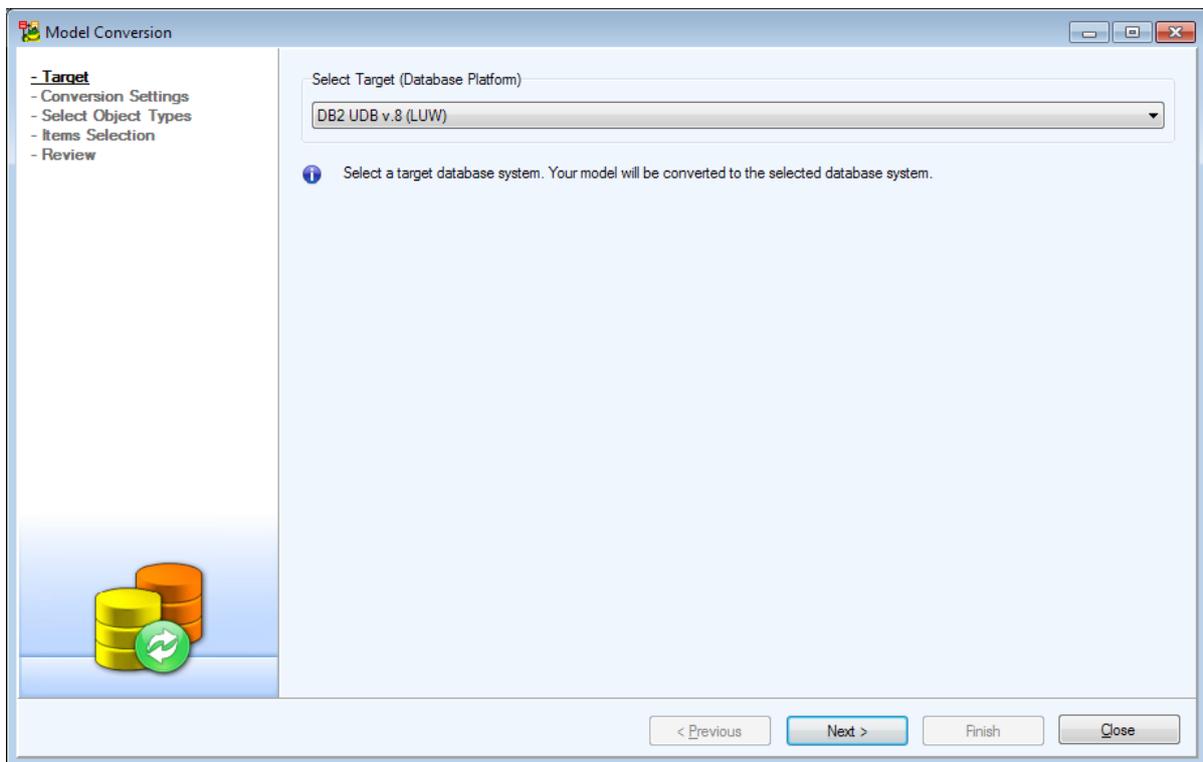
## Convert Logical Model to Physical Model

Toad Data Modeler allows you to convert your logical model to a physical model of any supported database system.

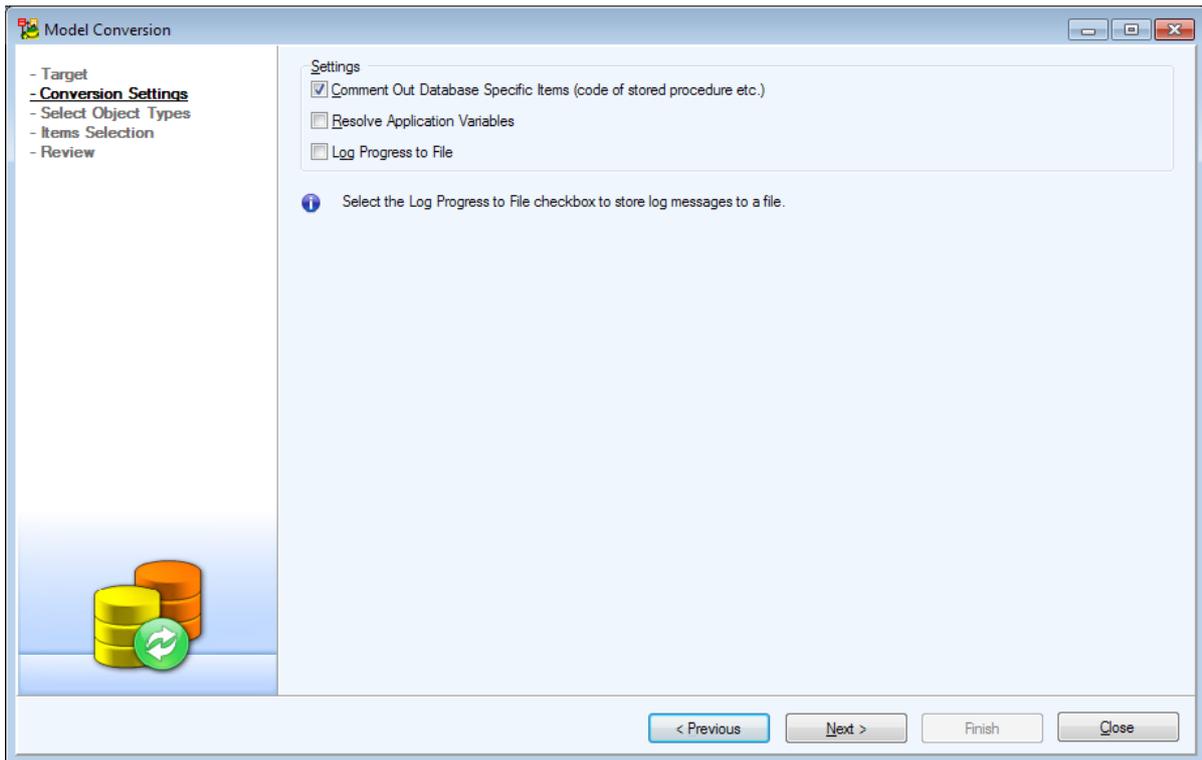
**i** Note:

Before you convert your Logical model to Physical model, you should be aware of the following:

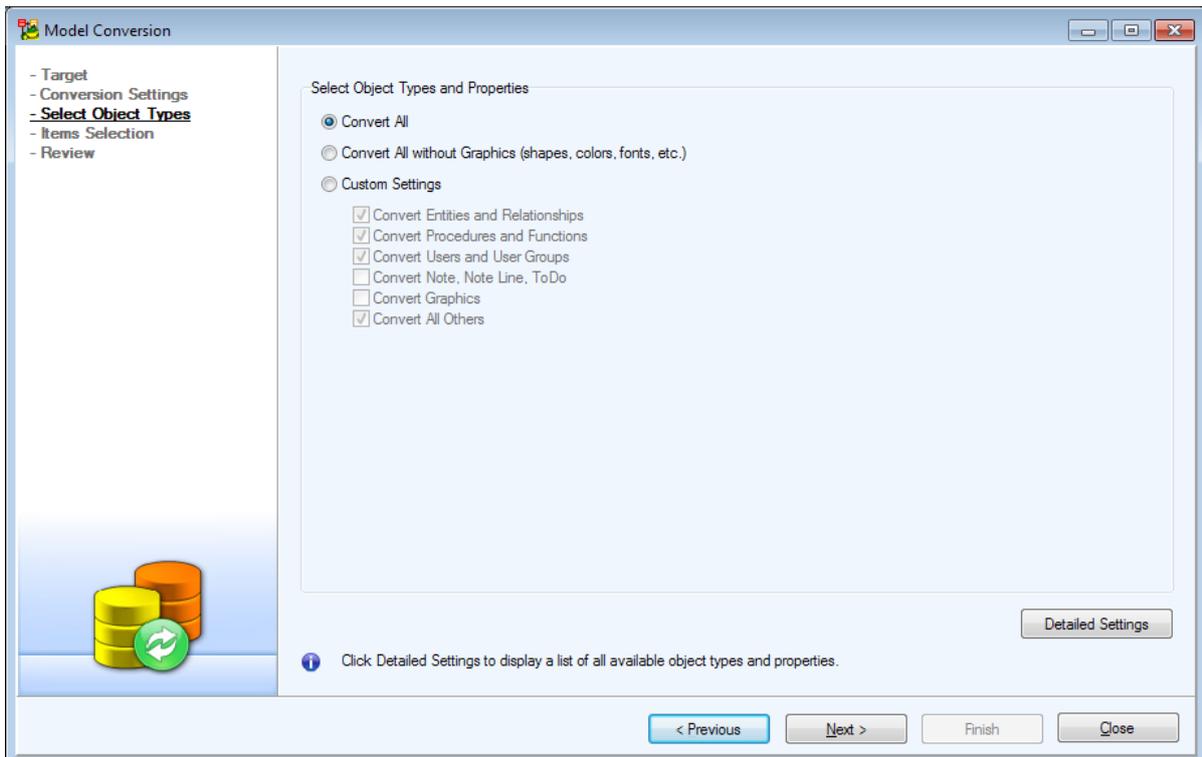
- Physical model supports only non-identifying self relationship.
- Inheritance is not supported in PER model. Toad Data Modeler solves this by converting Inheritance object into either **Single Table**, **N-1 Tables** or **N Tables**. [See Inheritance](#).
- Keys in LER models do not migrate.
- You can select a linking method in LER model.
- M:N relationships are supported in both models.
- Before you start the conversion, you can set up the conversion rules in the **Data Type Conversion Settings** dialog. This option is available only if Expert Mode is enabled.
- Cycled relationships will be ignored during LER to PER conversion and will not be converted. A message informing you about this will be displayed in Message Explorer Log.



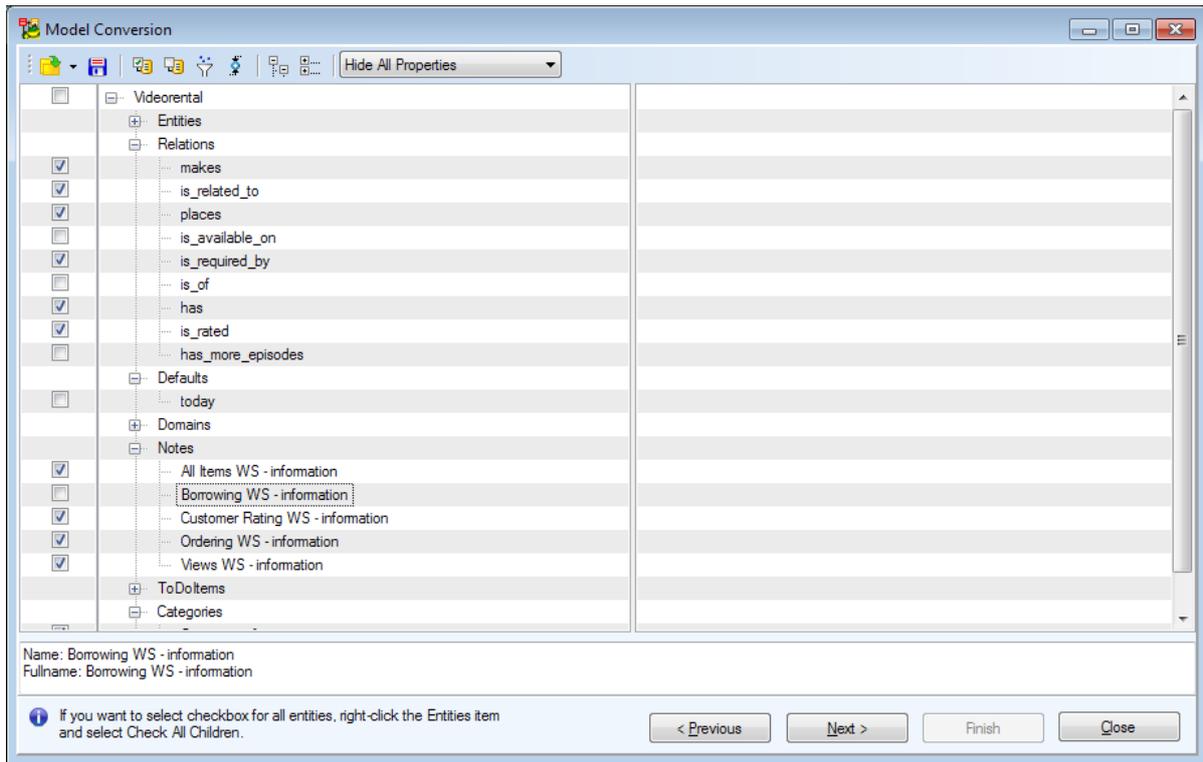
- Select your desired database platform.



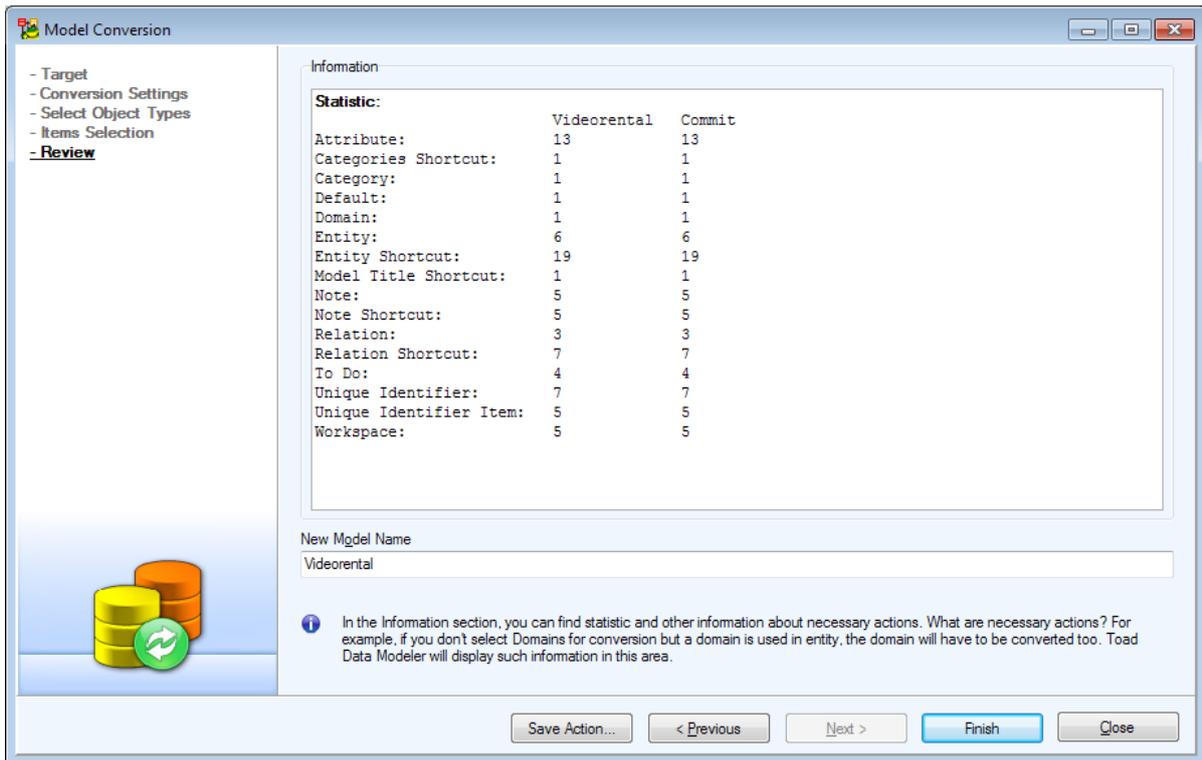
- Change any of the conversion settings, if necessary.



- Choose what object types will be converted. You can access full list of Objects and Properties by clicking on **Detailed Settings**.



- Check items you want to convert to another model. For easier item management use buttons located on the top.



- Review the statistic and when you're done, click the **Finish** button.
  - After a short while your Logical model will be converted to Physical model of your desired database platform.
- i** TIP: If you plan to do this action again in the future, you might want to click the **Save Action** button to save this **Action Definition**. See [Model Actions](#) for more information.

### Selection Tree Overview

 Collapse All.

 Import selection from a saved file.

 Save selection to a file.

 Checks all items.

 Unchecks all items.

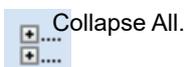
 Opens the **Wildcard Dialog** where you can define settings for bulk selection/deselection of the **Action** box of the items listed on page **Select Items**.



### Refresh Necessitated Items

Explanation: Some objects are related together (e.g. entity and domain, entity and relationship). Let's say you uncheck a Domain in **Select Object Types** dialog. However you keep an Attribute of the Domain type checked for conversion. In the next screen the Domain will be selected for conversion (and highlighted in gray), even if you don't want it to. This is because of its relationship with the Attribute, which cannot exist without the Domain.

Now, if you uncheck the Attribute, the Domain will still be checked for conversion. This is where you use this button. It runs through all checked objects and removes the Domain highlighted in gray since the Attribute is no longer checked. That means the Domain is no longer necessary, since it has no relationships with currently checked objects and you unchecked it in **Select Object Types** dialog.



**Right-click an item to see the following options:**

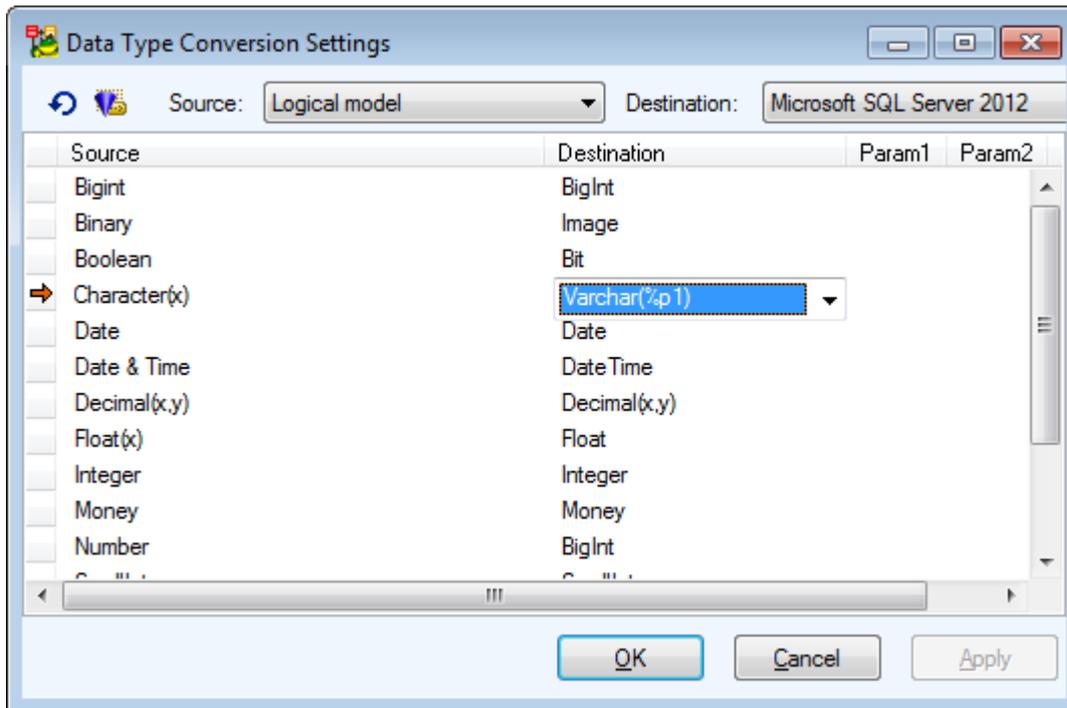
Option	Description
Expand All Children	Expands all sub-items of the selected item.
Collapse All Children	Collapses all sub-items of the selected item.

## Data Type Conversion

### To define rules for data type conversion

Select **Expert Mode** | **Expert Mode Settings** | **Data Type Conversion Settings**. [Data Type Conversion Settings](#)

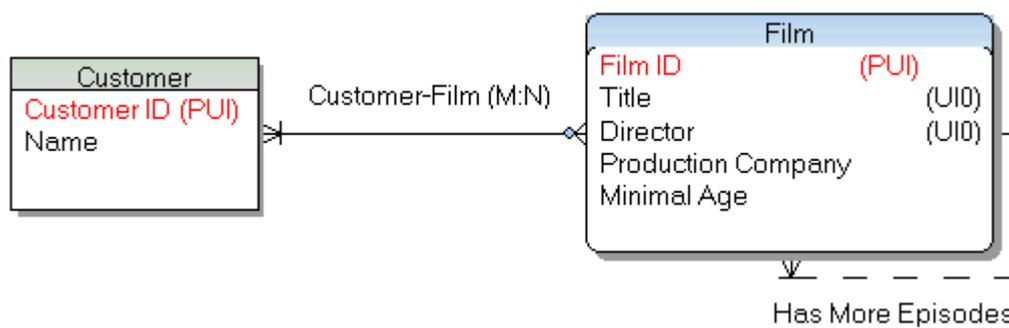
**i** Note: To enable Expert Mode, select **Settings** | **Options** | **General** | **Expert Mode** checkbox.



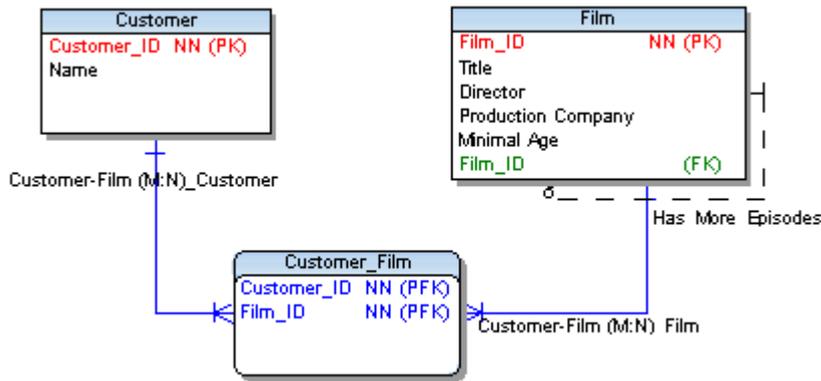
## M:N Relationships

LER and PER models both support M:N relationships. See below how they are converted.

### LER Model



### PER Model (After Conversion)



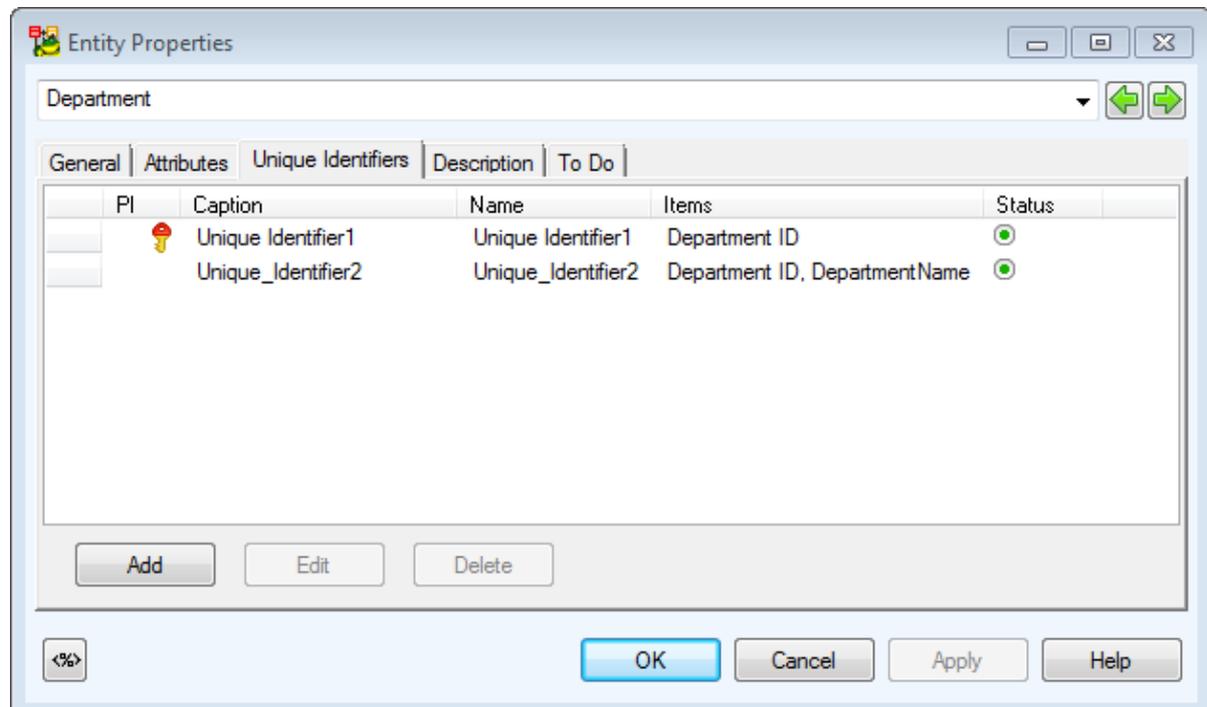
## Migration of Keys

In Logical model, keys **do not migrate** from parent entities to child entities. In Logical model, only logical information is available. The relationships have only a logical meaning (-> FKs do not migrate). However, in Physical model the information on foreign keys is necessary. Therefore when you convert your logical model to a physical model, foreign keys will be displayed properly in the physical model.

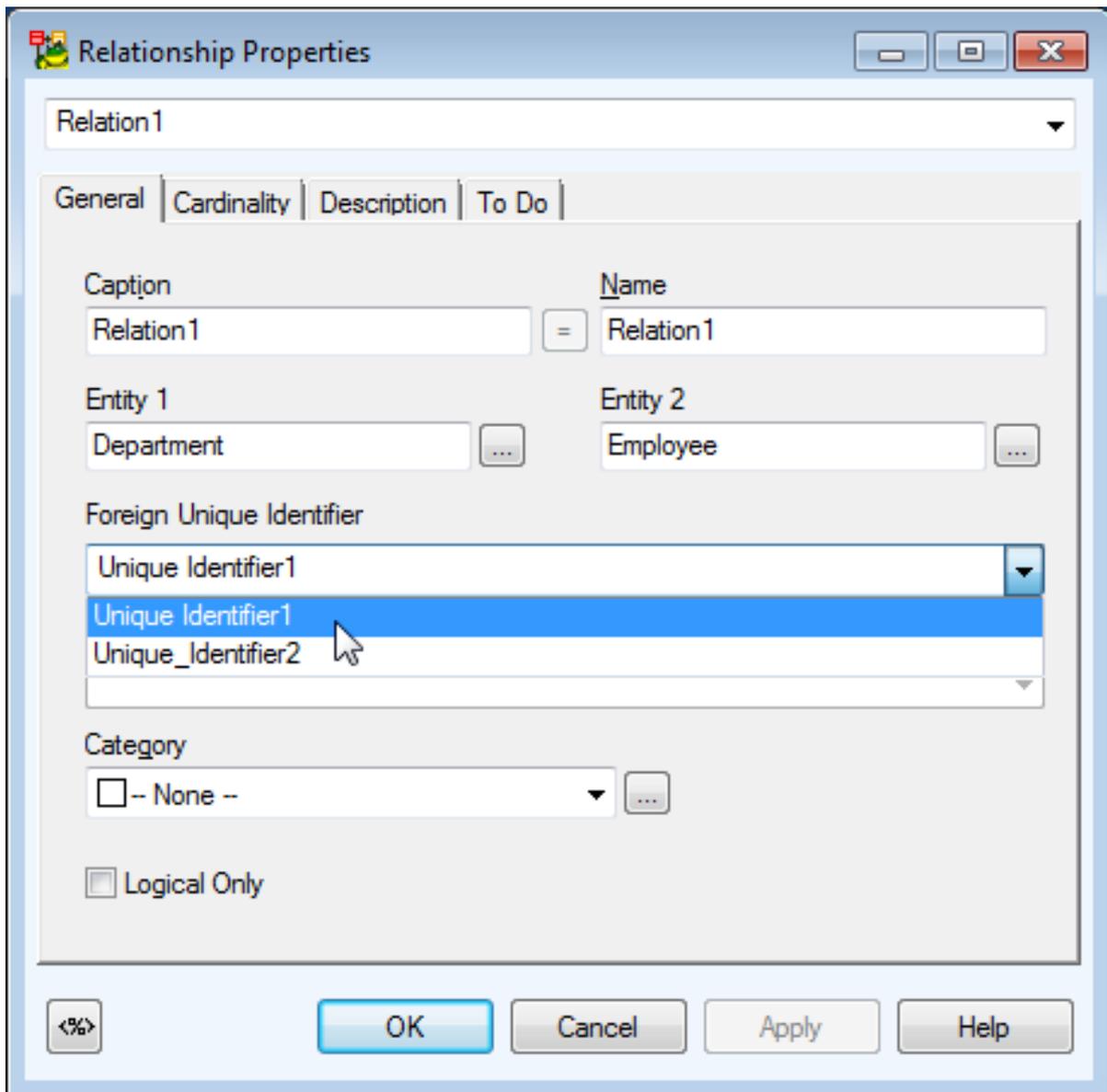
## Linking Method

Before you start the LER to PER conversion, you can select a linking method in relationships in your LER model. See the following example and the differences after the conversion to PER model:

**Example:** Department entity has two unique identifiers (non-identifying relationship).

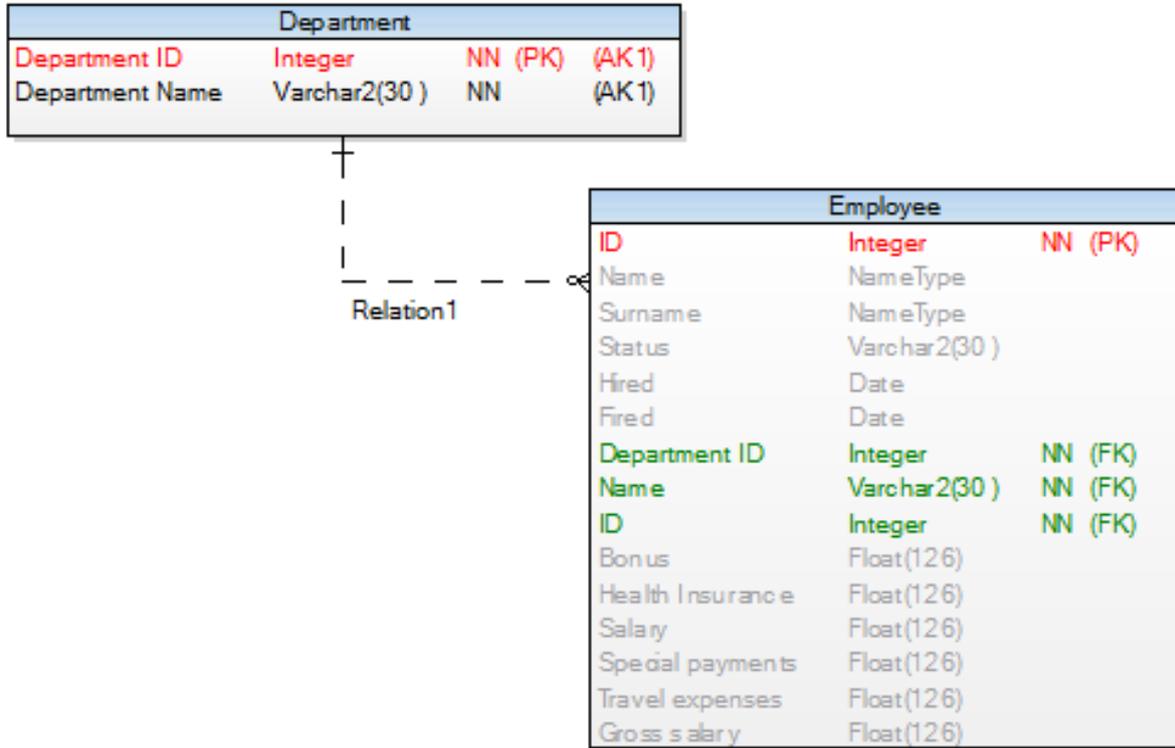


a) UI1 has been selected.



#### Result after Conversion to Physical Model

- Department ID FK is now in the Employee table (child table).
- Alternate key has been created in the PER model automatically.

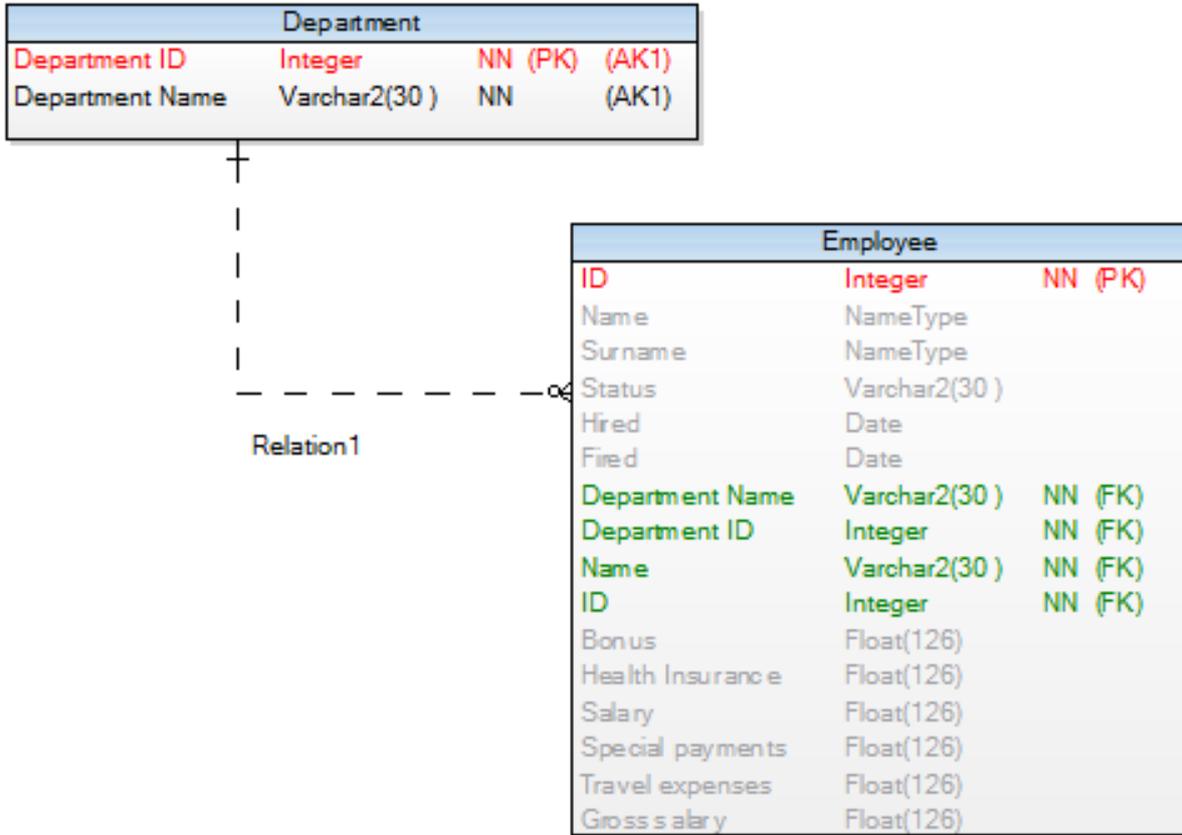


**b) UI2 has been selected in LER model.**

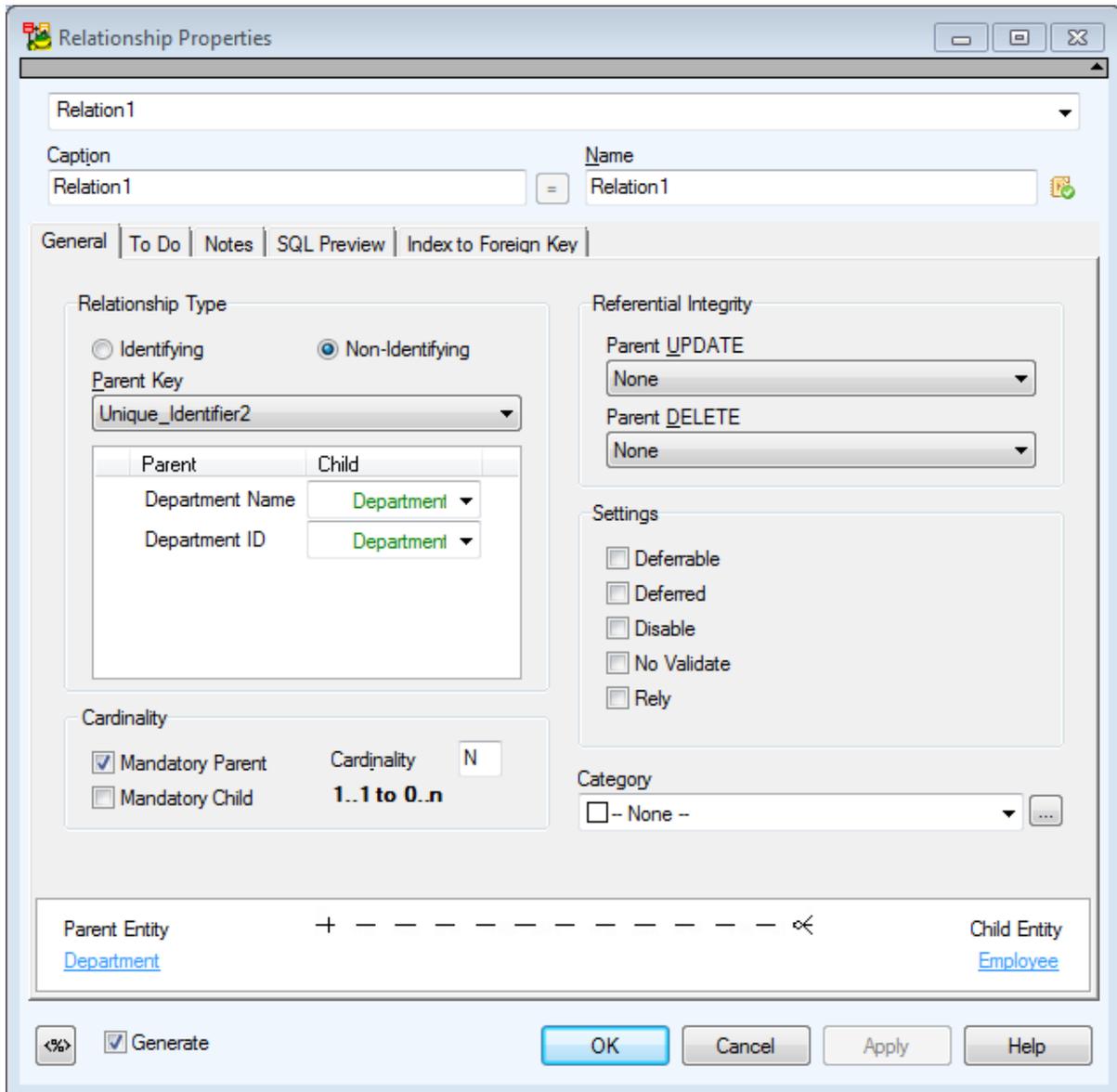
**i** Note: In previous Toad Data Modeler versions, during conversion to physical model, primary key was always selected for the relationship as a parent key. Now you can select also alternate key (e.g. UI2) and opposite foreign unique identifiers.

**Result after Conversion to Physical Model**

- Department ID and Department Name FKs are now in the Employee table (child table).
- Alternate key has been created in the PER model automatically.



See the **Relationship Properties** dialog | **Foreign Keys** tab in PER model after conversion:



## LER to PER Conversion - Self-Relationship

If there is a self-relationship in LER model, the entity has two columns, both of the same name (primary key), in converted PER model. Other modifications are necessary.

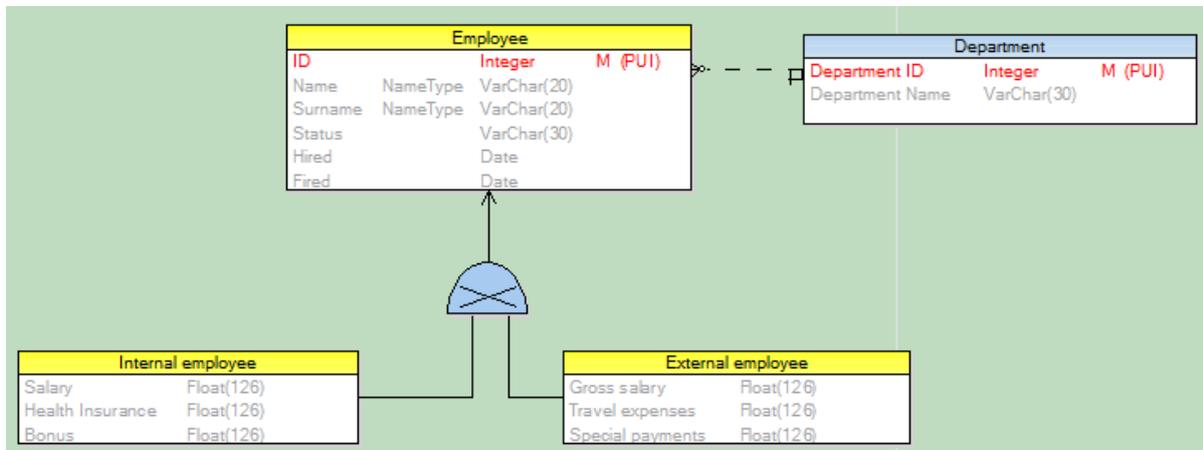
Possible solution: You can define a name for the propagated attributes in LER model before the conversion. Open the **Attribute Properties** dialog | **General** tab | enter the name to the **Propagated Name** box.

If this box is empty, Toad Data Modeler will behave standardly (two columns of the same name in PER model).

**i** Note: It is also possible to set a self relationship attribute name and caption for PER model in **Settings** | **Options** | **Physical Model** | **Self Relation Attribute Name, Caption**.

# Inheritance

Inheritance is a special abstract object which can be used in early phases of database development to visualize the inheritance process.

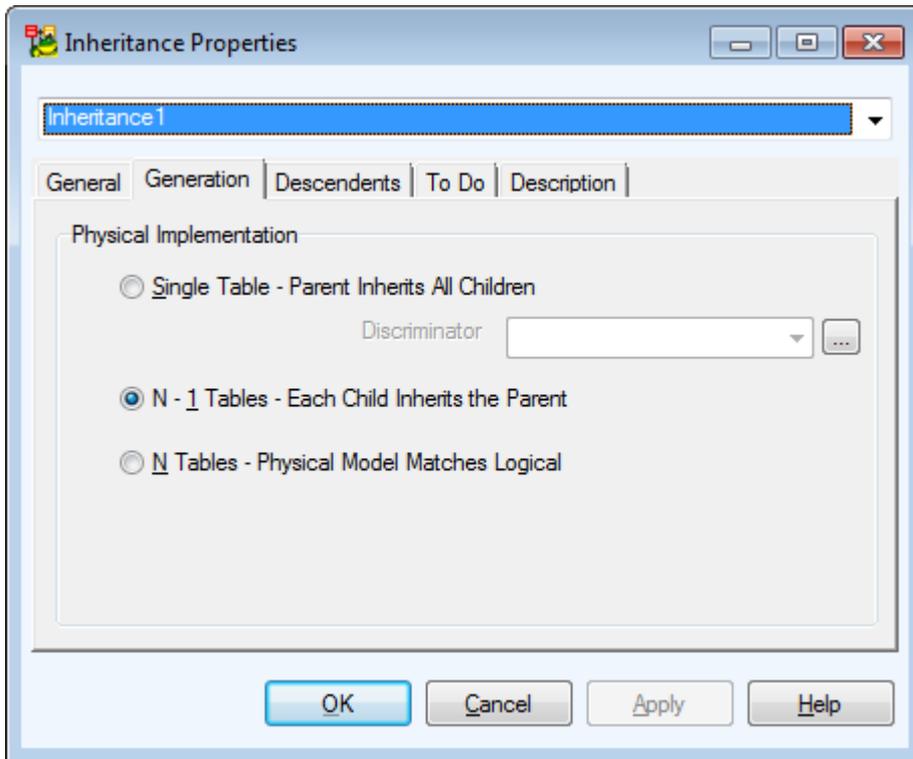


However, Inheritance is only supported in Logical Model. When converting Logical Model to Physical Model, Toad Data Modeler resolves Inheritance one of the three ways:

- **Single Table**
- **N - 1 Tables**
- **N Tables**

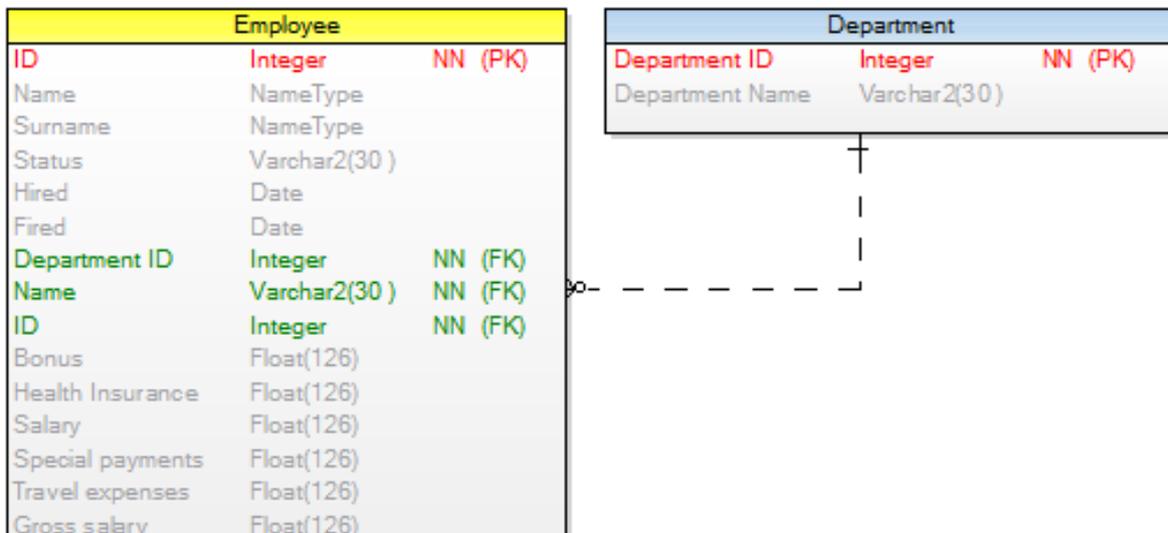
Before you convert your Model, you have the option to pick one of the three ways to resolve all Inheritance objects:

1. Double-click the Inheritance and select tab **Generation**.
2. In the **Physical Implementation** area, select any of available options:



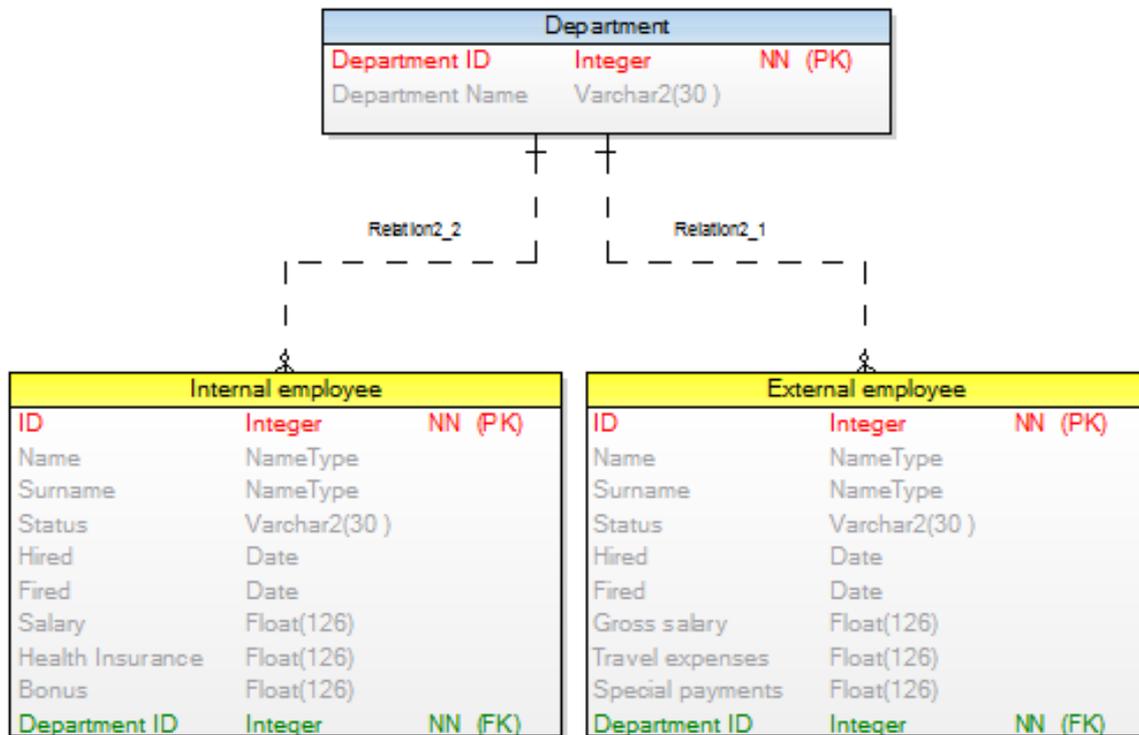
### a) Single Table - Parent Inherits All Children

The conversion output is the following:



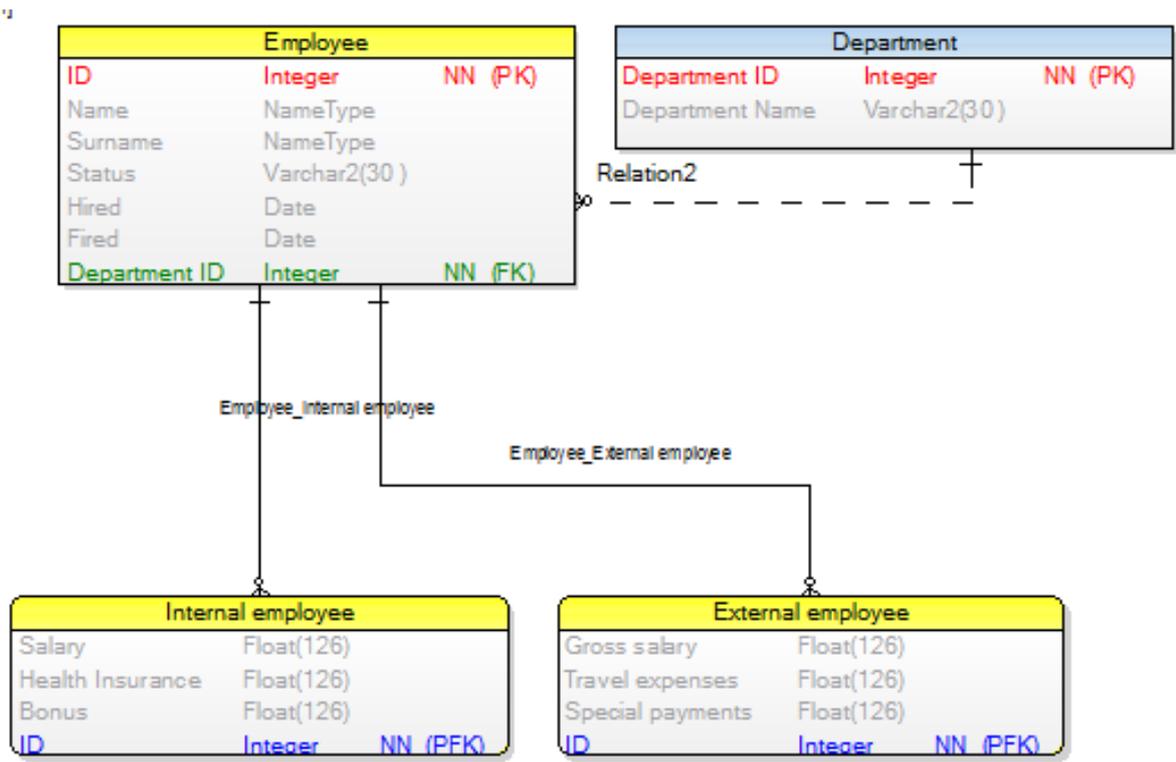
### b) N - 1 - Each Child Inherits the Parent

The conversion output is the following:



### c) N Tables - Physical Model Matches Logical

The conversion output is the following:



**i** Note:

- Single Table Generation (Parent Inherits All Children), Discriminator for Child Entities is Set** - After the conversion, a check constraint will be created in parent entity in PER model. This check constraint determines conditions for setting Not Null value of attributes taking into account options set in Discriminator.
- Inheritance Exclusive, N - 1 Generation (Each Child Inherits the Parent)** - During the conversion, all attributes of parent entity will be passed to child entities and triggers will be created in the child entities. These triggers determine creation of new items in child entities. - Rule: it is not possible that two child entities with the same key of parent entity would exist.
- N Tables Generation (Physical Model Matches Logical)** - Instead of inheritance, relationship (identifying by default) will be created between parent and child entities in PER model.
- Inheritance conversion - Relationship names are made unique in physical model.**
  - N - 1 Tables - Each Child Inherits the Parent: *Relationship name\_number* (e.g. Relationship1\_1, Relationship1\_2)
  - N Tables - Physical Model Matches Logical: *Parent Entity Name\_Child Entity Name*

## Defaults

### To add a default

Select **Model | Model Items | Defaults** and click **Add** in the **Defaults** dialog.

### To edit a default

Select **Model | Model Items | Defaults** and double-click the selected default or click **Edit**.

**i** | Note: You can also edit/rename/delete defaults in **Model Explorer** | **Defaults** folder | Right-click the selected default.

### **To select a default for attributes**

1. Open the **Attribute Properties** dialog of the selected attribute | **General** tab.
2. From the **Default Rule** box, select a default or write your own default to the **Default** box.

**i** | Note: You can define defaults for domains too.

## Rules

### **To add a rule**

Select **Model** | **Model Items** | **Rules** and click **Add** in the **Rules** dialog.

### **To edit a rule**

Select **Model** | **Model Items** | **Rules** and double-click the selected rule or click **Edit**.

**i** | Note: You can also edit/rename/delete rules in **Model Explorer** | **Rules** folder | Right-click the selected rule.

### **To select a rule for attributes**

1. Open the **Attribute Properties** dialog of the selected attribute | **Rules** tab.
2. Select a rule and click the Add arrow button.

**i** | Note: You can define rules for domains too.

## Add Objects

Toad Data Modeler allows you to add objects from multiple places. See the following options:

### **To add an object on Workspace**

1. Click the object icon on the toolbar, e.g.  for entity
2. Click anywhere on the workspace to add the object.

or

1. In **Objects Menu** | **Add New** select the object you want to add to Workspace
2. Click anywhere on the workspace to add the object

**i** | TIP: You can also use hot keys to add some objects, e.g. CTRL+E for entity, CTRL+I for inheritance. See **Hot Keys** for more information.

### **To add multiple objects to Workspace**

1. Press SHIFT and click the object icon on toolbar.
2. Click on workspace as many times as many objects you need to add.
3. Right-click the work area (or click the object icon again) to turn this function off.

### **To add objects to Model Explorer**

1. Find the object group in **Model Explorer** (e.g. **Entities**) | **right-click** and select **Add** (e.g. **Add Entity**).

### **To add objects from Model menu**

1. Select **Model** | **Model Items** | and the object type (e.g. **Entities**).
2. In following dialog click the **Add** button.

## **Edit Objects**

Toad Data Modeler allows you to edit objects from multiple places. See the following options.

### **To edit objects on the Workspace**

**Double-click** or **Enter** the selected object.

### **To edit objects from Model Explorer**

Find the object in tree and **double-click** it (or **right-click** | **Edit**).

**i** | Note: Double-clicking in Model Explorer does not highlight the object on the Workspace (WS) but opens the **Object Properties** dialog instead. To highlight a object on WS, right-click and select **Find on Workspace**.

To edit objects from Objects menu

1. Select an object on Workspace or in Model Explorer.
2. Select **Objects Menu** | **Edit**

To edit objects from Model menu

1. Select **Model Menu** | **Model Items** | the object type (e.g. **Entities**).
2. In the dialog (**Entities**), select the object and click **Edit**.

### **Buttons in Object Properties dialogs:**



- opens the **Application Variables** form

**OK** - confirms changes, closes the form/dialog

**Cancel** - cancels changes

**Apply**- confirms changes, doesn't close the form/dialog

**Help** - Help navigation

**Note:** Objects of your model that have the **Generate** checkbox disabled in their **Properties** dialogs are displayed in Model Explorer this way:



## Format Objects

- [Object Format for New Models](#)
- [Object Format for Existing Models](#)
- [Object Format for Particular Object](#)

### ***To set format for new models (models that you will create)***

1. Select **Settings | Options | Model section | Physical/Logical Model**.
2. Define options on tabs **General, Workspace, Shape, Note Line and Entity**.
3. Press CTRL+N to create a new model.

The format will be used for new models/workspaces. Using this option you cannot change format of objects in already existing models.

### ***To change format of objects in existing models***

- Right-click the Workspace and select **Workspace Format**.

The format is applied to all objects on current Workspace, that share the Workspace format and all objects that you will create on the Workspace.

### **Example**

You have two entities on your Workspace. One has blue brush color and the other one has red brush color. Your Workspace has red brush color. When you change your Workspace format brush color from red to white, the entity with red brush will also change.

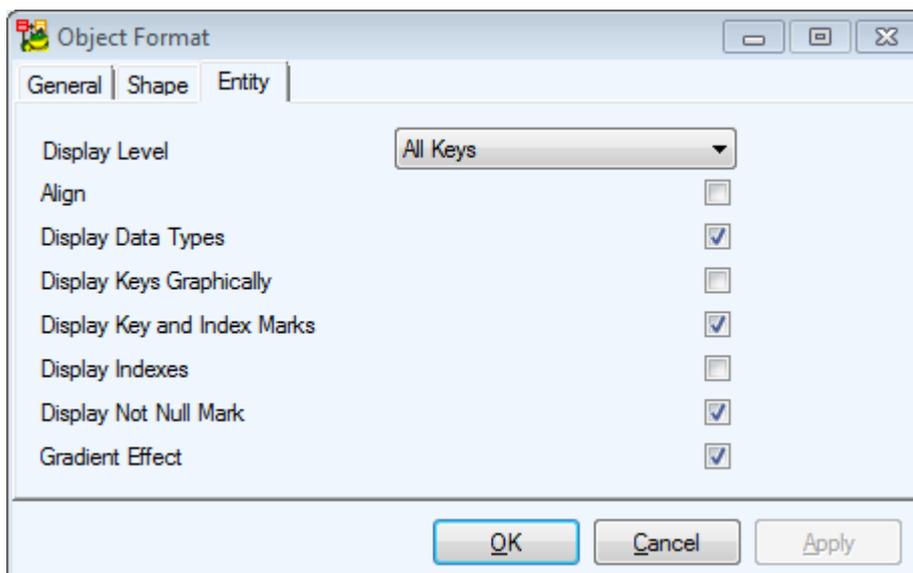
<b>General Tab</b>	<b>Description</b>
Auto Complete	<p>This option automatically adds newly created objects to all Workspaces where this option is enabled.</p> <p>Select <b>Settings   Options   Physical Model   Workspace</b> tab where you can:</p> <p>a] Check the <b>Auto Complete</b> checkbox</p> <ul style="list-style-type: none"> <li>• All newly create Workspaces will have this option <b>enabled</b>.</li> </ul> <p>b] Uncheck the <b>Auto Complete</b> checkbox</p> <ul style="list-style-type: none"> <li>• Except for the All Items workspace, all newly created workspaces will have this option <b>disabled</b>.</li> </ul>

**i** Note: The **All Items** workspace is intended to serve as a main workspace for your models, containing all model objects. By default, all newly created objects in other workspaces are created in **All Items** workspace as well. This can be disabled by unchecking **Auto Complete** option in the **Workspace Format** dialog (right-click the workspace in **Model Explorer**).

Hide Line Captions	Select this option to hide relationship names on Workspace.
Font Settings	Contains several font settings such as font type, style, size etc.
<b>Shape Tab</b>	<b>Description</b>
Recalculate Size	Select this option to automatically adjust size of an object (entity, note) to fit the length of the text it contains.
Shadow Effect	Displays shadows in ER diagram.
Use Brush Color for Full Shape	The object brush color will be used as main color of the object shape.
<b>Note Line Tab</b>	<b>Description</b>
End Type 1, 2	You can select endings for note lines here.
<b>Entity Tab</b>	<b>Description</b>
Display options	Defines display options for entities such as the display level, what properties should be displayed etc. Options for Physical model differ from those in Logical model.

### To change format of a particular object

Right-click the object on the Workspace and select **Format**.



**i** TIP:

1. You can arrange objects on Workspace in different layers. [Arrange Objects in Layers](#)
2. If you need to preserve format of a particular object when you change the format of your Workspace, select the **Lock Format** option in the **Object Format | General** tab.

## Select Objects

### To select multiple objects on Workspace:

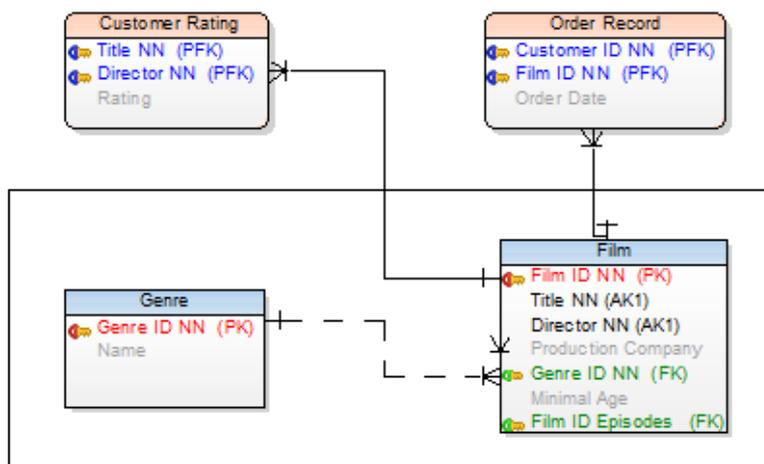
- Hold SHIFT and click the objects you want to select. This option works regardless of the currently used Selection tool.
- Drag your mouse over the objects on Workspace.

**i** Note: If you drag your mouse from the left side, only the objects that are entirely in the selection box will be selected. If you drag your mouse from the right side, all objects that are partly in the frame will be selected.

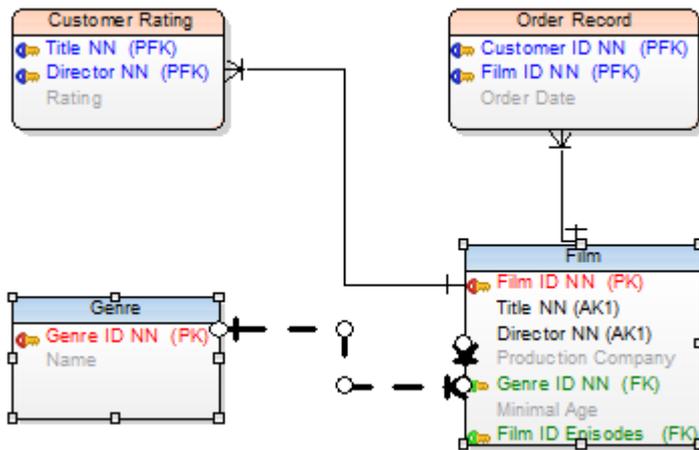
### Scenario

You want to select *Customer* and *Order Record* entities and the relationship between them.

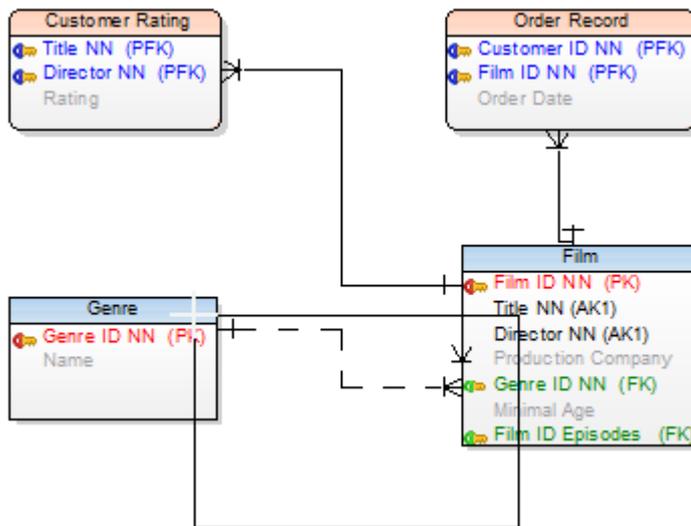
**Solution:** Drag your mouse from the left side as long as both entities are completely in the frame (see the image below).



Release your mouse button. Objects have been selected.



The result will be the same if you drag your mouse from the right side and include even a small part of the two entities in the frame.



**To select more objects on WS by category or schema/owner**

1. Right-click the Workspace | **Select Objects**.
2. Define a category or owner, or category and owner and click **Select**.

**To select child and parent objects of the selected entity on the Workspace**

Right-click the selected entity and select **Select | Parent Objects** or **Select Child Objects** or **Select Parent and Child Objects**.

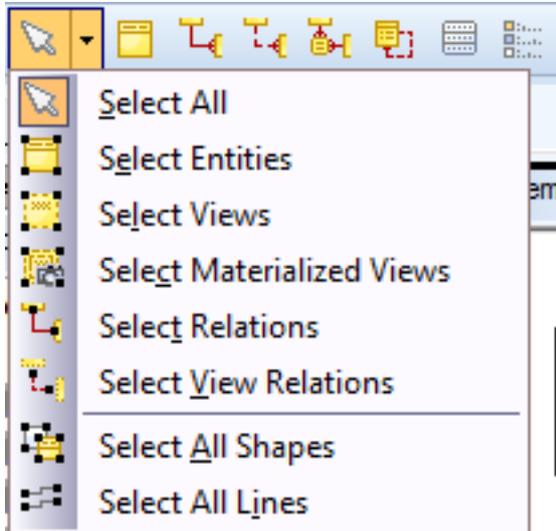
**To select multiple objects in Model Explorer and grids (e.g. Entity Properties form)**

Use SHIFT or CTRL keys.

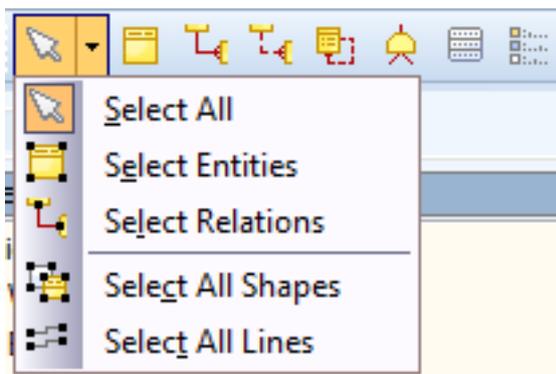
### To select a specific type of objects on Workspace:

There are several tools to select specific objects in Toad Data Modeler. All of them are located under the  button on **Model Objects Toolbar**. Once you select a tool from the dropdown menu, you are only able to select the object type you chose (e.g. with Select Entities tool you are able to select entities only).

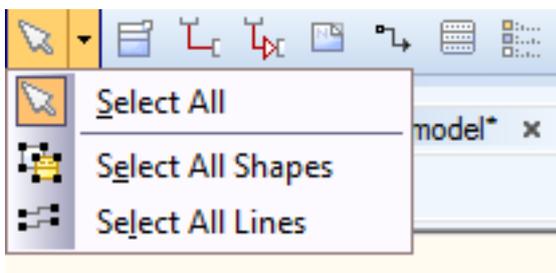
**Physical Model** Selection tool options:



**Logical Model** Selection tool options:



**Metamodel** Selection tool options:



## Align Objects

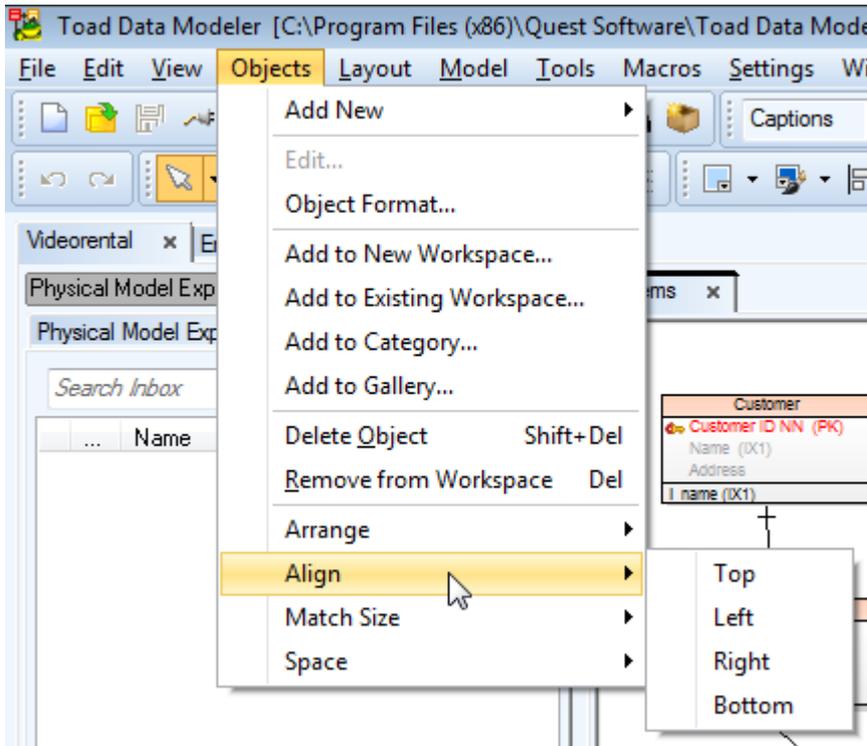
 TIP: When aligning objects manually, you might want to use the **Snap to Objects** function.

## To align objects on Workspace

1. Select objects you want to align.
2. Select align style from the **Alignment Toolbar**.



Additional options for object arrangement and size change can be found in **Objects Menu**.



Arrange

Various options for arranging objects into layers:

- **Bring to front** - brings the object to the top layer.
- **Bring forward** - brings the object one layer up.
- **Send backward** - sends the object one layer down.
- **Send to back** - sends the object to the bottom layer.

Arrange | Setting

Opens **Object Format** dialog and focuses the **Z-order** (layer of selected object, higher number - upper layer)

Align

Aligns multiple selected objects:

	<ul style="list-style-type: none"> <li>• Top</li> <li>• Left</li> <li>• Right</li> <li>• Bottom</li> </ul>
Match size	<p>Matches sizes of multiple selected objects:</p> <ul style="list-style-type: none"> <li>• Width</li> <li>• Height</li> <li>• Width and Height</li> </ul>
Space	<p>Offsets selected objects by the same length:</p> <ul style="list-style-type: none"> <li>• Vertical Equally</li> <li>• Horizontal Equally</li> </ul>

## Rename Objects

### *To rename objects on Workspace*

Select the object (entity, view etc.), press F2.

### *To rename object in Model Explorer*

Select the object and press F2 or right-click and select **Rename**.

## Copy Objects

In Toad Data Modeler you can copy objects between models of the same or different database platforms and versions.

**i** Note: Even though it is possible to copy and paste objects to a different model of different database platform or version, it is encouraged to use **Model Convert** function, even for single objects. The copy-pasting method is faster, but more error-prone, while the **Model Convert** method is slower, but more robust.

**i** TIP: Instead of copying one object multiple times, it is better to add it to a **Gallery**. See **Gallery** for more information.

### *To copy objects on Workspace*

- Drag an object on Workspace, hold **CTRL** and drop the object copy somewhere else on Workspace.
- Select **Edit Menu | Copy/Paste**
- Use CTRL+C and CTRL+V shortcuts

### **To copy objects between Workspaces/Models**

- Select **Edit Menu | Copy** in source model, **Paste** in target model
- Use CTRL+C and CTRL+V shortcuts

### **To copy objects in Model Explorer**

- Drag an object in Model Explorer, hold **CTRL** and drop the object copy on the object type folder (e.g. Entities).

### **To copy objects in object type dialogs (Model Menu | Model Items | Entities, Views etc.)**

- Drag an object in the dialog, hold **CTRL** and drop the object.
- Use CTRL+C and CTRL+V shortcuts

Combinations of these copy methods are possible - e.g. copy from Model Explorer to Workspace.



TIP:

- To copy more objects at once, make multiple selection and use one of the copy techniques.
- Making a Copy of Multiple Objects in Workspace (CTRL+A, CTRL+C, CTRL+V): Before you press CTRL+V to paste the objects, close the **Model Explorer** dialog to accomplish the operation much faster. (The larger your model is, the more significant difference in speed you will notice.)

## **Move Objects**

### **To move objects on Workspace, you can use:**

- Drag&Drop technique
- Keyboard arrows



TIP: To set the size of a step for moving shapes on Workspace, select **Settings | Options | Application | Graphics | Move Objects by (mm/10)** (in tenths of millimeter).

### **To move objects between Model Explorers**

Use Drag&Drop technique. You need to drop the object onto its root folder in the target Model Explorer.

### **To move objects in object type dialogs (Model Menu | Model Items | Entities, Views etc.)**

Use Drag&Drop technique.

### **To move objects between Workspaces/Models**

Use Drag&Drop technique.



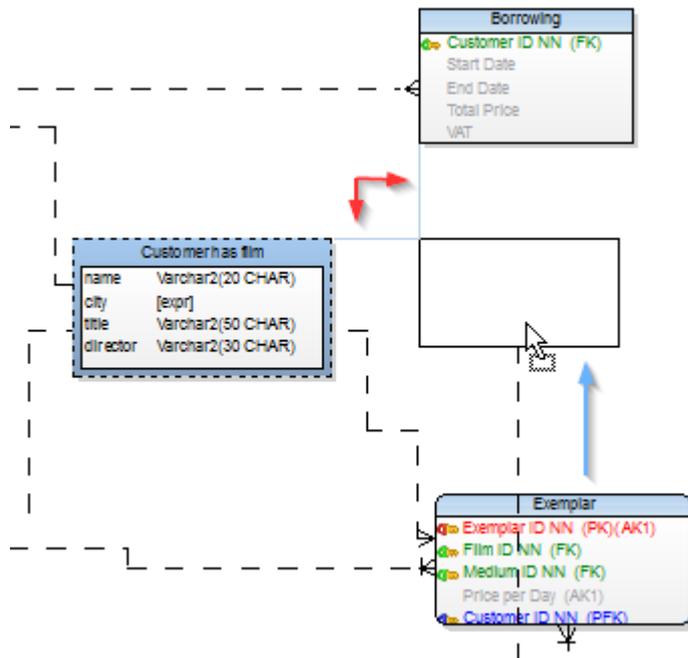
Note: Even though it is possible to move objects from a model to a different model of different database platform or version, it is encouraged to use **Model Convert** function, even for single objects. The Drag&Drop method is faster, but more error-prone, while the **Model Convert** method is slower, but more robust.



TIP: Combinations of the listed methods are possible - e.g. moving object from Model Explorer in one Model to Model Explorer in another Model.

## Snap to Objects

When moving an object on workspace (e.g. entity or relationship line), light blue guidelines appear and help you to align the object more precisely.



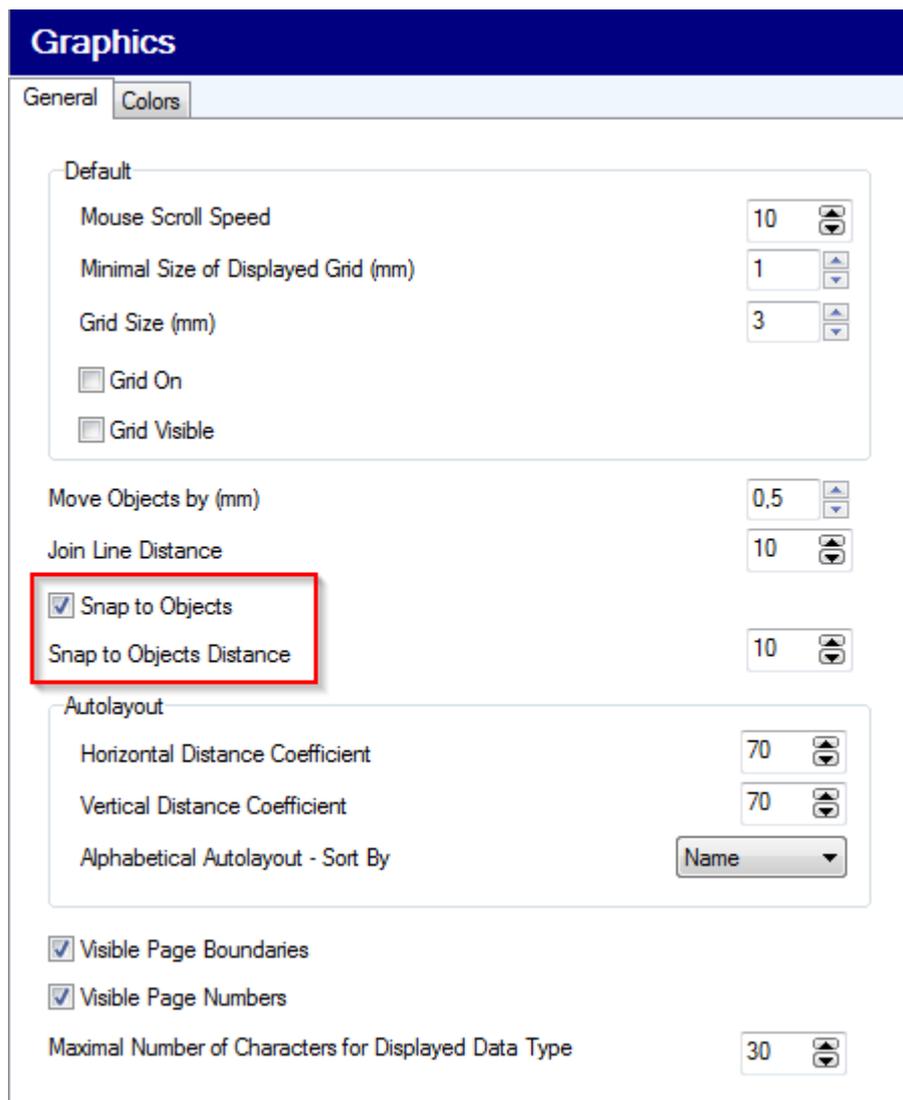
The Snap to Objects option is turned on by default - see  on the toolbar.

### To turn it off

- Click the **Snap to Objects** icon on the **Grid Toolbar**.

or

- Select **Settings Menu | Options | Graphics** and uncheck the **Snap to Objects** checkbox



### ***To turn Snap to Objects off temporarily***

Press and hold ALT Key while dragging an object.

## **Delete Objects**

Objects in Toad Data Modeler can be deleted in two ways:

- **Delete Object (Shift+Del)** - **Deletes** an object with all its shortcuts **from model**.
- **Remove from Workspace (Del)** - **Removes** an object shortcut **from Workspace**. The object is still accessible in the model, only its graphical representation is deleted.

### **Scenario**

Your model has two Workspaces - **WS1** and **WS2**. You have added a new entity to your

model. The entity is placed on both Workspaces (assuming the **Auto Complete** function is enabled). However, you need to have the entity on **WS1** only.

**Solution:** You select the entity on **WS2** and simply press **Delete**. The entity graphical representation on **WS2** will be removed, but the entity is still in your model and can be accessed in **Model Explorer**.

If you would have wanted to delete the entity from your model completely (including all its shortcuts on all Workspaces), you would select **Delete Object** option in **Objects Menu** (or press **SHIFT+Delete**).

### **To remove shortcut of object from Workspace**

Select an object (shortcut of object) on the WS and press **Delete**.

**Tip:** You can delete shortcuts in **Model Explorer** too - select particular shortcut of an object in the **Shortcuts** folder | right-click | **Delete Item**.

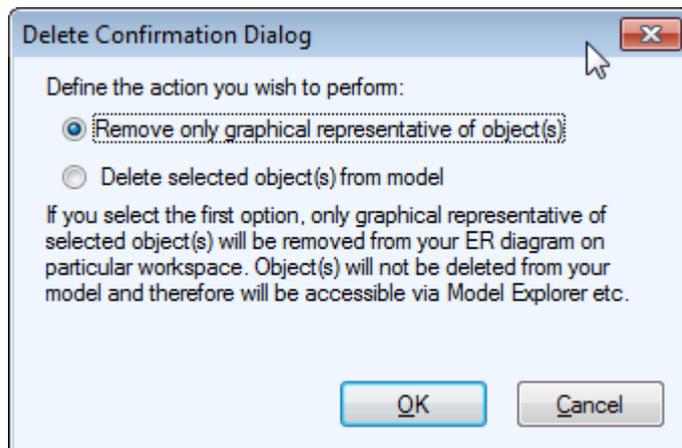
### **To delete objects from model (including all its shortcuts)**

Select the shortcut of object on Workspace and press **SHIFT+Delete**.

**i** TIP: You can also delete the objects:

- In **Model Explorer** - select an object | right-click | **Delete Item**.
- In the Object dialog (**Model** menu | **Model Items** | **Entities**, **Relationships** etc.) - select the object and click **Delete**.

When you press **Delete** in your ER diagram, the following message will display:



Select the action you want to perform.

### **To set the default Delete options in Toad Data Modeler:**

Select **Settings** | **Options** | **Dialog Boxes** | **Other** tab.

<b>Option</b>	<b>Description</b>
Display Dialog	Whenever you press <b>Delete</b> or <b>SHIFT+Delete</b> in your ER diagram, the <b>Delete Confirmation Dialog</b> will pop up and you will be able to select what action you want to perform in particular case.

Option	Description
Remove Graphical Representative of Object	If this option is selected, the <b>Delete</b> will always remove selected shortcut(s) of object from particular Workspace. <b>SHIFT+Delete</b> will have to be used to completely delete an object from your model.
Delete Object	If this option is selected, the <b>Delete</b> will completely delete selected object(s) from your model, including all the object shortcuts. There is no option to only remove selected shortcut(s) of object. (The <b>Delete</b> will replace <b>SHIFT+Delete</b> .)

## Find Objects

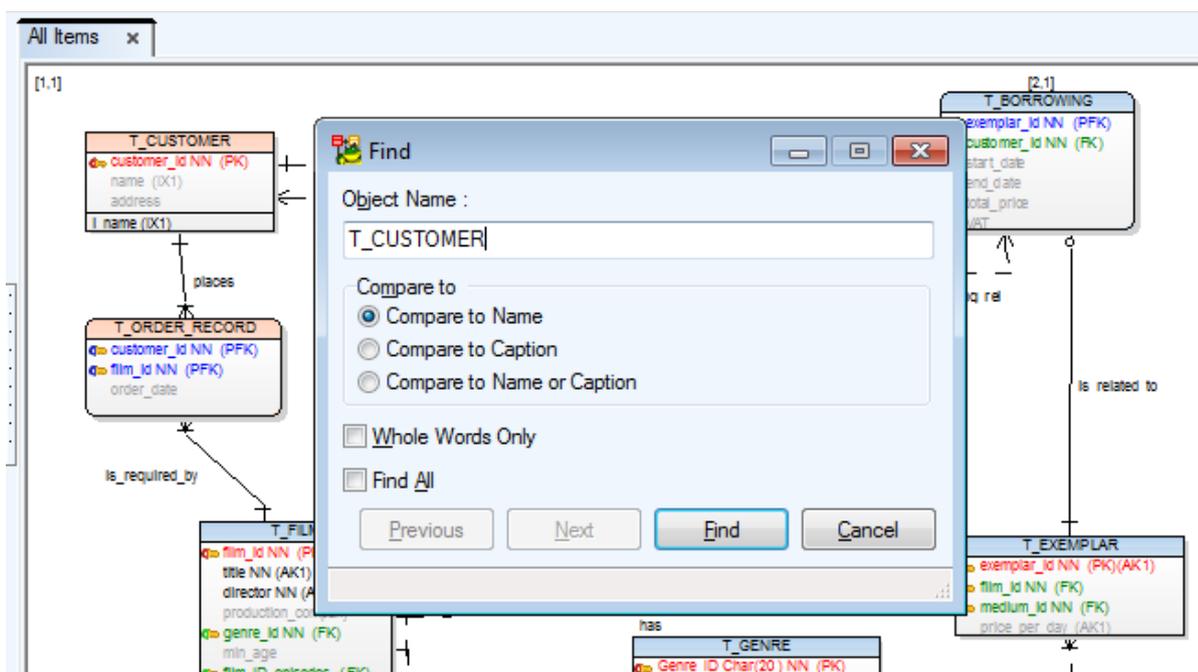
### To quickly find any object on any Workspace

- Use the Quick Search function (CTRL + F). It helps you find any object swiftly by showing you instant results as you type.

For more information see [Quick Search](#).

### To find a specific object on specific Workspace in your large model

- Use the **Find** function
  1. Activate the Workspace in **Application Window**.
  2. From **Edit Menu**, select **Find** (SHIFT + CTRL + F shortcut).
  3. Write the name of the searched object to the **Object Name** box and click **OK** to find and highlight the object on the Workspace.

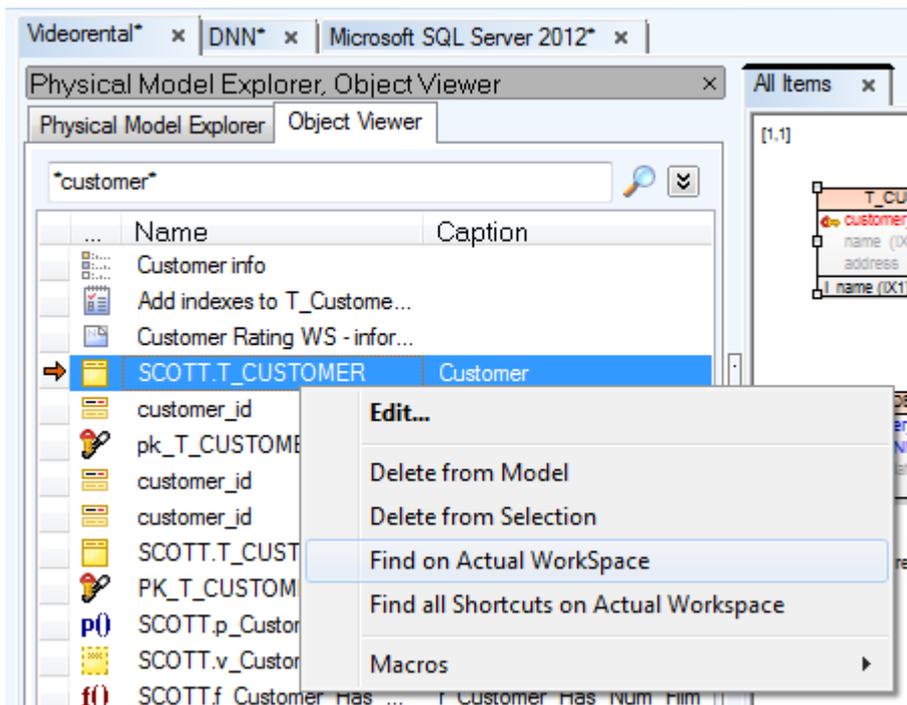


- **Model Explorer**

Right-click an object in Model Explorer | **Find on Workspace**. The object will be focused and highlighted.

- **Object Viewer**

Object Viewer displays a complete lists of all objects in your model. You can find and highlight any selected object (and also its shortcuts) by right-clicking it and selecting **Find on Actual Workspace** or **Find all Shortcuts on Actual Workspace**.



Object Viewer

## External Objects

External objects are objects dynamically linked from another model. This feature can help you to divide a large model into several smaller elements.

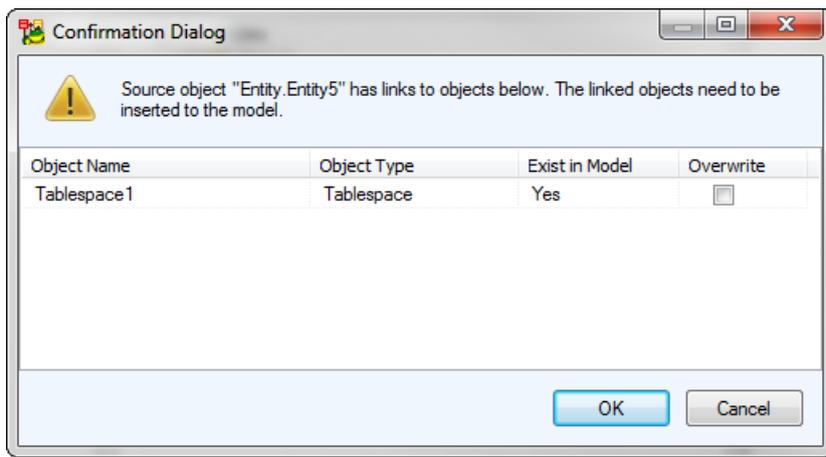
External objects are read-only, and therefore it is not possible to insert any objects that would have effect on dependencies such as relations. However, it is possible to add entities, procedures, domains and others.

External objects are synchronized with their source objects *only* when prompted. Both models need to be available during the synchronization process.

### **Add External Objects to a Model**

- Right-click the root of a model in **Model Explorer** or **Application View** and select **External Sources** to import external objects
- Both models must be opened in Toad Data Modeler in order to use external sources
- Select the source model to import from and check objects you want to use in your destination model

- If an object with the same name as the imported object already exists in the model you will be asked about overwriting it:
  - Click **Yes** to overwrite the original object with the external object
  - Click **No** to insert the external object and keep the name
- If asked about inserting linked objects:
  - Click **OK** to insert external objects and the objects that are linked to it
  - Click **Cancel** to cancel import. The external object will not be inserted into the model
  - For linked objects that are named identically to already existing objects:
    - Check **Overwrite** to replace the original target object with the imported linked objects
    - Leave **Overwrite** unchecked to keep the names of both objects



**i NOTE:** Exist in Model field has four states:

- No - There is no object with the same name as the imported object
- Yes - There is a non-external object with the same name as the imported object
- Mapped External - There is a external object with the same name as the imported object, which comes from the same source as the currently imported object
- External - There is a external object with the same name as the imported object, which comes from a different source than the currently imported object

### **Manage External Objects**

- Right-click a model and select **External Objects | Update All** to easily update all external objects
- Select **Window | External Dependencies Explorer** to manage imported and exported objects
- **Imported Objects** shows a list of objects from an external source and **Exported Objects** shows a list of objects inserted into other models
- Click **Update** to update the imported objects if they had been modified in the source model
- External objects are marked by an icon (📄) in **Designer** and in **Model Explorer**

## External Dependencies Explorer Actions

Button	Description
Refresh	Refreshes the list of external objects
Check Item	Performs a validity check against the external source model
Check All	Performs a validity check for all external objects against the external source models
Delete Object	Deletes the object in the target model
Create Reference in Source Model	Creates a dependency reference in the source model in order to indicate that the object has been linked from other models
Delete Reference	Deletes the reference to the target model from this model
View Object/Item	Displays properties of the object
Open Model	Opens the model that contains the external object
Update from External Source	Updates the model with changes from the external source
Propagate to External Source	Updates the source model with changes from this model

**i** **NOTE:** There is one exception when the external source is not read-only. Create an object in the source model and synchronize it with your target model. Then create a relation to the object in the target model and an attributed is created in consequence. This attribute will be propagated to the source model when synchronized.

## About Shortcuts of Objects

In Toad Data Modeler, you can create multiple graphical representatives of an object - **Shortcuts**.

- A shortcut is a graphical representative of an object in the Workspace (WS). It is **not** a copy of an object.
- A shortcut has the same name as its object + a number indicator.
- A shortcut has the same data properties as its object. You can edit any shortcut and all changes will be applied to its object. Also, changes made to an object will be reflected on its shortcuts.
- An object can have none or many shortcuts. Their number is not limited.
- You can set individual graphical format for each single shortcut of the same object.

You can create shortcuts of these objects:

- Entity
- Relationship
- Inheritance
- View (Materialized View)
- Note
- Stamp

- Category (caption of Category)
- Image

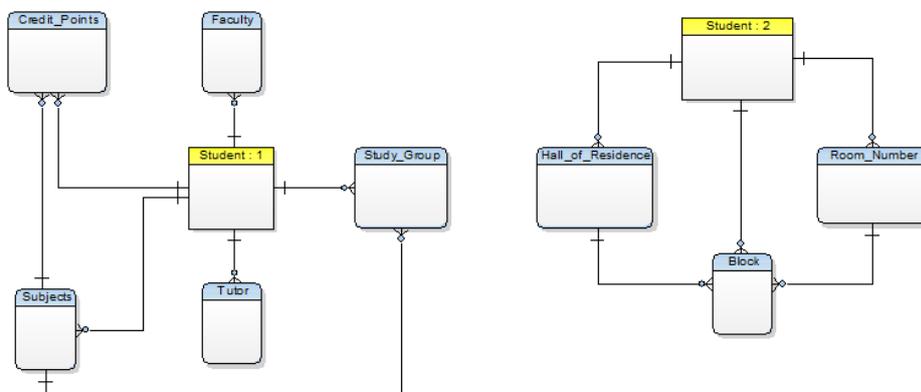
Shortcuts help you to organize your large models. Let's say you have a model, where an entity is related to a large number of other entities. You can break the model down into multiple workspaces and create a shortcut of the entity for each workspace. The structure of model is not affected, but you get to see the important things more clearly.

For further distinction of model objects using colors, see [Categories](#).

### Scenario

You have a *Student* entity, which is related to many other entities across the entire model. You want to divide the model into two blocks and you also want the *Student* entity to be contained in both of these blocks.

Solution: A shortcut of the *Student* entity has been created and used in another part of the model. This way, the model has been logically divided into two blocks and became much clearer and better arranged.



## Create Shortcuts

Toad Data Modeler allows you to create shortcuts of objects on Workspaces in many ways.

### To add a shortcut within a Workspace

Drag&Drop techniques and CTRL+SHIFT

#### Example:

Create a shortcut of the *Customer* entity on one WS - *Borrowing WS*:

1. Click the *Customer* entity on the *Borrowing WS* and hold the mouse key down.
2. Press CTRL+SHIFT keys and hold them down.
3. Drag the *Customer* entity to any place on the work area (the mouse cursor will change its appearance).
4. Release the mouse button and the keys.

**i** Note: This method cannot be used for creating shortcuts of relationship and inheritance objects.

### To add a shortcut from Model Explorer to Workspace

Drag&Drop techniques from **Model Explorer** to particular WS

### Example:

1. Activate the Workspace where you want to add a shortcut of the *Customer* entity.
2. In **Model Explorer**, find the *Customer* entity.
3. Click the *Customer* entity and hold the mouse key down.
4. Drag the *Customer* entity to the WS (the mouse cursor will change its appearance).
5. Release the mouse button.

### **To add a shortcut of selected object(s) from one Workspace to another Workspace**

Right-click the selected object(s) (e.g. entity) on the WS and select:

- **Add into Workspace** to select the particular WS.
- or
- **Add into New Workspace** to create a new WS and add the shortcut there. The layout and format of the shortcuts remain preserved when you use this option.

### **To add a shortcut of objects related to an entity on Workspace (including relationships)**

Right-click the entity and select from the following options:

- **Fill Parent Objects** to add shortcuts of parent objects of the selected entity to the WS.
- **Fill Child Objects** to add shortcuts of child objects of the selected entity to the WS.
- **Fill Parent and Child Objects** to add shortcuts of parent and child objects of the selected entity to the WS.

### **To add a shortcut of object(s) to the selected Workspace**

Right-click the selected Workspace and select **Add All Model Objects to Workspace** to add shortcuts of all objects that exist in your model.

## Edit Shortcuts

### **To edit object property**

Double-click the selected shortcut on the Workspace.

You can edit any shortcut you want. The changed properties will be automatically applied to the parent object and to any other existing shortcuts of the object.

### **To change graphical format of a shortcut**

Right-click the selected shortcut on the Workspace and select **Format**.

The changed graphical format will be applied only to the selected shortcut. Toad Data Modeler allows you to define a different graphical format settings for each shortcut of the same object.

## Remove and List Shortcuts

### **To remove a shortcut of object on the Workspace**

Select a shortcut on the WS and press **Delete**.

Only the selected shortcut will be removed from the Workspace. The object itself still exists in the model.

**i** TIP: More ways to do this:

1. Right-click the shortcut in **Model Explorer** and select **Delete Item**.
2. To remove more shortcuts on WS at once, make multiple selection and then press **Delete**.

## List Shortcuts

To list all shortcuts that exist in your model see **Model Explorer** and the following folders.

### List all shortcuts by object type (entity, relationship, inheritance)

Select the particular object folder (e.g. **Entities**) | **Shortcuts** folder. Here, you can also see names of the Workspaces where the particular shortcuts are located.

### List all shortcuts by Workspace

Click the **Workspaces** folder | particular workspace folder (e.g. *Ordering*) | **Entity Shortcuts, Relationship Shortcuts, View Shortcuts**.

### To find shortcuts of objects on the Workspace quickly

Double-click the shortcut in Model Explorer to highlight it on the Workspace.

**i** Note: Shortcuts can also be found in **Workspace Properties** which can be opened by right-clicking a workspace in **Model Explorer, Application View** or **Designer** and selecting **Edit**.

## Shortcut Right-Click Options in Model Explorer

**Right-click a shortcut in Model Explorer to see the following options:**

Option	Description
Add Object to Gallery	Adds the selected object to gallery.
Edit	Opens the Object <b>Properties</b> dialog.
Select on Workspace	Highlights the shortcut on the Workspace.
Add into Workspace	Adds the shortcut to the WS that you select from the list of existing Workspaces.
Add into New Workspace	Creates a new WS and adds the shortcut there.
Macros	Opens the list of available macros.
Delete Item	Removes the shortcut from Workspace.

**Right-click an entity shortcut in Model Explorer to see other options:**

Option	Description
Add	Creates a new item in the entity: <ul style="list-style-type: none"><li>• Attribute</li><li>• Key</li><li>• Index</li></ul>
Fill   Parent Objects	Displays shortcuts of parent objects of the selected entity on Workspace.
Fill   Child Objects	Displays shortcuts of child objects of the selected entity on Workspace.
Fill   Parent and Child Objects	Displays shortcuts of parent and child objects of the selected entity on Workspace.

## 2-D Shapes

Toad Data Modeler allows you to insert the following 2-D shapes to your ER diagram:

- Note
- Line
- Stamp
- Categories
- Image
- Rectangle
- Ellipse
- Text
- Label Quadrangle
- Label Ellipse

The objects are available on **Graphics Objects Toolbar** and in the **Objects Menu | Add New** for both physical and logical model.

### **To change format of these objects**

Right-click the object and select **Format**.

**i** TIP: Feel free to arrange and order the objects on Workspace. [Arrange Objects in Layers](#)

## Note and Line

A note can refer to a model, Workspace, particular entity, attribute, relationship etc.

### To add a note to your model

1. Click  icon on **Graphics Objects Toolbar** or select **Objects | Add New | Note** and click the work area.
2. Double-click the Note to edit it.
3. Write a text on tab **General**. The automatic word wrap function is available.

**i** TIP: To change a format of Note, right-click it and select **Format**. To adjust the size of a Note to a length of text contained, click the **Shape** tab and select **Recalculate Size**.

### Note Shortcuts

You can also create shortcuts of Notes:

---

#### Scenario

You have a Note which describes your *Customer* entity. The entity is places on multiple Workspaces. You would like to have the Note on every Workspace where is the entity.

**Solution:** Create multiple shortcuts of your Note object and place each onto a different Workspace.

---

To make a connection between a Note and an object that the note relates to, you can use a **Line**.

### Lines

#### To add a Line

1. Click  or select **Objects | Line**.
2. Click the object and then click the Note.

#### To change the look of the lines on the Workspace

1. Right-click the WS and select **Workspace Format**.
2. Click the **Line** tab and select the line end type.
3. Right-click the line and select **Line Style**.

### Image

You can add logos and other images to your ER diagrams and then relate them to any object on Workspace using a Line.

#### To insert an image

1. Click  icon on **Graphic Objects Toolbar** (or select **Objects | Add New | Image**).
2. Click on the Workspace where you want to insert the image.
3. Select the image from the **Open** dialog and click **Open**.

**i** | Note: Images that you insert to your ER diagrams aren't saved together with your model.

### To edit an image

Double-click the image on the Workspace.

### Image in Model Explorer

Take notice of the **Image** item in Model Explorer. From here, you can manage your images as well.



### Image and Shortcuts

You can create shortcuts of an Image on every Workspace of your model or create multiple shortcuts on one Workspace.

## Stamp

Add a Stamp to the Workspace to display information about your model such as Author, Company, Date of Creation etc.

### To add a stamp

1. Click  icon on **Model Objects Toolbar** (or select **Objects | Add New | Stamp**), and click anywhere on Workspace.

Project	Videorental Project
Model	Videorental
Author	Radim Mario Tkacik
Firm	Quest Software, Inc.
Version	
Date of Creation	3/15/2007 08:50
Last Change	2/5/2008 13:01

2. To change the format of the Stamp, right-click it and select **Format**.

### To edit a stamp

Double-click the Stamp on Workspace. **Model Properties** dialog displays, here you enter the information which is shown in Stamp.

### Stamp and Shortcuts

You can create shortcuts of a Stamp on every Workspace of your model or create multiple shortcuts on one Workspace.

## Caption of Categories

Caption of Categories is an object that lists all categories used and displayed on your Workspace and their respective colors.

See [Categories](#) for more information.

### To add Caption of Categories

- Click  icon on **Model Objects Toolbar** and place Caption of Categories anywhere on your Workspace.

### To change format of Caption of Categories

- Right-click and select **Format**.

### Caption of Categories and Shortcuts

You can create shortcuts of Caption of Categories and add them to several Workspaces.

#### Scenario

You have created several Categories and colorfully distinguished the objects on Workspace. As some of the objects occur in multiple Workspaces, you would like to display the Caption of Categories there too.

#### Solution:

1. Right-click the existing **Caption of Categories** and choose **Add into Workspace**.
2. Select a Workspace from list.

## Application Variables

In Toad Data Modeler you can use application variables in:

- Names/captions in **Physical Model Explorer** and **Designer** - use the percent button () to enter the selected variable in the current cursor location
- **DDL Script** and **Change Script Generators**
- **Report Generation** - Check **Resolve Application Variables** in **Report Wizard | Options**
- **Default Values** (e.g. relationship names) - Select **Settings | Default Values** to adjust default values for objects
- [About Templates](#)
- Certain properties (most notable ones are SQL, Before Script, After Script)

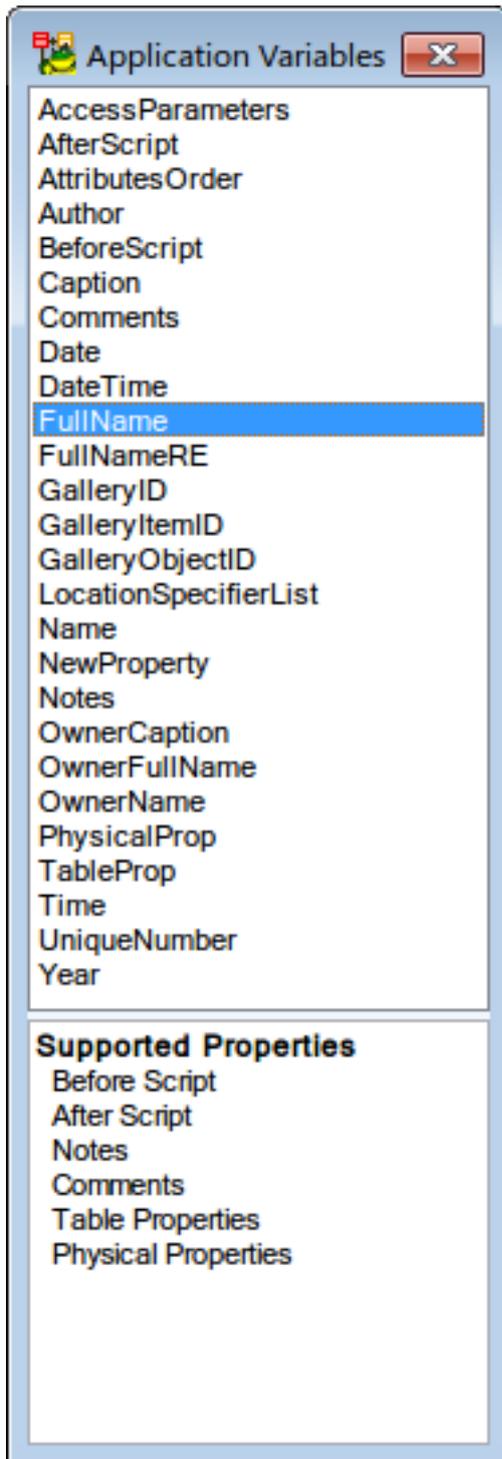
**i** Note: To see where you can use application variables, see **Expert Mode Menu | Reference Guide**. If a property has **Resolve Application Variables** attribute, you can use application variables within it (e.g. *PEREntity - BeforeScript*).



Application variables are usually resolved during SQL script generation.

## Accessing Application Variables

Application variables can be easily accessed via the  which can be found in object properties forms. When you click it, the Application Variable dialog opens and displays available variables and in which properties they are supported.



## Syntax of Application Variables

1. <%ApplicationVariableName%> - Example: <%Date%>
2. <%<% ApplicationVariableName %>%> - Example: <%<%Date%>%>

Syntax 2) is useful in **Templates** and **Default Values**. Variables using this syntax will be resolved only during SQL Script/Report generation as opposed to variables with syntax 1) which are resolved immediately after creating an object.

**i** Note: OwnerName, OwnerCaption - Explanation: E.g. For attribute it is an entity, for entity it is a model. It has nothing to do with object Owner/Schema.

## Application Variables - Examples

### Application Variables and User Template

The screenshot shows the 'Template Editor' window in Toad Data Modeler. The 'Model Type' is set to 'Oracle 10g'. A table lists various object categories and their templates. The 'Complete' template is selected, and its body is displayed in the editor below.

Object Category	Parent	Template Name	Default	Active	Status	Template
<b>Entities</b>						
Attributes						
Check Constraints						
Check Constraints						
Triggers						
- SQL	Model\Entity	Body	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		BEGIN /trigger_body*/EN
SQL	Model\Entity	Complete	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Domains						
Check Constraints						
Defaults						
Check Constraint Rules						

```

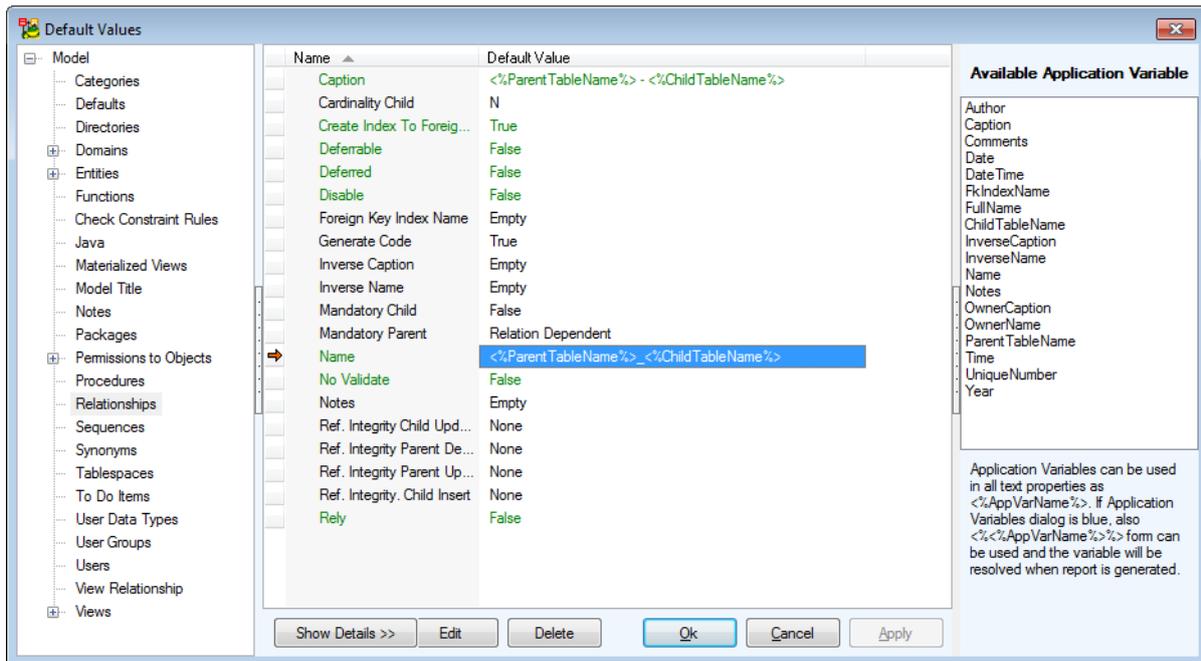
9
10
11     NOTES:
12
13     *****/
14
15     CREATE OR REPLACE TRIGGER <<Full Name>>
16     AFTER /*BEFORE*/
17     INSERT /*UPDATE | DELETE*/
18     ON <<Table Full Name>>
19     BEGIN
20     /*trigger_body*/
21     END
  
```

### Application Variables and Default Values

**Example:** Define a new name and caption for relationships of your model.

Caption: <%ParentTableName%> - <%ChildTableName%>

Name: <%ParentTableName%>\_<%ChildTableName%>

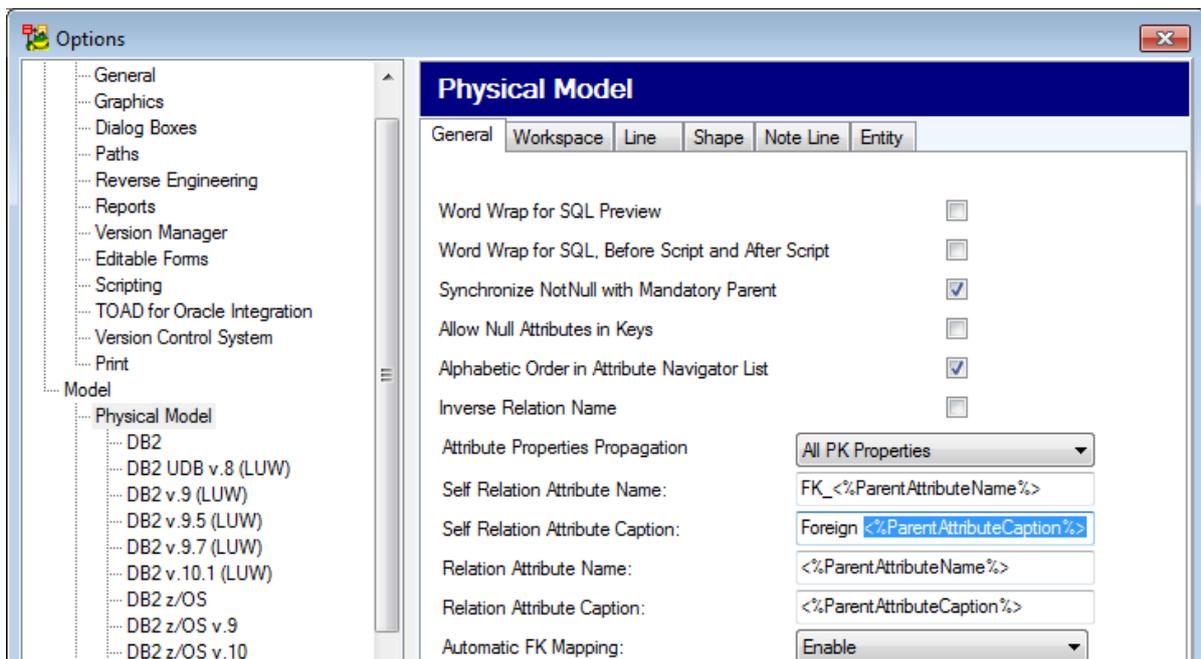


## Application Variables and Self-Relationships

Possibility to define a name for propagated attributes in self-relationships (e.g. via prefix, suffix etc.)

Select **Settings | Options | Physical Model | Self Relation Attribute Name/Caption**.

**Example:** Name: FK\_<%ParentAttributeName%>, Caption: Foreign <%ParentAttributeCaption%>



## Application Variables and Text Properties

You can use some text properties of particular objects in a similar way as application variables.

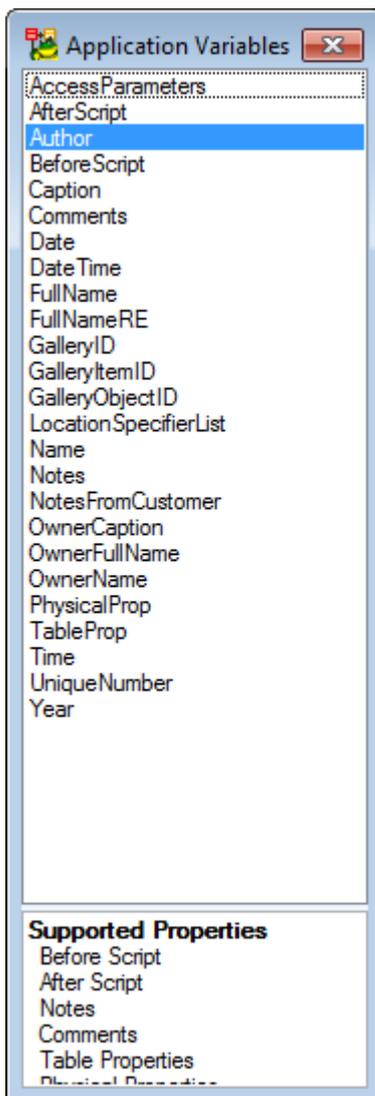
**Example:** You want to see your comments in a pop-up window on the Workspace whenever you point your mouse cursor at the entity name. But comments are not displayed this way, only notes. OR You want to generate notes in SQL script.

Possible solution: In the **Entity Properties** form,

1. On tab **Notes**, write the text, e.g. *My Description*.
2. On tab **Comments**, write: `<%Notes%>`.
3. Confirm **Apply**.
4. See the **SQL Preview** tab.



5. Click  to display a quick help bar with application variables that are possible to use in entity.



**Example:** Name is property of entity. It is of widestring data type, so you can use application variable `<%Name%>`. Author is not property of entity. However, you can use the application variable

<%Author%> in entity. - All the application variables that you can use for entity are available in the list.

**i** Note:

- Double-click the selected application variable to use it in the box/place where you have your cursor.
- Press Esc to close the dialog.

## Application Variables in Wizards

**Sync & Convert Wizard** | page **Settings** and **Report Wizard** | page **Options** and the **Resolve Application Variables** checkbox.

Uncheck the checkbox to compare models/generate report without resolving the application variables.

Check the checkbox to compare models/generate report while resolving the application variables.

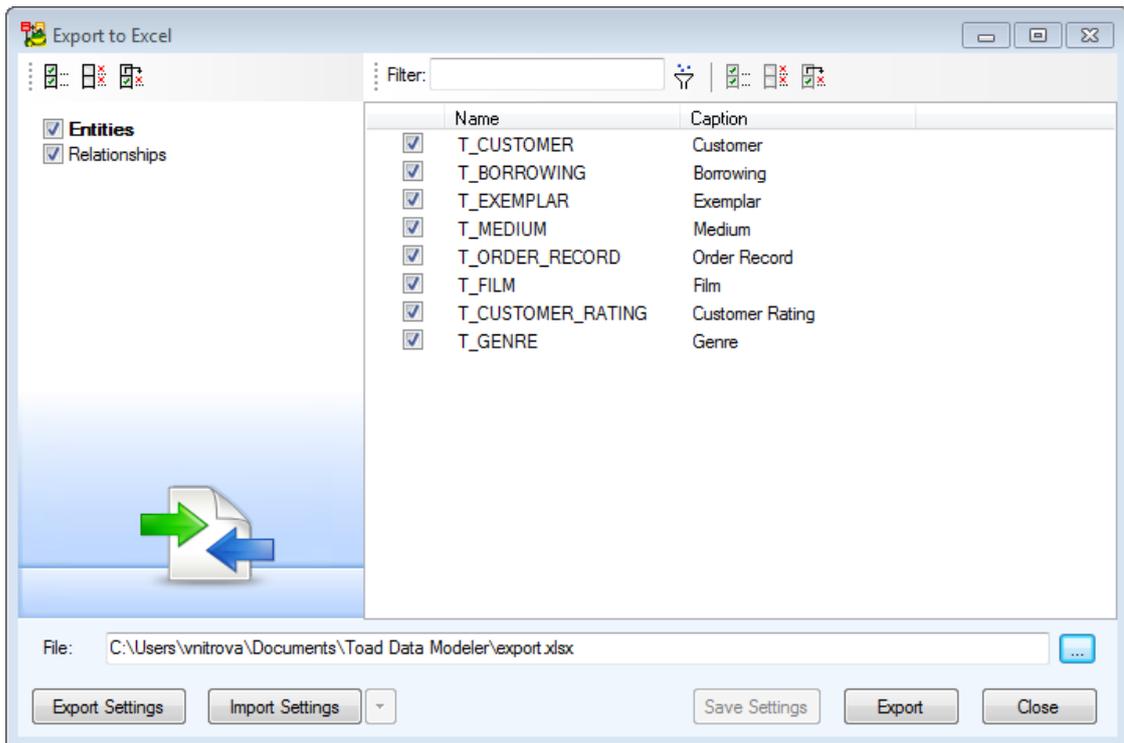
## Export/Import - Microsoft Excel

Toad Data Modeler allows you to export metadata to Microsoft Excel. Comments, notes and other properties can be modified and the file can be then imported back into Toad Data Modeler.

### *To export your model to a Microsoft Excel document*

1. Select **File Menu** | **Export** | **Export to Excel**.
2. Select the entities and relationships you want to export.

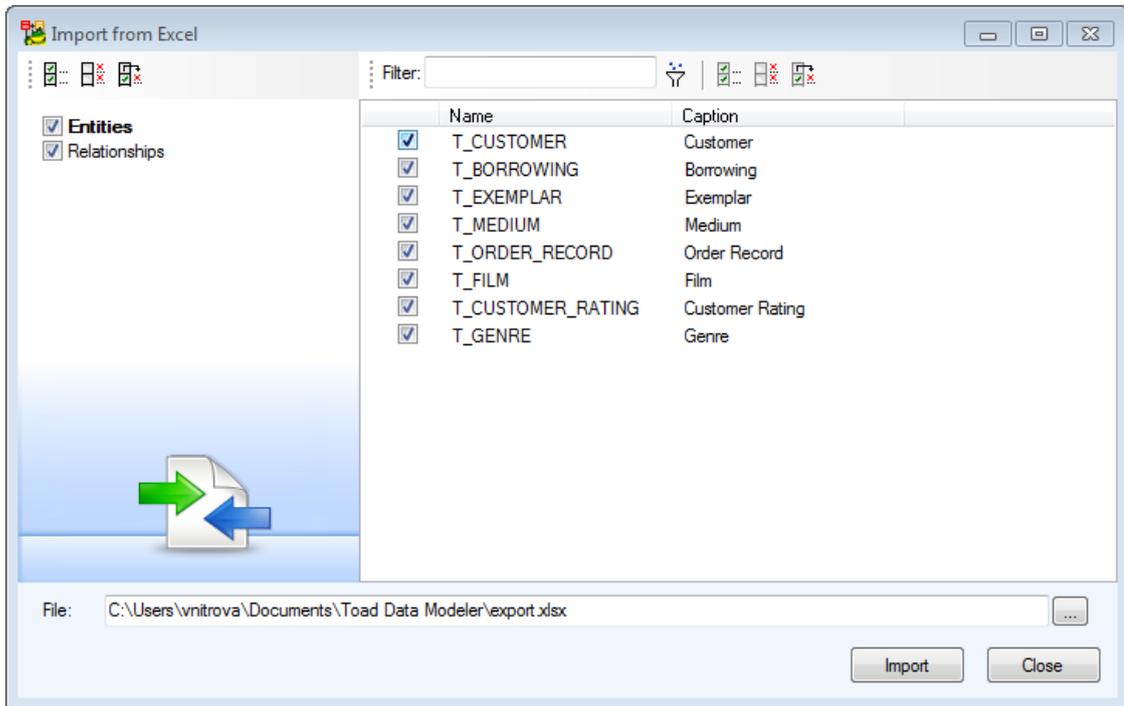
3. Click **Export**. A new Microsoft Excel document will be created in the specified destination.



### ***To import the data back from a Microsoft Excel document***

1. Select **File Menu | Import | Import from Excel**.
2. Locate the Excel document and click **Open**.
3. Select which tables and relationships should be imported.

4. Click **Import**. The existing model will be updated to reflect the changes made in the Excel document.



The following table illustrates the modified properties:

Modification	Property type	Properties
Unlocked	String	Alias Where Having SQL Comment Note BeforeS AfterS Boolean GenerateSQLOnly SelectViewsInText
Locked	List	Attributes From Order Group

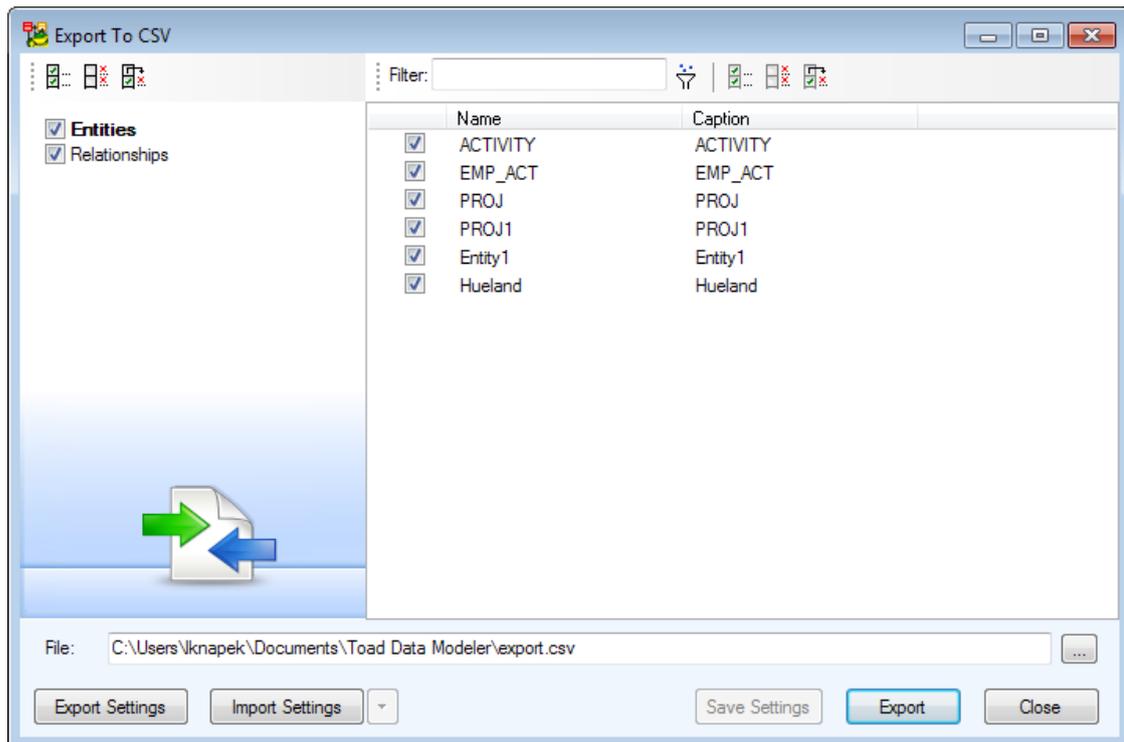
**i** **NOTE:** Aliases of entities and attributes are in columns “From” or “Attributes” in a format: “Object AS alias” in an exported file

# Export/Import - CSV

Toad Data Modeler allows you to export metadata to a CSV file. Comments, notes and other properties can be modified and the file can be then imported back into Toad Data Modeler.

## *To export your model to a CSV file*

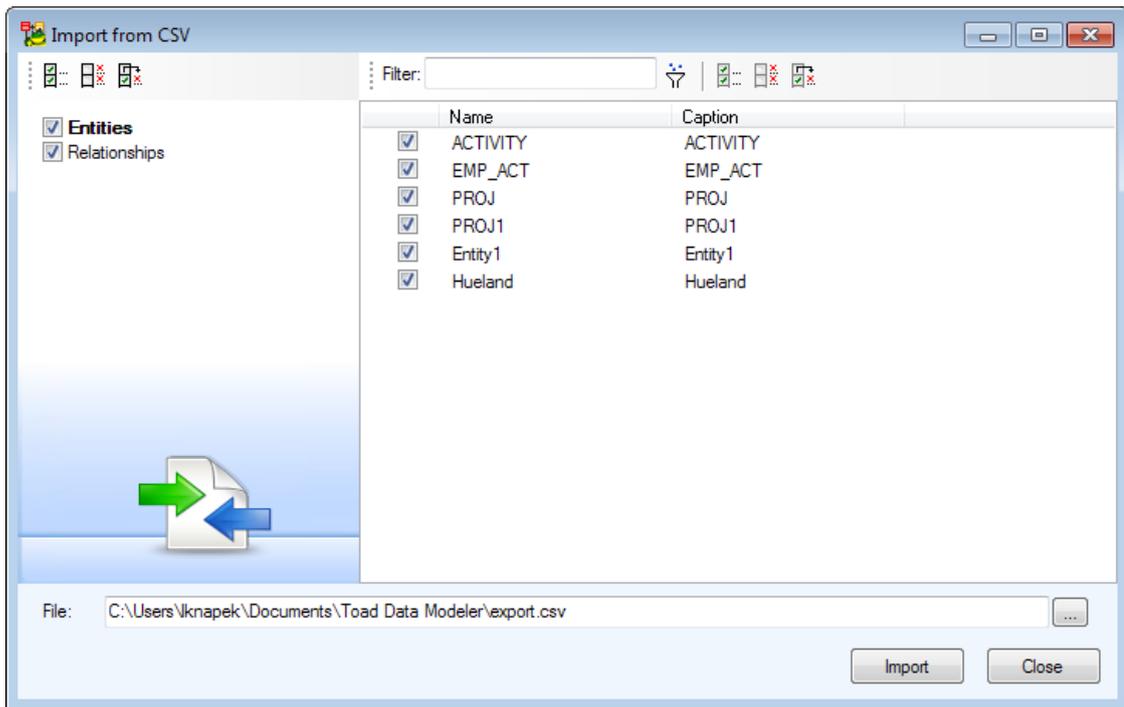
1. Select **File | Export | Export to CSV**.
2. Select the entities and relationships you want to export.
3. Click **Export**. A new CSV file will be created in the specified destination.



## *To import the data from a CSV file*

1. Select **File Menu | Import | Import from Excel**.
2. Locate the CSV file and click **Open**.
3. Select which tables and relationships should be imported.

4. Click **Import**. The existing model will be updated to reflect the changes made in the CSV file.



## Export to Graphic File

In Toad Data Modeler, you can export your ER diagram into the following graphical formats:

- BMP
- JPEG
- PNG
- SVG

### ***To export your ER diagram to a graphic file***

1. Select **File Menu | Export | Export to Image**.
2. Define options on **Settings** tab.

Option	Description
File Type	Choose one of the available formats.
Color	When not enabled, the image will be black and white only.
Pages	Creates an image for each page on workspace. <i>(Not available for SVG)</i>

Option	Description
Paint Frame of Pages	Displays page boundaries. <i>(Not available for SVG)</i>
Scale	Size of the output in percentage scale (Initial value is 100 per cent.)
Width	Changes automatically according to the set percentage scale.
Height	Changes automatically according to the set percentage scale.
Margin	Set Margin for the exported graphics. <div style="margin-left: 20px;">  TIP: Measurement units can be changes in <b>Settings Menu   Options   General.</b> </div>
Destination File	The output image destination path.
Open Folder After Export	When checked, the destination folder is opened once the export is finished.

3. Define other settings specific to the selected graphical format on other tabs.

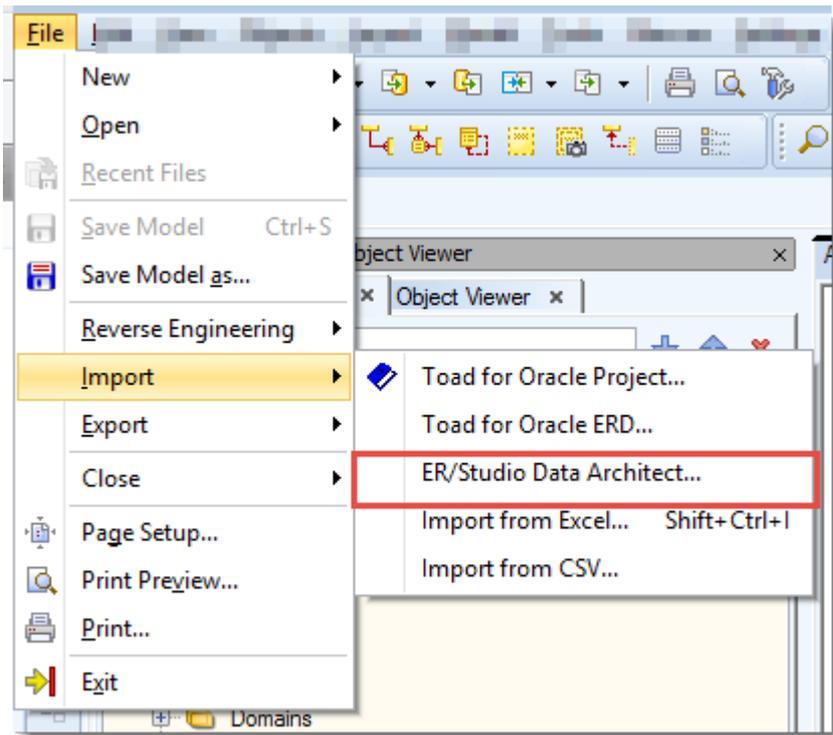
 NOTE: Images of large models are generated as multiple smaller ones joined together in a html file.

## Export to Image Limitations

Please note that there is a limit to the size of the exported images imposed by Windows interface. The maximum size depends on chosen **Pixel Format**, for **32-bit**, the size limit is about **12500x10000 px**. The lower the Pixel Format, the larger image you are able to export. This limitation affects all image formats.

## Import from ER/Studio Data Architect 11

- Toad Data Modeler is able to import physical models for Oracle versions 9, 10, 11, and 12
- ER/Studio Data Architect 11 needs to be installed in order to import physical models into Toad Data Modeler
- Select **File | Import | ER/Studio Data Architect** to import models



## DDL Script Generation Preparation

Before generating a DDL script, you might want to configure the following things:

- **Order of Generated Objects**
- **Script Encoding**

### Order of Generated Objects

Toad Data Modeler allows you to set the order of objects before SQL/DDL script generation.

**Example:** You want to generate Users before User Permissions to a table.

You can change order of the following objects:

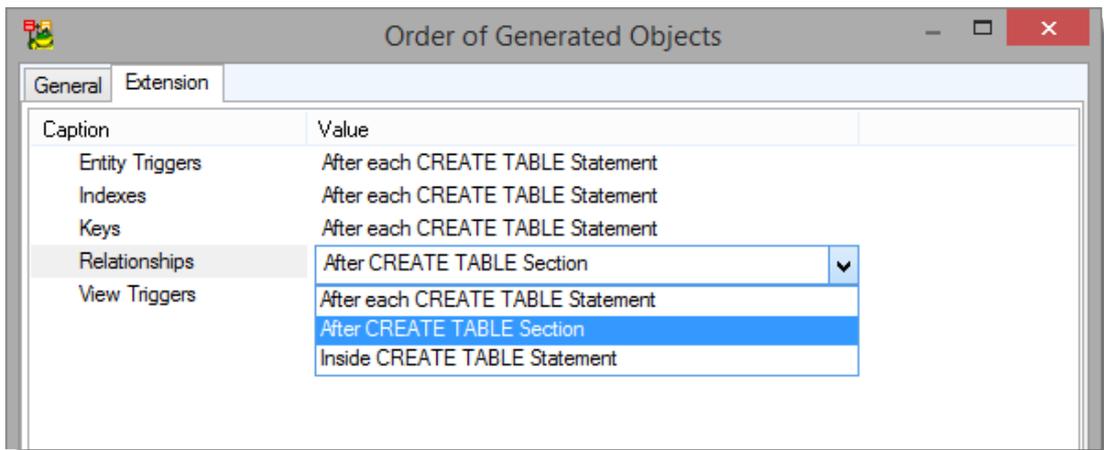
- Domains
- Entities
- Views
- Dictionary types
- Sequences
- Stored procedures
- Functions
- Users
- User data types

### To set an order of objects for SQL/DDL script generation

1. Go to **Model Menu | Order of Generated Objects**.
2. Select an object, or an object type.
3. Use  to move your selection up/down by one step or use drag&drop to place your selection wherever you want.

 Note: To display full names of objects, click on 

4. Switch to **Extension tab** for more options. Press F2 or left-click and hold any of the values to set it as desired.



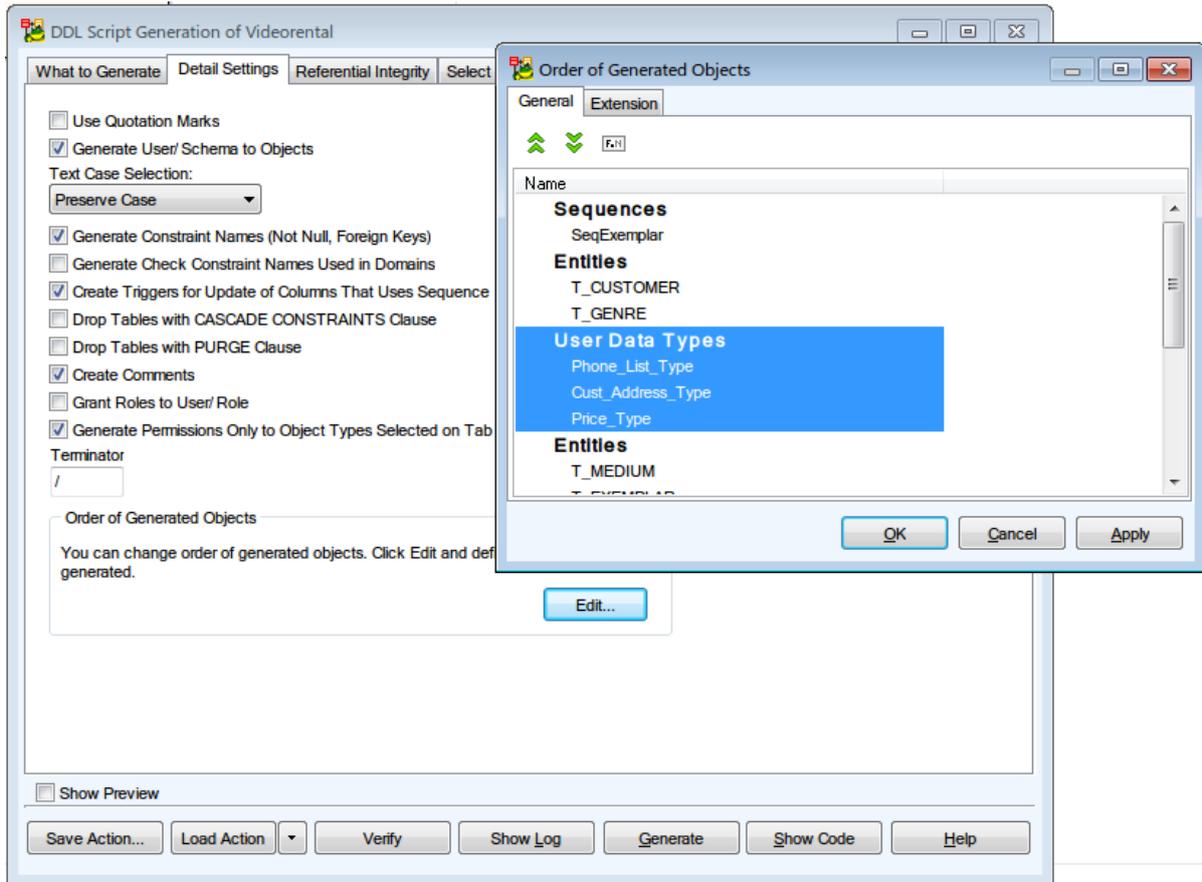
You can configure when should be the listed object types generated.

### To sort objects automatically according to their relationships

1. Select **Model | Order of Generated Objects** to set the desired order of entities
  - a. Select **Move Parent Entity before Child Entity** - lists each parent entity before their respective child entities or
  - b. Select **Move Child Entity after Parent Entity** - lists all child entities after their parent entities

 Note: When you set *After CREATE TABLE Section* value for an object, the object will not be shown in **Entity SQL Preview**.

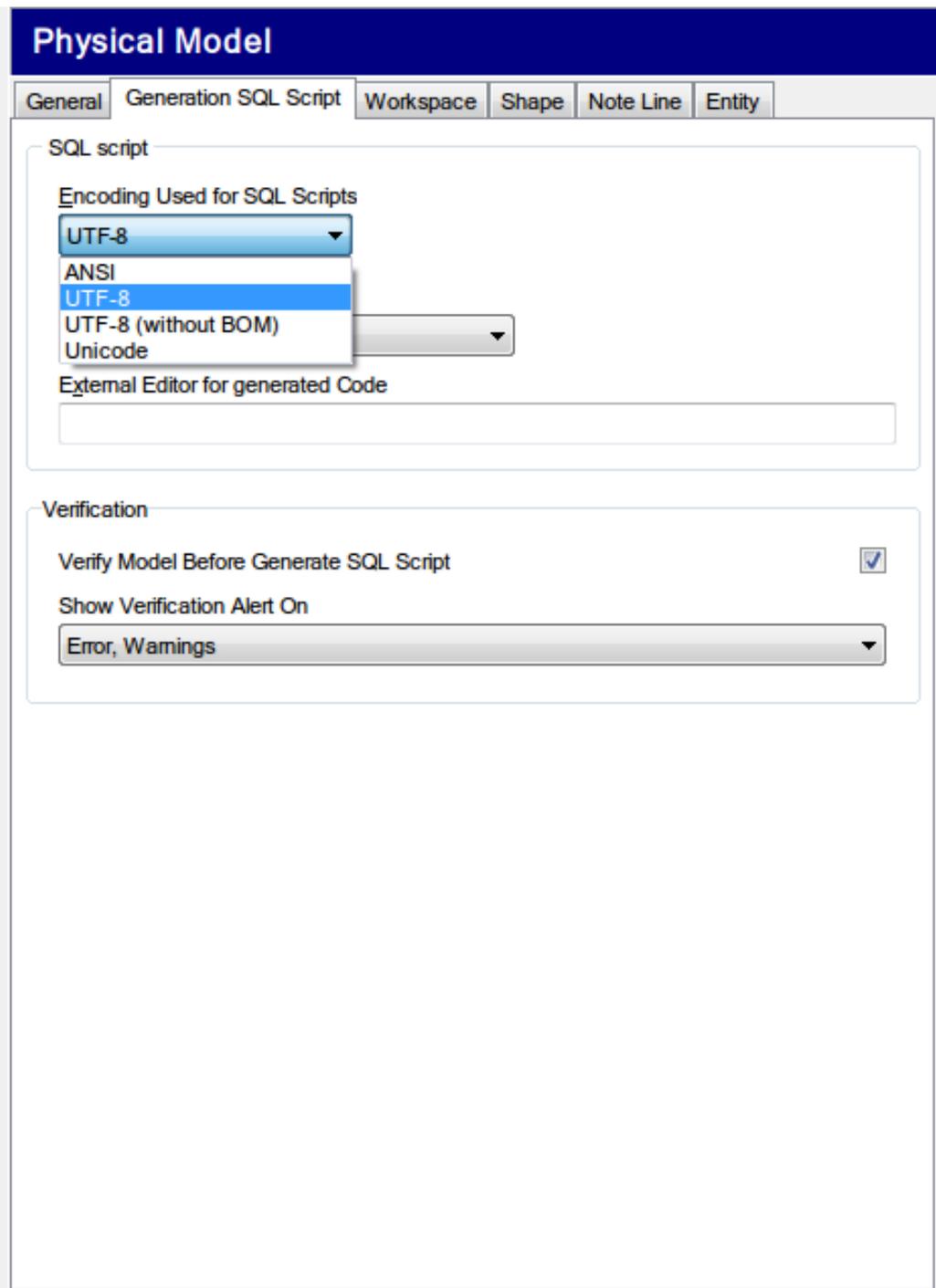
**Order of Generated Objects** dialog is also accessible from **DDL Script Generation** dialog | **Detail Settings** tab| **Edit**.



## Script Encoding

### To set encoding for generated scripts

1. Go to **Settings Menu | Options | Model | Physical Model | SQL Generation Script tab | SQL Script** section.
2. Choose the preferred encoding from **Encoding Used for SQL Scripts** box.



## How to Generate DDL Script

This topic describes full configuration of a DDL script generation. In most cases, you will not need to configure all of the available settings.

- **Selecting Items for Generation**
- **Detailed Settings**
- **Referential Integrity**
- **Select List**
- **Script Preview**

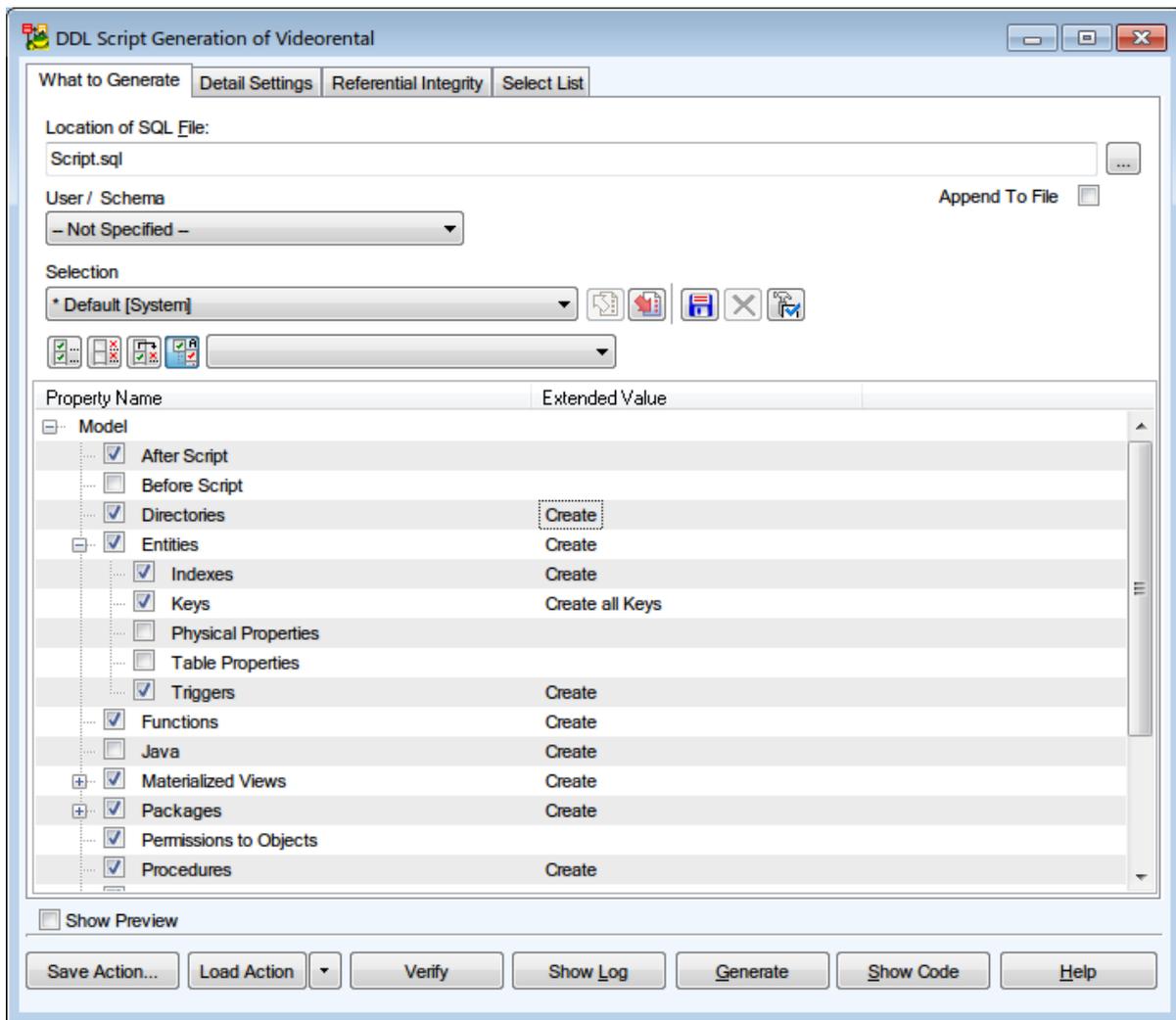
**i** Note: An **Oracle 10g** model is used in this topic. The dialogs and options available may vary depending on your model database platform and version.

Start by opening DDL Script Generation dialog using any of these methods:

- Click the  button on **Model Toolbar**
- Go to **Model Menu | Generate DDL Script | Run**
- Use shortcut - **F9**

## Selecting Items for Generation

This essential part of script generation, you need to select items which should be generated.

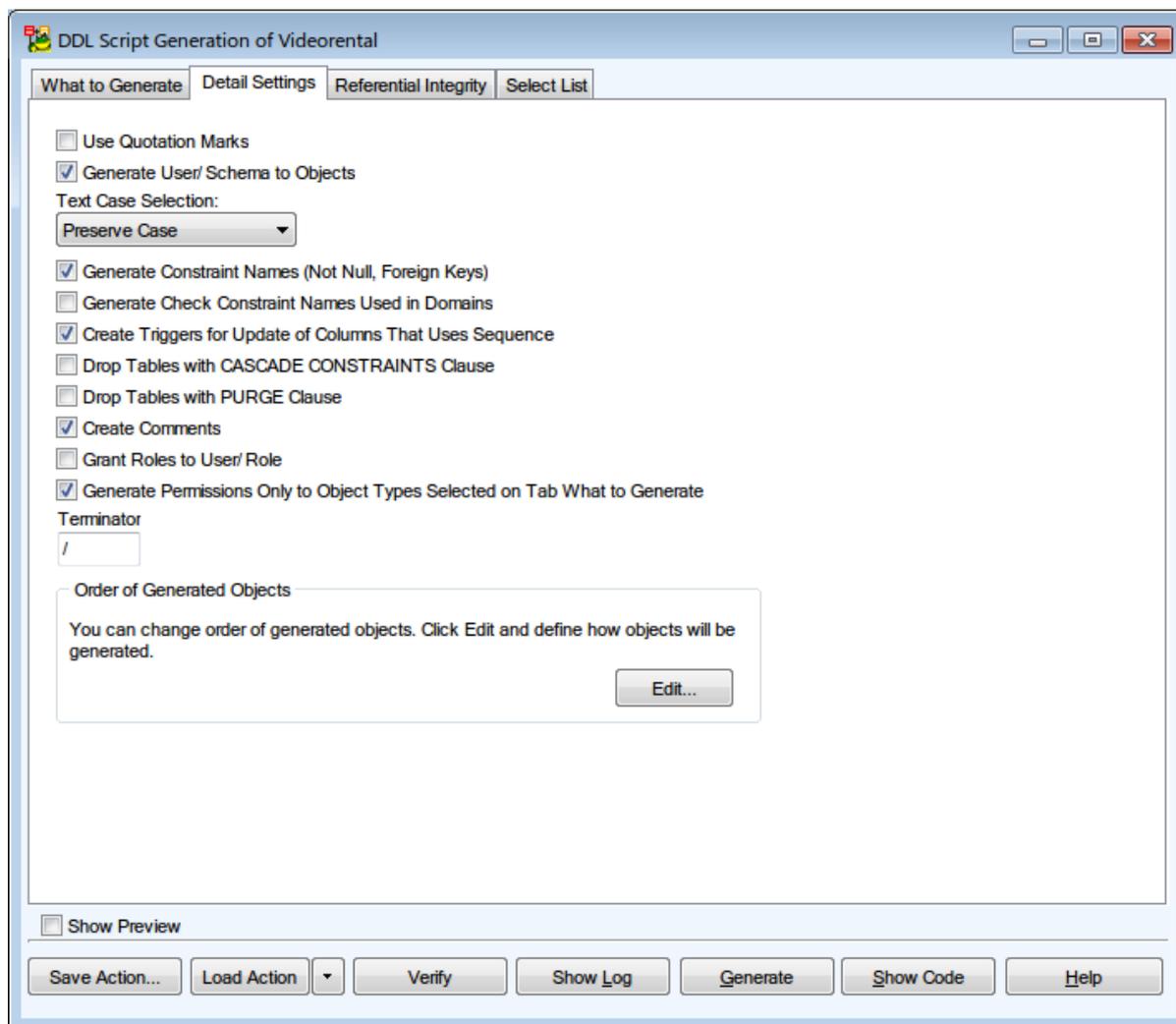


Option	Description
Location of SQL File	Select the location where the generated script will be saved. You can also select an already existing script and check <b>Append to File</b> checkbox to append the generated code to the original file.
User/Schema	Choose a user/schema from this box to only generate all of its assigned items.
Selection	You can save/load your custom selections. To do that, simply select the objects you want to be generated from the Objects Grid and then click  button to save your selection. If you will generate scripts in the future, you can load your selection  or even make it default  .
Bulk Selection	This section contains several buttons that help you select multiple objects quickly. You can:

Option	Description
Objects Grid	<ul style="list-style-type: none"> <li>• Select All</li> <li>• Deselect All</li> <li>• Invert Selection</li> <li>• Auto Check (if enabled, automatically checks sub-items when the parent item is checked)</li> </ul> <p>There is also a combobox which can be used to assign <b>Extended Value</b> to all items in <b>Objects Grid</b> at once.</p> <p>In this grid you select objects you want to generate by checking them in the <b>Property Name</b> column. And in the <b>Extended Value</b> column you can choose the SQL statement which should be used for the specific item in the final script (e.g. create, replace, drop).</p>

## Detailed Settings

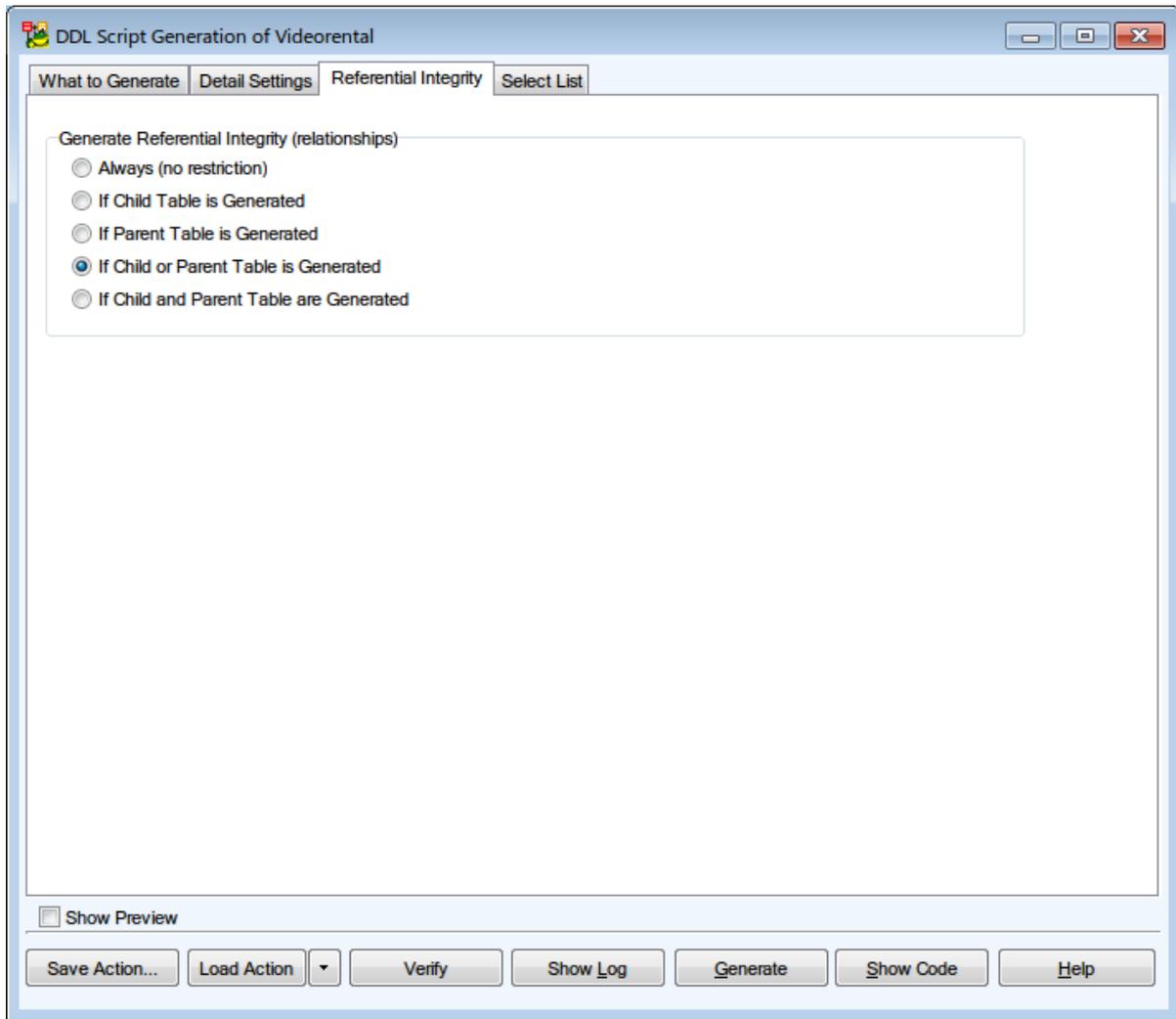
Specific and database related settings can be found on this tab. Usually it is not necessary to change them in any way.



## Referential Integrity

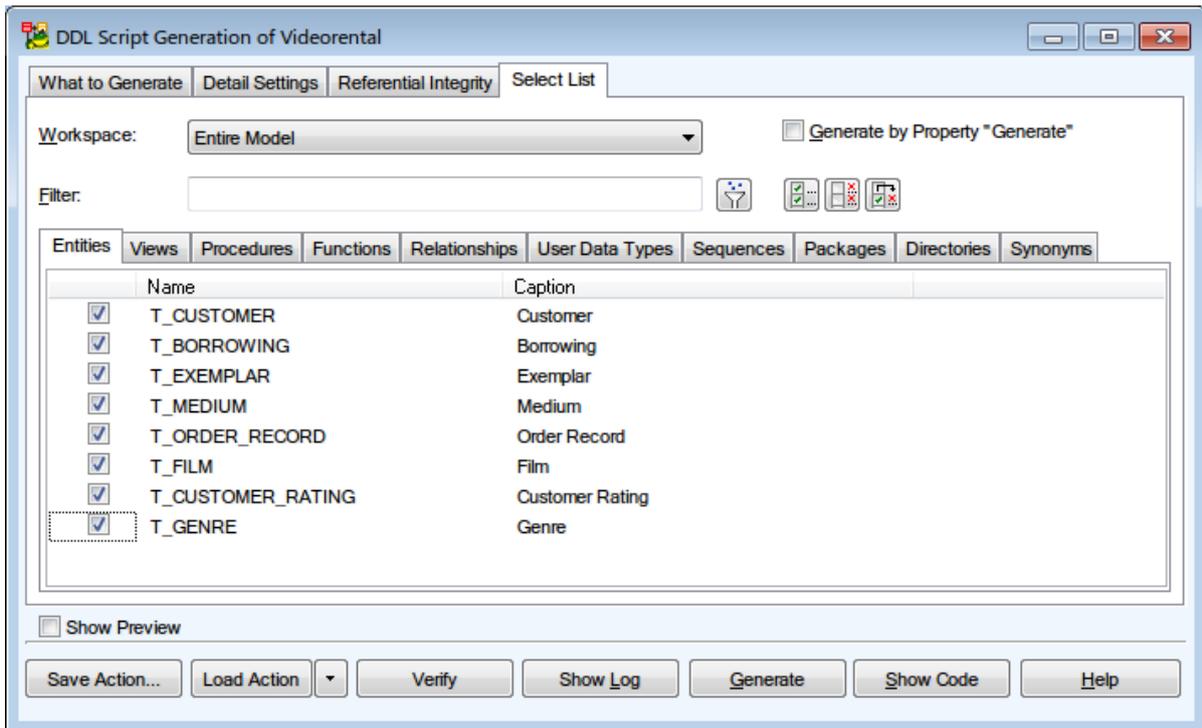
The main selection of objects for generation takes place in **Select List**. Referential integrity further specifies which relations are generated. It has the following options:

- Always - relationships are always generated if they are included in **Select List**
- If Child Table is Generated - relationship is generated only when its child table is included in **Select List**
- If Parent Table is Generated - relationship is generated only when its parent table is included in **Select List**
- If Child or Parent Table is Generated - relationship is generated only when child, parent or both are included in **Select List**
- If Child and Parent Tables are Generated - relationship is generated only when both child and parent tables are included in **Select List**



## Select List

If you want to generate only a set of specific model objects, you can do exactly that on **Select List** tab. By default, all objects that have **Generate** checkbox checked (e.g. **Entity Properties**) will be generated (unless you edited some other setting). If you uncheck the **Generate by Property "Generate"**, you are able to select any objects you want.

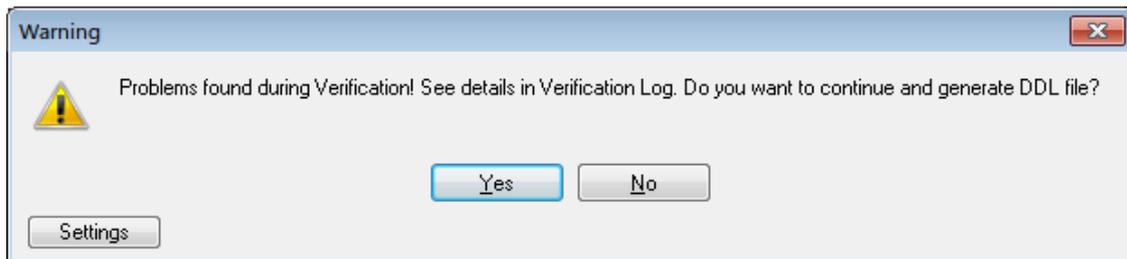


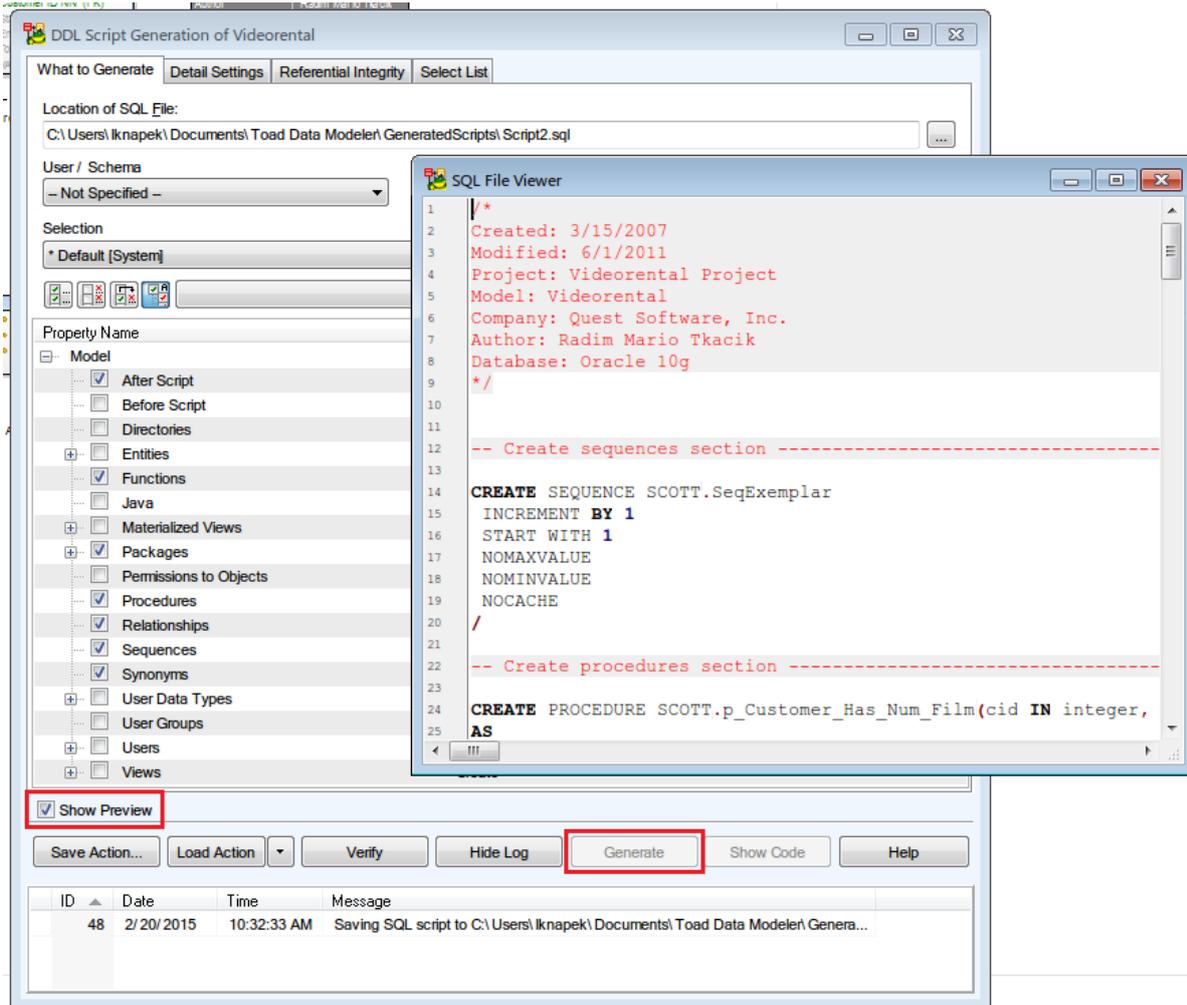
**i** Note: This tab is linked to the **What to Generate** tab | **Objects Grid**. If you uncheck an object type there, it you won't be able to select any of its objects in **Select List** tab.

## Script Preview

You can preview the generated SQL script at any time. To do that, check the **Show Preview** checkbox on **What to Generate** tab. Then click the **Generate** button. Toad Data Modeler will offer you a preview of the script before you decide to save it. If you wouldn't have checked the **Show Preview** checkbox, the script would be saved first before you could preview it.

**i** Note: Models are automatically verified before generating DDL script. To learn more, see [Model Verification](#).





**i** TIP: If you plan to do this action again in the future, you might want to click the **Save Action** button to save this **Action Definition**. See [Model Actions](#) for more information.

## Example of Generated Script

### Videorental Model

```
Oracle 10g database/*
Created: 15.3.2007
Modified: 6.12.2013
Project: Videorental Project
Model: Videorental
Company: Quest Software Inc.
Author: Jan Novak
Database: Oracle 10g
*/
```

```
-- Create user data types section -----
-----
```

```

CREATE TYPE SCOTT.Phone_List_TypeAS VARRAY(5) OF VARCHAR2(25)/
CREATE TYPE SCOTT.Cust_Address_TypeAS OBJECT
( street_address      VARCHAR2(40)
, postal_codeVARCHAR2(10)
, cityVARCHAR2(30)
, state_province     VARCHAR2(10)
, country_idCHAR(2)
, phonePhone_List_Type
)/
CREATE TYPE SCOTT.Price_TypeAS OBJECT (price NUMBER(10,2),
MEMBER FUNCTION total_price_VAT (vat number) RETURN NUMBER)/
CREATE TYPE BODY SCOTT.Price_TypeISMEMBER FUNCTION total_price_VAT (vat
NUMBER)
RETURN NUMBER ISBEGINRETURN (price*((vat+100)/100));
END;
END;/
-- Create sequences section -----
-
CREATE SEQUENCE SCOTT.SeqExemplarINCREMENT BY 1START WITH
1NOMAXVALUENOMINVALUENOCACHE/
-- Create tables section -----
-- Table SCOTT.T_CUSTOMERCREATE TABLE SCOTT.T_CUSTOMER(
customer_id Integer NOT NULL,
name Varchar2(20 CHAR),
address SCOTT.Cust_Address_Type
)/
-- Create indexes for table SCOTT.T_CUSTOMERCREATE INDEX i_name ON SCOTT.T_
CUSTOMER (name)/
-- Add keys for table SCOTT.T_CUSTOMERALTER TABLE SCOTT.T_CUSTOMER ADD
CONSTRAINT pk_T_CUSTOMER PRIMARY KEY (customer_id)/
-- Table and Columns comments sectionCOMMENT ON COLUMN SCOTT.T_CUSTOMER.name
IS 'Name column can contain First and Middle name. Surname must be in different
column.'/
-- Table SCOTT.T_GENRECREATE TABLE SCOTT.T_GENRE(
genre_id Integer NOT NULL,
name Varchar2(20 CHAR)
)/
-- Add keys for table SCOTT.T_GENREALTER TABLE SCOTT.T_GENRE ADD CONSTRAINT
pk_T_GENRE PRIMARY KEY (genre_id)/
-- Table SCOTT.T_MEDIUMCREATE TABLE SCOTT.T_MEDIUM(
medium_id Integer NOT NULL,
medium_type Varchar2(20 BYTE)
)/
-- Add keys for table SCOTT.T_MEDIUMALTER TABLE SCOTT.T_MEDIUM ADD
CONSTRAINT pk_T_MEDIUM PRIMARY KEY (medium_id)/
-- Table SCOTT.T_EXEMPLARCREATE TABLE SCOTT.T_EXEMPLAR(
exemplar_id Integer NOT NULL,
film_id Integer NOT NULL,
medium_id Integer NOT NULL,
price_per_day Integer
)/

```

```

-- Add keys for table SCOTT.T_EXEMPLARALTER TABLE SCOTT.T_EXEMPLAR ADD
CONSTRAINT pk_T_EXEMPLAR PRIMARY KEY (exemplar_id)/
ALTER TABLE SCOTT.T_EXEMPLAR ADD CONSTRAINT ak_exemplar_ppd UNIQUE
(exemplar_id,price_per_day)/
-- Table SCOTT.T_FILMCREATE TABLE SCOTT.T_FILM(
film_id Integer NOT NULL,
title Varchar2(50 CHAR) NOT NULL,
director Varchar2(30 CHAR) NOT NULL,
production_company Varchar2(50 CHAR),
genre_id Integer NOT NULL,
min_age Integer,
film_ID_episodes Integer
)/
-- Add keys for table SCOTT.T_FILMALTER TABLE SCOTT.T_FILM ADD CONSTRAINT
pk_T_FILM PRIMARY KEY (film_id)/
ALTER TABLE SCOTT.T_FILM ADD CONSTRAINT ak_title_director UNIQUE
(title,director)/
-- Table and Columns comments sectionCOMMENT ON COLUMN SCOTT.T_
FILM.production_company IS 'Company name must contain also information about company
type - LTD, Inc. and so on.'/
-- Table SCOTT.T_BORROWINGCREATE TABLE SCOTT.T_BORROWING(
exemplar_id Integer NOT NULL,
customer_id Integer NOT NULL,
start_date Date DEFAULT sysdate,
end_date Date,
total_price SCOTT.Price_Type,
VAT Number(4,2) DEFAULT 19,
CONSTRAINT check_end_after_start CHECK ((end_date>start_date) or (end_date
is null))
)/
-- Add keys for table SCOTT.T_BORROWINGALTER TABLE SCOTT.T_BORROWING ADD
CONSTRAINT pk_T_BORROWING PRIMARY KEY (exemplar_id)/
-- Create triggers for table SCOTT.T_BORROWINGCREATE TRIGGER SCOTT.tri_
BORROWINGBEFORE INSERTON SCOTT.T_BORROWINGFOR EACH ROWdeclare price number(10,2);
beginselect price_per_dayinto pricefrom T_EXEMPLARwhere T_EXEMPLAR.exemplar_
id = :new.exemplar_id;
:new.total_price := Price_Type((:new.end_date-:new.start_date)*price);
end;/
CREATE TRIGGER SCOTT.tru_BORROWINGBEFORE UPDATEON SCOTT.T_BORROWINGFOR EACH
ROWdeclareprice number(10,2);
total_price_old number(10,2);
end_d date;
start_d date;
beginselect e.price_per_dayinto pricefrom T_EXEMPLAR ewhere e.exemplar_id =
:new.exemplar_id;
total_price_old := :new.total_price.price;
end_d := :new.end_date;
start_d := :new.start_date;
if (total_price_old is null) thentotal_price_old := 0;
end if;
if ((end_d-start_d)*price != total_price_old) then

```

```

        :new.total_price := Price_Type((end_d-start_d)*price);
    end if;
end;/
-- Table and Columns comments sectionCOMMENT ON TABLE SCOTT.T_BORROWING IS
'Borrowed items'//
-- Table SCOTT.T_ORDER_RECORDCREATE TABLE SCOTT.T_ORDER_RECORD(
customer_id Integer NOT NULL,
film_id Integer NOT NULL,
order_date Date
)/
-- Add keys for table SCOTT.T_ORDER_RECORDALTER TABLE SCOTT.T_ORDER_RECORD
ADD CONSTRAINT pk_T_ORDER_RECORD PRIMARY KEY (customer_id,film_id)/
-- Table and Columns comments sectionCOMMENT ON TABLE SCOTT.T_ORDER_RECORD
IS 'All records are stored in list of records. It will be possible to book a
movie.'//
-- Table SCOTT.T_CUSTOMER_RATINGCREATE TABLE SCOTT.T_CUSTOMER_RATING(
title Varchar2(50 CHAR) NOT NULL,
director Varchar2(30 CHAR) NOT NULL,
rating Integer DEFAULT 3CONSTRAINT check_rating CHECK (rating > 0)
)/
-- Add keys for table SCOTT.T_CUSTOMER_RATINGALTER TABLE SCOTT.T_CUSTOMER_
RATING ADD CONSTRAINT pk_T_CUSTOMER_RATING PRIMARY KEY (title,director)/
-- Table and Columns comments sectionCOMMENT ON TABLE SCOTT.T_CUSTOMER_
RATING IS 'Movie ratings (by customers)'/
-- Create procedures section -----
--
CREATE PROCEDURE SCOTT.p_Customer_Has_Num_Film(cid IN integer, num OUT
integer)
ASBEGINSELECT count(*)
INTO numFROM T_CUSTOMER c, T_BORROWING b, T_EXEMPLAR e, T_FILM fWHERE
c.customer_id=cidand c.customer_id=b.customer_idand b.exemplar_id=e.exemplar_idand
e.film_id=f.film_id;
END;/
-- Create views section -----
CREATE VIEW SCOTT.v_Customer_Has_Film ASSELECT DISTINCT c.name,
c.address.city AS city, f.title, f.directorFROM SCOTT.T_CUSTOMER c, SCOTT.T_
BORROWING b, SCOTT.T_EXEMPLAR e, SCOTT.T_FILM fWHERE c.customer_id=b.customer_idand
b.exemplar_id=e.exemplar_idand e.film_id=f.film_id/
-- Create functions section -----
-
CREATE FUNCTION SCOTT.f_Customer_Has_Num_Film(cid IN integer)
RETURN integerISSol integer;
BEGINp_Customer_Has_Num_Film(cid,sol);
RETURN (sol);
END;/
-- Trigger for sequence SCOTT.SeqExemplar for column exemplar_id in table
SCOTT.T_EXEMPLAR -----
CREATE OR REPLACE TRIGGER SCOTT.ts_T_EXEMPLAR_SeqExemplar BEFORE INSERTON
SCOTT.T_EXEMPLAR FOR EACH ROWBEGINSELECT SCOTT.SeqExemplar.nextval INTO
:new.exemplar_id FROM DUAL;
END;/

```

```

CREATE OR REPLACE TRIGGER SCOTT.tsu_T_EXEMPLAR_SeqExemplar AFTER UPDATE OF
exemplar_id ON SCOTT.T_EXEMPLAR FOR EACH ROW BEGIN RAISE_APPLICATION_ERROR(-
20010, 'Cannot update column exemplar_id in table SCOTT.T_EXEMPLAR as it uses
sequence. ');
END; /
-- Create relationships section -----
-----
ALTER TABLE SCOTT.T_BORROWING ADD CONSTRAINT makes FOREIGN KEY (customer_id)
REFERENCES SCOTT.T_CUSTOMER (customer_id) /
ALTER TABLE SCOTT.T_BORROWING ADD CONSTRAINT is_related_to FOREIGN KEY
(exemplar_id) REFERENCES SCOTT.T_EXEMPLAR (exemplar_id) /
ALTER TABLE SCOTT.T_ORDER_RECORD ADD CONSTRAINT places FOREIGN KEY
(customer_id) REFERENCES SCOTT.T_CUSTOMER (customer_id) /
ALTER TABLE SCOTT.T_EXEMPLAR ADD CONSTRAINT is_available_on FOREIGN KEY
(medium_id) REFERENCES SCOTT.T_MEDIUM (medium_id) /
ALTER TABLE SCOTT.T_ORDER_RECORD ADD CONSTRAINT is_required_by FOREIGN KEY
(film_id) REFERENCES SCOTT.T_FILM (film_id) /
ALTER TABLE SCOTT.T_FILM ADD CONSTRAINT is_of FOREIGN KEY (genre_id)
REFERENCES SCOTT.T_GENRE (genre_id) /
ALTER TABLE SCOTT.T_EXEMPLAR ADD CONSTRAINT has FOREIGN KEY (film_id)
REFERENCES SCOTT.T_FILM (film_id) /
ALTER TABLE SCOTT.T_CUSTOMER_RATING ADD CONSTRAINT is_rated FOREIGN KEY
(title, director) REFERENCES SCOTT.T_FILM (title, director) /
ALTER TABLE SCOTT.T_FILM ADD CONSTRAINT has_more_episodes FOREIGN KEY (film_
ID_episodes) REFERENCES SCOTT.T_FILM (film_id) /
-- Grant permissions section -----
--
insert into T_GENRE values (1, 'crime') /
insert into T_GENRE values (2, 'western') /
insert into T_GENRE values (3, 'drama') /
insert into T_GENRE values (4, 'biography') /
insert into T_GENRE values (5, 'comedy') /
insert into T_MEDIUM values (1, 'CD - DivX') /
insert into T_MEDIUM values (2, 'CD - VideoCD') /
insert into T_MEDIUM values (3, 'DVD') /
insert into T_MEDIUM values (4, 'Videotape') /
insert into T_FILM values (1, 'The Shawshank Redemption', 'Frank
Darabont', 'Castle Rock Entertainment', 1, 15, null) /
insert into T_FILM values (2, 'The Godfather', 'Francis Ford
Coppola', 'Paramount Pictures', 1, 15, null) /
insert into T_FILM values (3, 'The Godfather: Part II', 'Francis Ford
Coppola', 'Paramount Pictures', 1, 15, 2) /
insert into T_FILM values (4, 'The Good, the Bad and the Ugly', 'Sergio
Leone', 'PEA', 2, 15, null) /
insert into T_FILM values (5, 'Pulp Fiction', 'Quentin Tarantino', 'Miramax
Films', 1, 18, null) /
insert into T_FILM values (6, '12 Angry Men', 'Sidney Lumet', 'Orion-Nova
Productions', 3, 15, null) /
insert into T_FILM values (7, 'Schindler''s List', 'Steven
Spielberg', 'Universal Pictures', 4, 15, null) /
insert into T_FILM values (8, 'One Flew Over the Cuckoo''s Nest', 'Milos

```

```

Forman','Fantasy Films',3,15,null)/
    insert into T_EXEMPLAR values (1,1,3,3)/
    insert into T_EXEMPLAR values (2,1,3,3)/
    insert into T_EXEMPLAR values (3,1,3,3)/
    insert into T_EXEMPLAR values (4,1,4,2)/
    insert into T_EXEMPLAR values (5,1,4,2)/
    insert into T_EXEMPLAR values (6,2,3,3)/
    insert into T_EXEMPLAR values (7,2,4,2)/
    insert into T_EXEMPLAR values (8,3,3,3)/
    insert into T_EXEMPLAR values (9,4,4,2)/
    insert into T_EXEMPLAR values (10,5,3,3)/
    insert into T_EXEMPLAR values (11,6,4,2)/
    insert into T_EXEMPLAR values (12,6,4,2)/
    insert into T_EXEMPLAR values (13,7,4,2)/
    insert into T_CUSTOMER values (1,'Audrey',Cust_Address_Type('92A Campton
Avenue','60021','Fox River Grove',null,'us',null))/
    insert into T_CUSTOMER values (2,'Simon',Cust_Address_Type('12 Hillpeak
Street','60006','Arlington Heights',null,'ca',null))/
    insert into T_CUSTOMER values (3,'Dave',Cust_Address_Type('31D South
Avenue','60001','Alden',null,'us',Phone_List_Type('541 123 456')))/
    insert into T_CUSTOMER values (4,'Chris',Cust_Address_Type('5 Quiet
Street','60061','Vernon Hills',null,'us',Phone_List_Type('541 123 456','596 815
641')))/
    insert into T_CUSTOMER values (5,'Elen',Cust_Address_Type('47 My
Avenue','60083','Beach Park','Illinois','us',null))/
    insert into T_BORROWING (exemplar_id,customer_id,start_date,end_date) values
(11,1,to_date('01-01-2010','DD-MM-YYYY'),to_date('03-01-2010','DD-MM-YYYY'))/
    insert into T_BORROWING (exemplar_id,customer_id,start_date,end_date) values
(12,2,to_date('02-01-2010','DD-MM-YYYY'),to_date('10-01-2010','DD-MM-YYYY'))/
    insert into T_BORROWING (exemplar_id,customer_id,start_date) values (1,2,to_
date('06-01-2010','DD-MM-YYYY'))/
    insert into T_BORROWING (exemplar_id,customer_id,start_date) values (6,4,to_
date('07-01-2010','DD-MM-YYYY'))/
    insert into T_BORROWING (exemplar_id,customer_id,start_date) values (7,3,to_
date('07-01-2010','DD-MM-YYYY'))/
    insert into T_BORROWING (exemplar_id,customer_id) values (9,3)/
    insert into T_ORDER_RECORD values (1,2,to_date('01-01-2010','DD-MM-YYYY'))/
    insert into T_ORDER_RECORD values (1,1,to_date('01-01-2010','DD-MM-YYYY'))/
    insert into T_ORDER_RECORD values (2,2,to_date('03-01-2010','DD-MM-YYYY'))/
    insert into T_ORDER_RECORD values (3,3,to_date('06-01-2010','DD-MM-YYYY'))/

```

## Autolayout

There are three autolayout methods available in Toad Data Modeler.



1. Top to Bottom
2. Left to Right

### 3. Alphabetic

#### **To re-arrange objects on your Workspace automatically**

Click any of the autolayout icons on the **Layout Toolbar** (these options are also available in **Layout Menu**).  
Autolayout is especially handy when you:

- Reverse engineer a database with a large number of items
- Add a large number of model objects to a new Workspace

#### **To configure properties of Autolayout**

Go to **Settings Menu | Options | Graphics | Autolayout**.

You can configure horizontal and vertical distance coefficients (the distance between objects) and Alphabetic Autolayout sorting property.

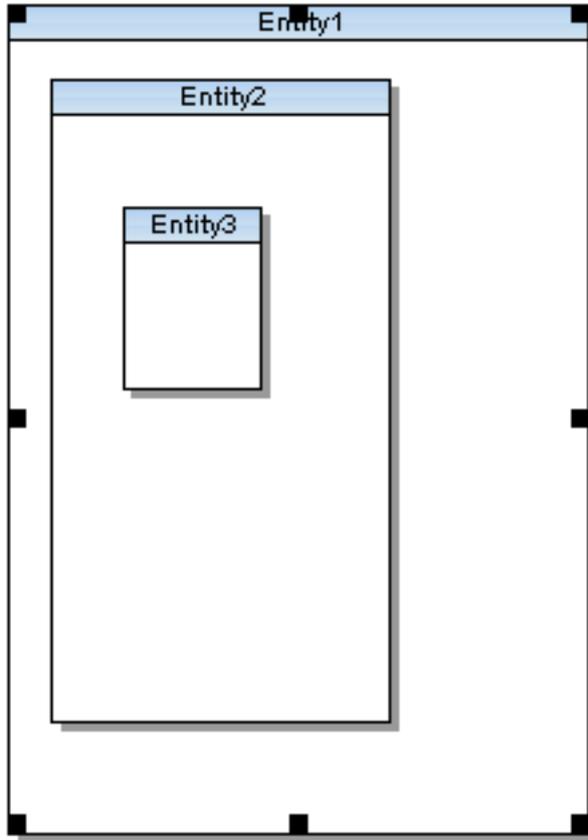


The screenshot shows a settings panel titled "Shapes". It contains three rows of configuration options:

- Horizontal Distance Coefficient**: A numeric input field with the value "70" and a small icon with up and down arrows.
- Vertical Distance Coefficient**: A numeric input field with the value "70" and a small icon with up and down arrows.
- Alphabetical Autolayout - Sort By**: A dropdown menu currently displaying "Name".

## Arrange Objects in Layers

Toad Data Modeler allows you to rearrange/order objects on the Workspace in different layers.



### **To re-arrange the selected object on the Workspace**

Right-click the object | **Arrange** and select any of the following options:

- **Bring to Front** – Brings the object on the top layer.
- **Bring Forward** – Brings the object one layer up.
- **Send Backward** – Sends the object one layer down.
- **Send to Back** – Sends the object to the lowest layer.

## **Z-Order Box**

### **To specify exact layer for an object on Workspace:**

1. Right-click the object and select **Arrange** | **Settings**.
2. In the **Object Format** dialog, change the **Z-Order** value. Object with greater Z-Order value are placed on top of objects with lower Z-Order value.

## **Arrange Relationship Lines**

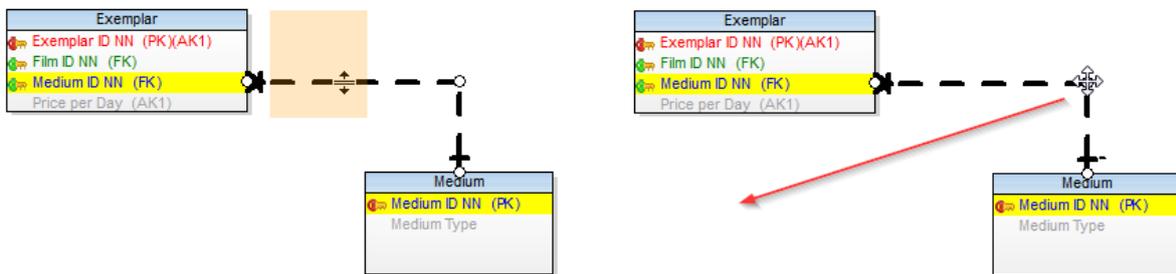
Relationship lines consist of:

- **Lines**
- **Anchor points** - End points directly connected to entity boxes. You can move them.
- **Handle points** - Other points that can be added to relationship lines via CTRL key. Handle points allow you to select a part of relationship line to move or delete.

## Move Lines, Anchor Points and Handle Points

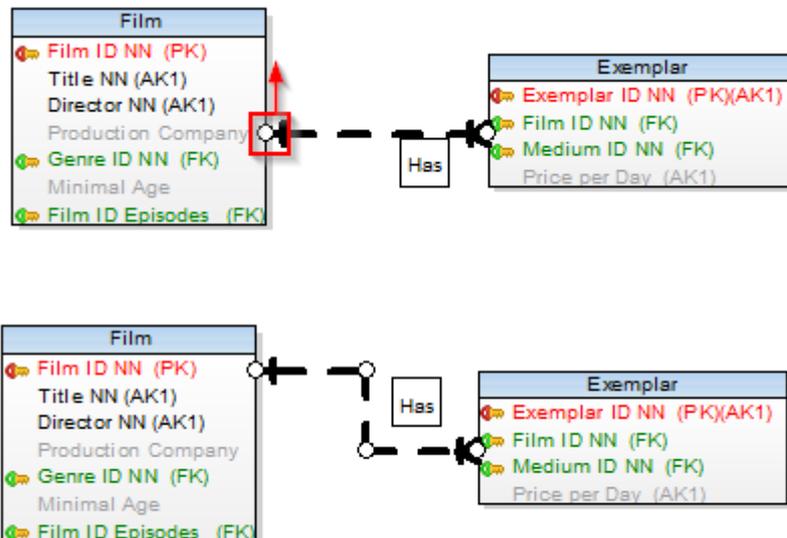
### To move Line or Handle point

1. Select a relationship line.
2. Place mouse cursor over a **part of the line** or over a **Handle point**.
3. Drag and drop to change the position of the line/handle point.



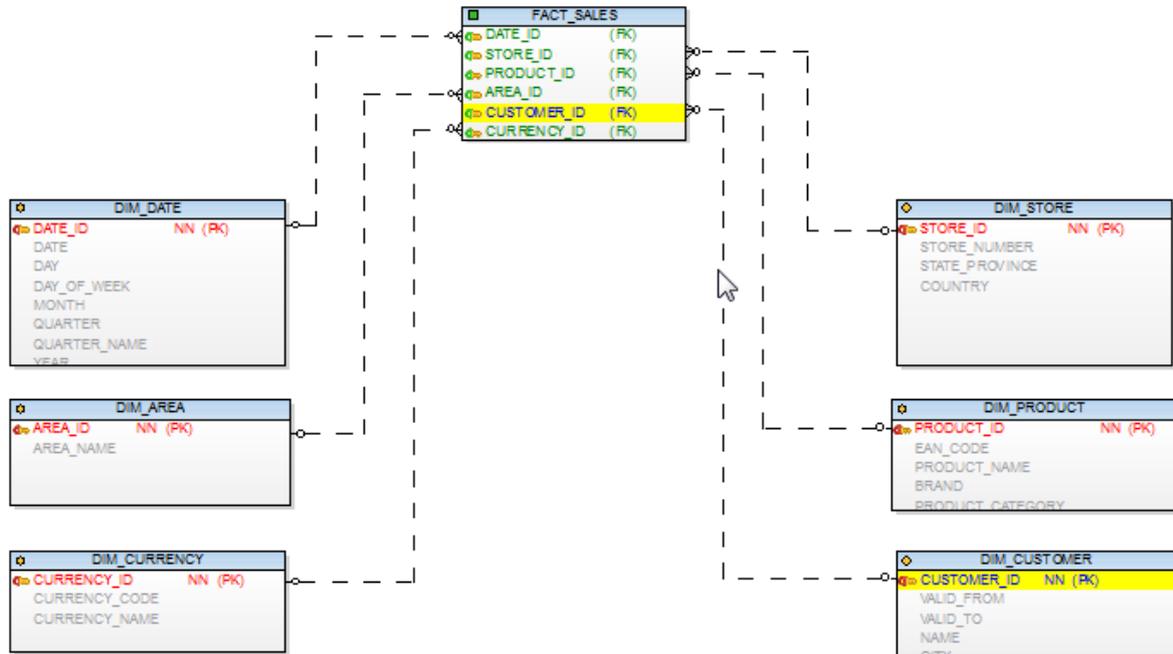
### To move single anchor point and break a line

1. Select a relationship line.
2. Place mouse cursor over an anchor point.
3. Use drag and drop technique to change position of the break point.



# Column to Column Alignment

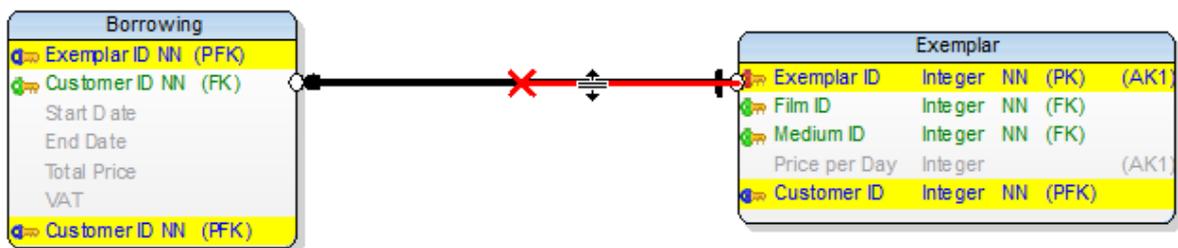
Lines can be rearranged manually to clearly express a link between columns in entities.



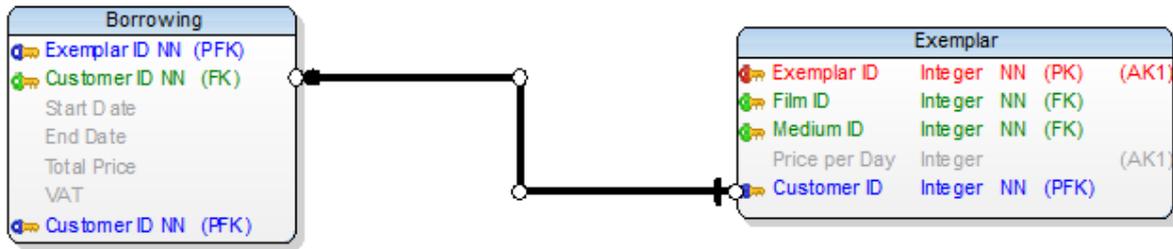
# Add Handle Points

**To break a horizontal line using one Handle point:**

1. Select a relationship line.
2. Press CTRL and click the line in any place. A red cross icon appears. Release CTRL key.
3. Move your mouse cursor left or right of the cross icon and move the highlighted part of the line up or down.

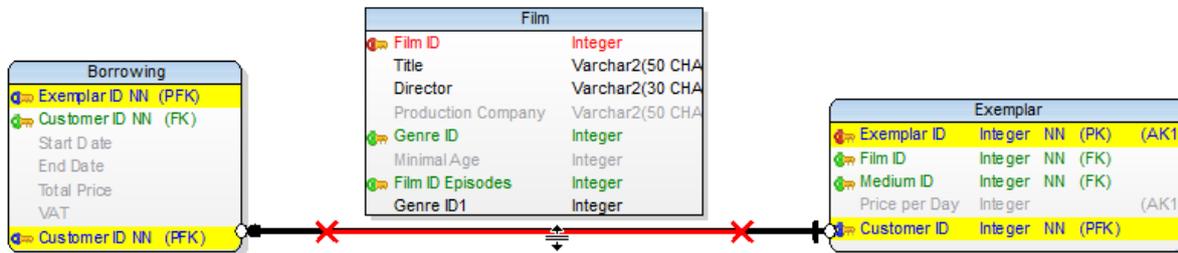


Result:



**To break horizontal line using two handle points:**

1. Select a relationship line.
2. Press CTRL and click the line. A red cross icon appears.
3. Click the line somewhere else. A second red cross appears.
4. Move the highlighted part between the two crosses up or down.



Result:

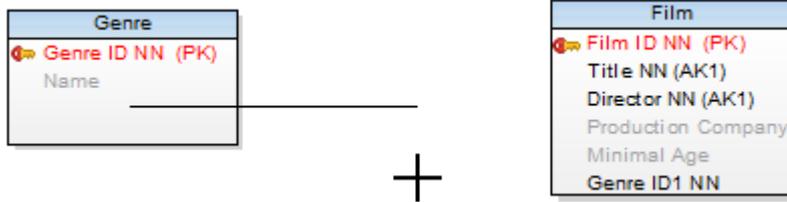


## Add Horizontal or Vertical Lines Only

**To create horizontally or vertically straight lines**

(Assuming it is possible to link two entities using only horizontal/vertical line)

1. Click the identifying or non-identifying relationship icon on **Model Objects Toolbar**.
2. Click parent entity and hold SHIFT key. As long as you hold it, you can only create vertically/horizontally straight line.
3. Move your mouse cursor over the desired position (child entity)
4. Click the target entity and release SHIFT key.



**i** TIP: Relationship lines between entities are created automatically once you click the parent and the child entity. The line created may not be ideal and you may have to edit it so it looks as you wish. But you can also create your own custom relationship line from scratch:

1. Click the identifying or non-identifying relationship icon on **Model Objects Toolbar**.
2. Click the parent entity and hold **SHIFT** key. As long as you hold it, you can only create vertically/horizontally straight line.
3. Lead the line to the child entity. Click on Workspace to create a break point. By using break points, you can create any path you want.
4. Finally, click the target entity and release **SHIFT** key.

## Unhide Line

### *To display relationship line hidden behind objects on Workspace*

Select the relationship and click  button on **Layout Toolbar** or right-click the line | **Line Style** | **Unhide Line**.

## Move, Hide, Find Relationship Names

### *To move relationship name*

Right-click the selected relationship in particular place and select **Move Caption Here** or **Move Caption to Parent** or **Move Caption to Child**.

### *To hide relationship names on the Workspace*

Click  **Hide Captions** button on **Layout Toolbar**.

**i** Note: You can also right-click the Workspace | **Workspace Format** | **General** tab | **Hide Line Captions**.

### *To find a relationship line by its caption on Workspace*

Click the relationship caption. The appropriate relationship line will be highlighted on the Workspace.

### *To find a relationship caption by its line on Workspace*

Click the relationship line, its caption will be highlighted in a frame.

# Format Relationship Lines

## ***To change format of all relationship lines on Workspace***

Right-click the WS and select **Workspace Format**.

## ***To change format of selected relationship(s)***

Right-click the relationship and select **Format**.

# Categories

Categories allow you to colorfully distinguish parts of your model. Categories can be assigned to **entities, views** and **relationships**.

---

### **Scenario**

You would like to graphically distinguish all entities related to Ordering process. Or maybe you want to mark all entities containing personal data of your employees.

*Solution:* You can simply create a category, select its color and assign it to the appropriate entities. All the entities will share the category color.

---

## ***To add a category to your model***

- Select **Model Menu | Categories** and click **Add** in the **Category List** dialog.

or

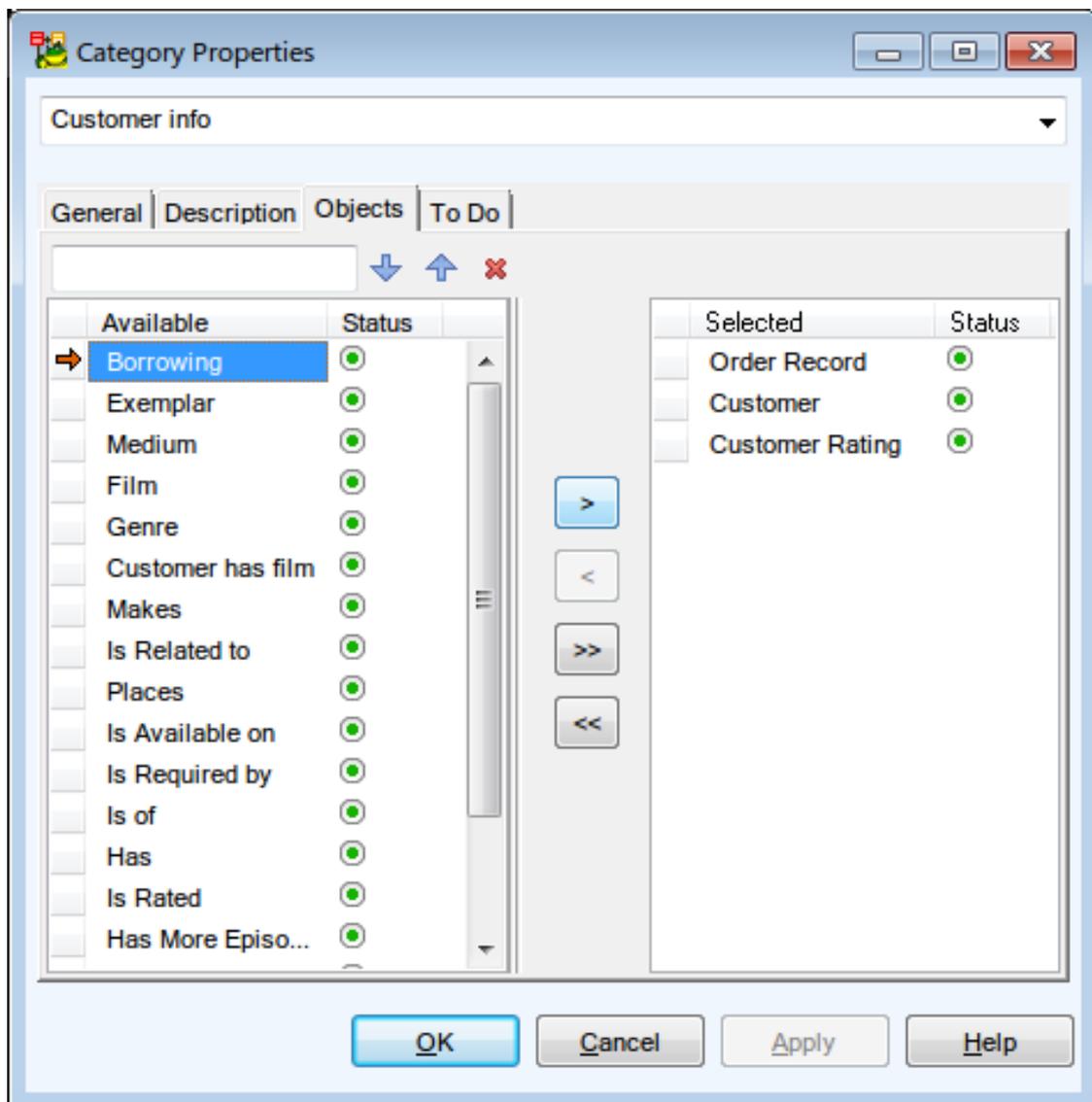
- **Right-click** the **Categories** folder in **Model Explorer** and select **Add Category**.,

or

- In **Entity/View/Relationship Properties** dialog, click the ... button in the **Category** section.

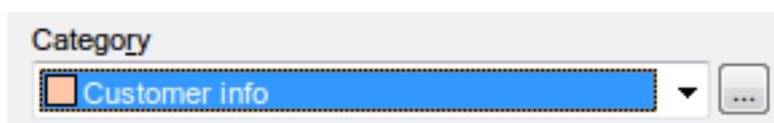
### To assign category to objects

- Open **Category Properties** (via **Model Menu** or **Model Explorer**) | **Switch to Objects** tab and move objects to right section.



or

- In **Entity/View/Relationship Properties**, choose the desired category in the **Category** section on **General** tab.





TIP:

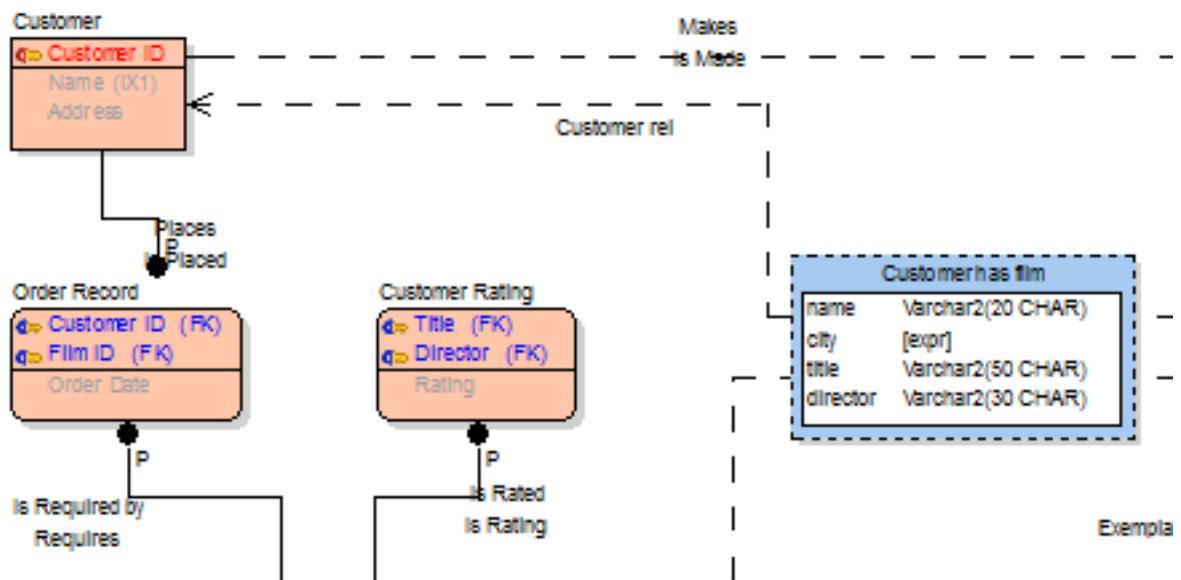
- If your model contains multiple categories, you can display captions of categories on Workspace. See [Caption of Categories](#) for more information.
- To use category colors in Model Explorer, right-click in **Model Explorer** | **Settings** | enable **Use Colors of Category to Draw**. Object names will be now match their category color.

## Change Notation

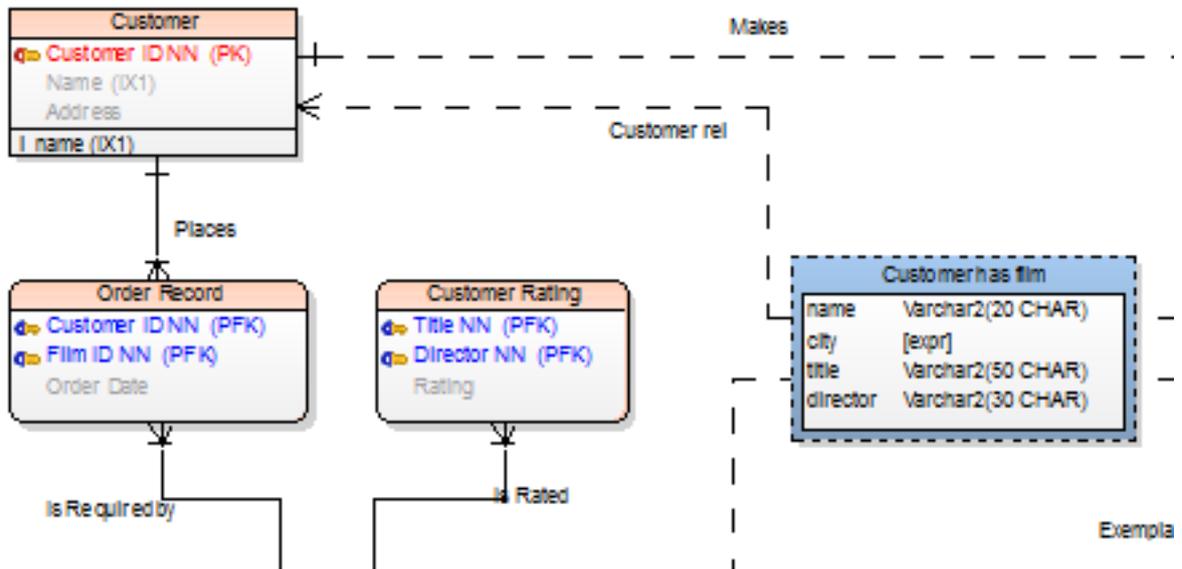
To change notation in your model

- Go to **View Menu** | **Notation** | choose **IE** or **IDEF1X**.

IDEF1X



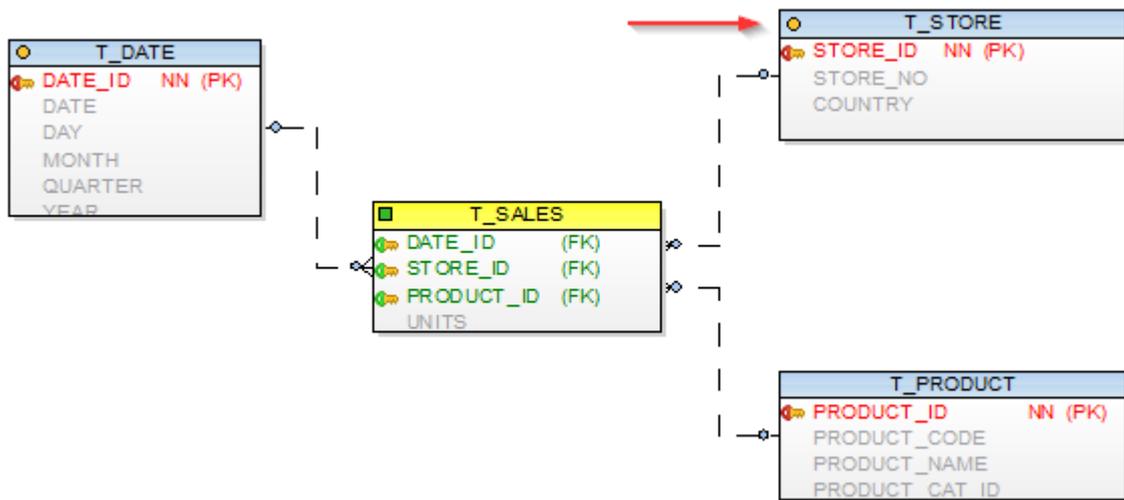
IE



## Data Warehouse Types

In physical model, you can define a Data Warehouse Type for your entities and graphically distinguish Fact and Dimension entities.

**Example:**

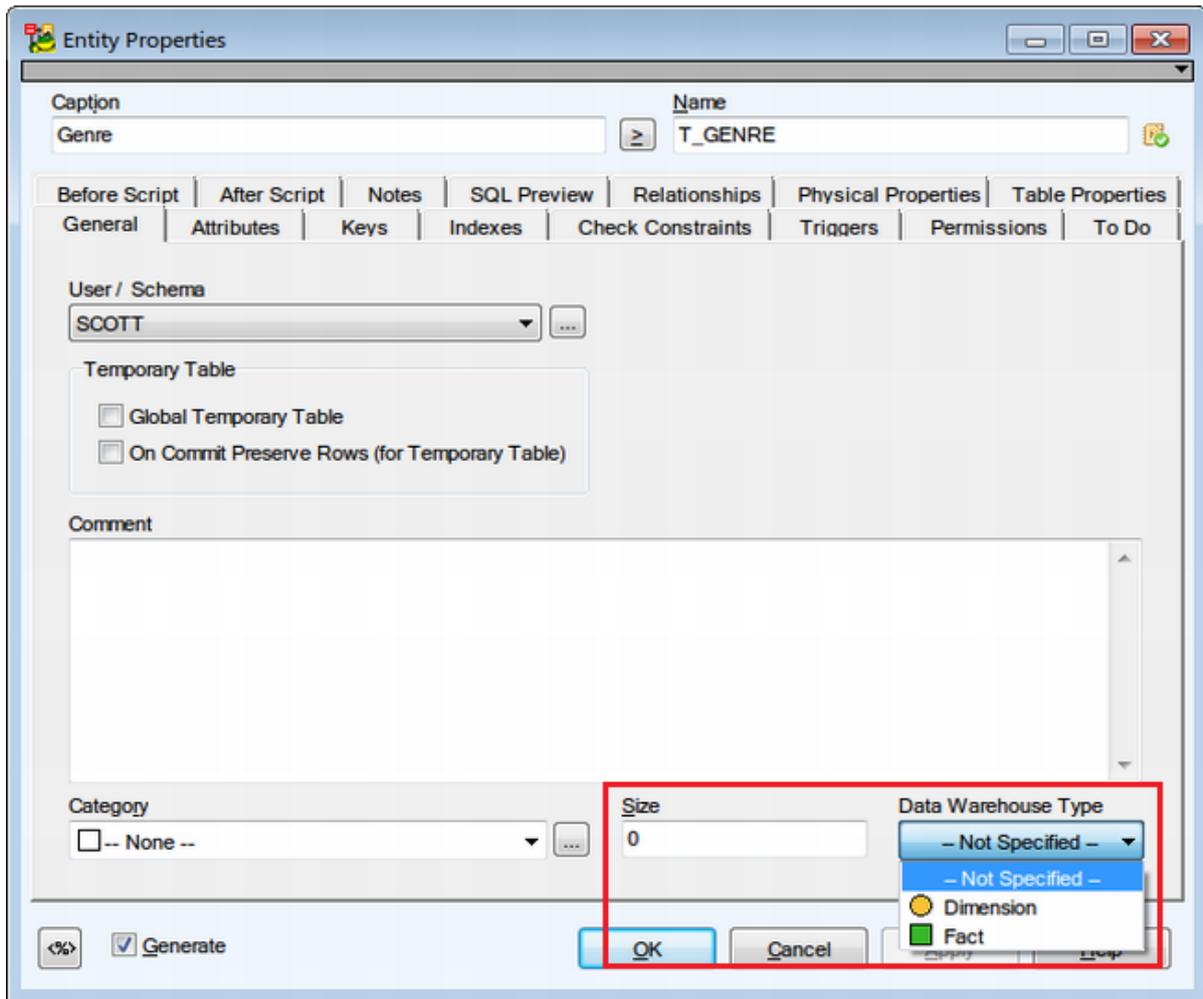


By default, the Data Warehouse options are not available and have to be activated in **Settings** menu.

### To display Data Warehouse Type options

Select **Settings Menu** | **Options** | **Physical Model** | **Entity tab** | **Form Settings** | check **Display Data Warehouse Type and Size**.

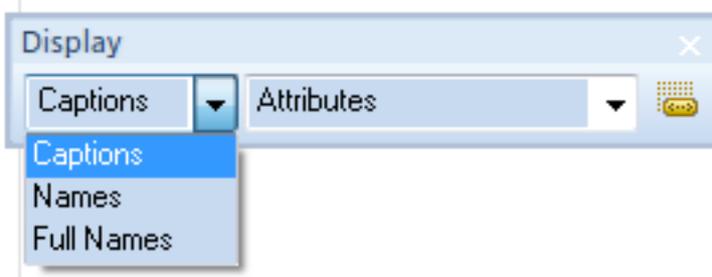
**Data Warehouse Type and Size** options then become available in **Entity Properties**.



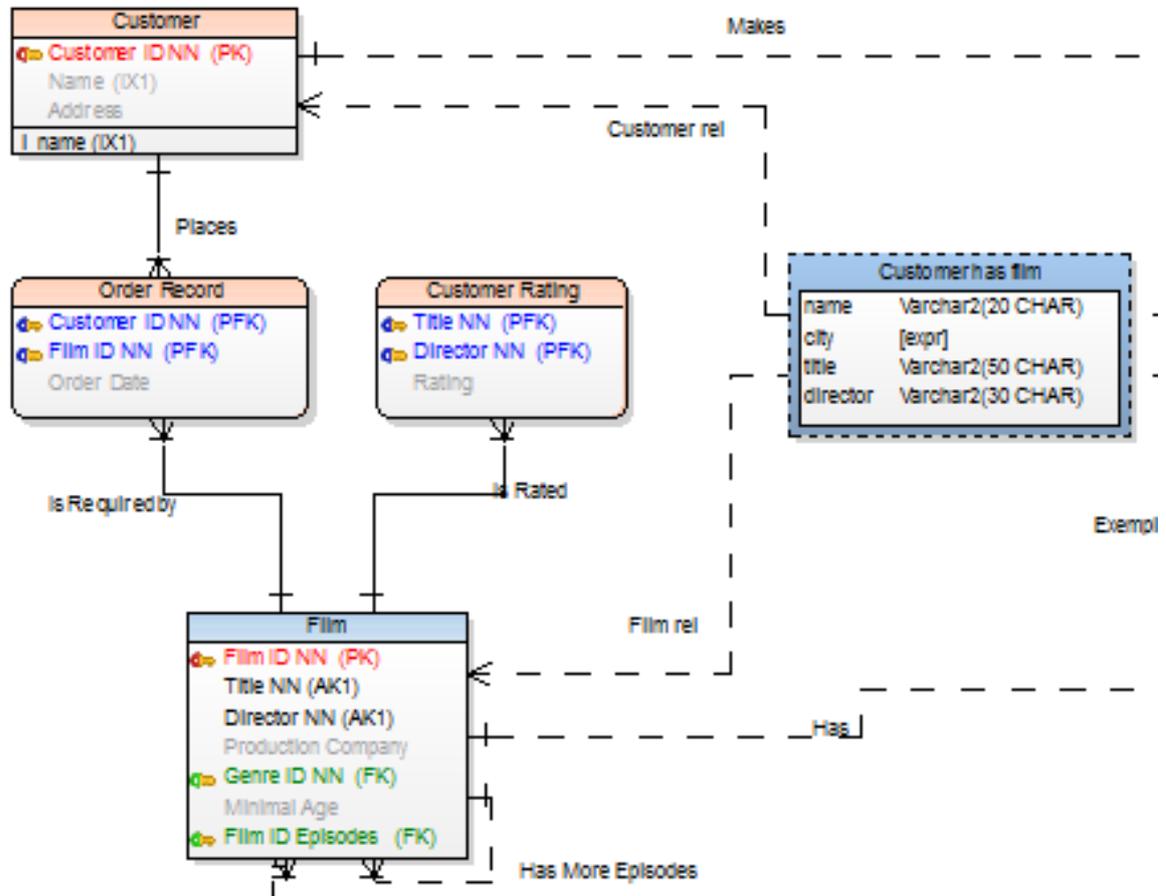
## Display Modes

In Toad Data Modeler, you can switch between **Logical (Captions)**, **Physical (Names)** and **Full Names** view of object names.

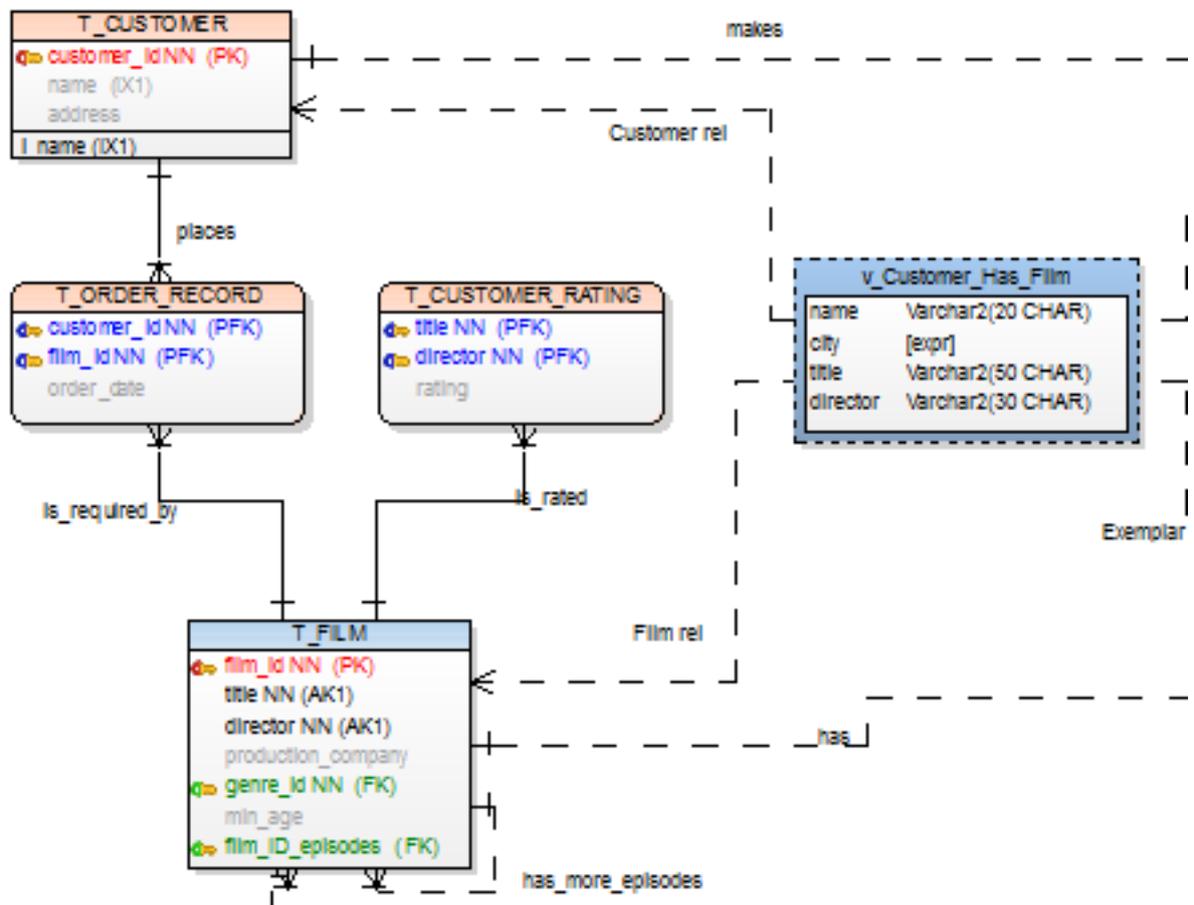
Select the display mode on **Display Toolbar** or in **View Menu | Display Mode**.



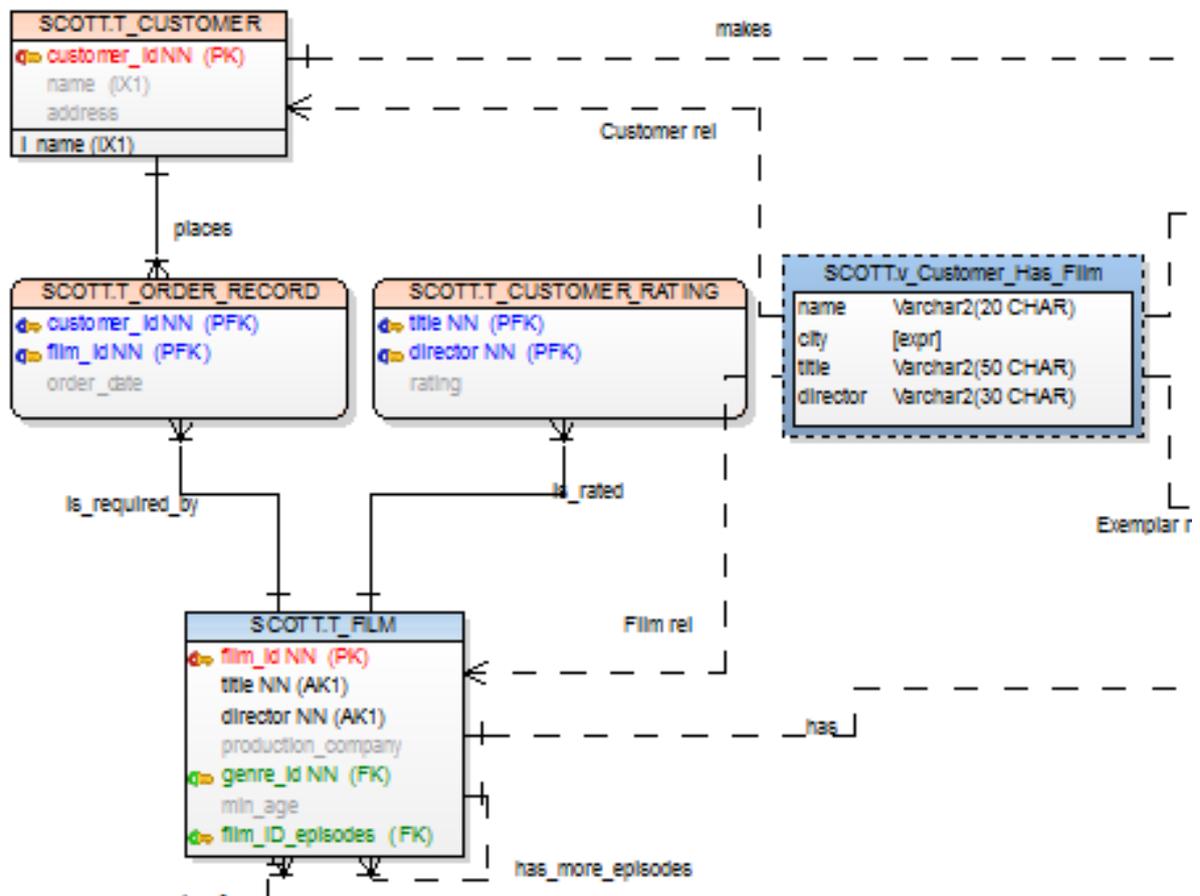
## Logical View (Captions)



## Physical View (Names)



## Full Names



## Display Level of Entities

Toad Data Modeler allows you to display entities on Workspaces in several levels:

### PER Model

- Entities
- Primary Keys
- PK and FK keys
- All Keys
- Attributes

### LER Model

- Entities
- Primary Identifiers
- Unique Identifiers

- Attributes
- Descriptions - Text written in the **Description** tab of entity will be displayed on the WS.

#### **To set the default display level for the selected Workspace**

Change the display level from the **Display Level** box on the toolbar (also **View | Display Level**).

#### **To set the default display level for new model (models that you will create)**

Select **Settings | Options | Physical/Logical Model | Entity** tab.

## Format Workspaces and Objects

You can set format for all Workspaces of your model, for each Workspace separately and also for particular objects.

#### **To set format for new models (models that you will create)**

1. Select **Settings | Options | Model section | Physical/Logical Model**.
2. Define options on tabs **Workspace**, **Shape**, **Note Line** and **Entity**.
3. Press CTRL+N to create a new model.

#### **To change format of objects in existing models**

Right-click the Workspace and select **Workspace Format**.

#### **To change format of a particular object**

Right-click the object on the Workspace and select **Format**.

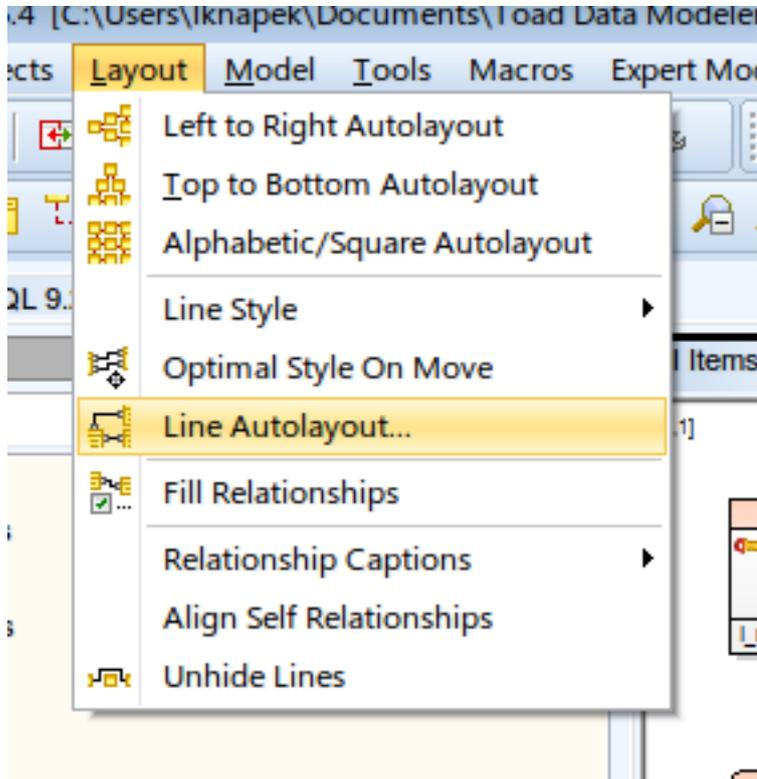
**i** | Note: If you need to preserve format of a particular object against changes of format of your WS, select the **Lock Format** option in the **Object Format** dialog | **General** tab.

## Line Autolayout

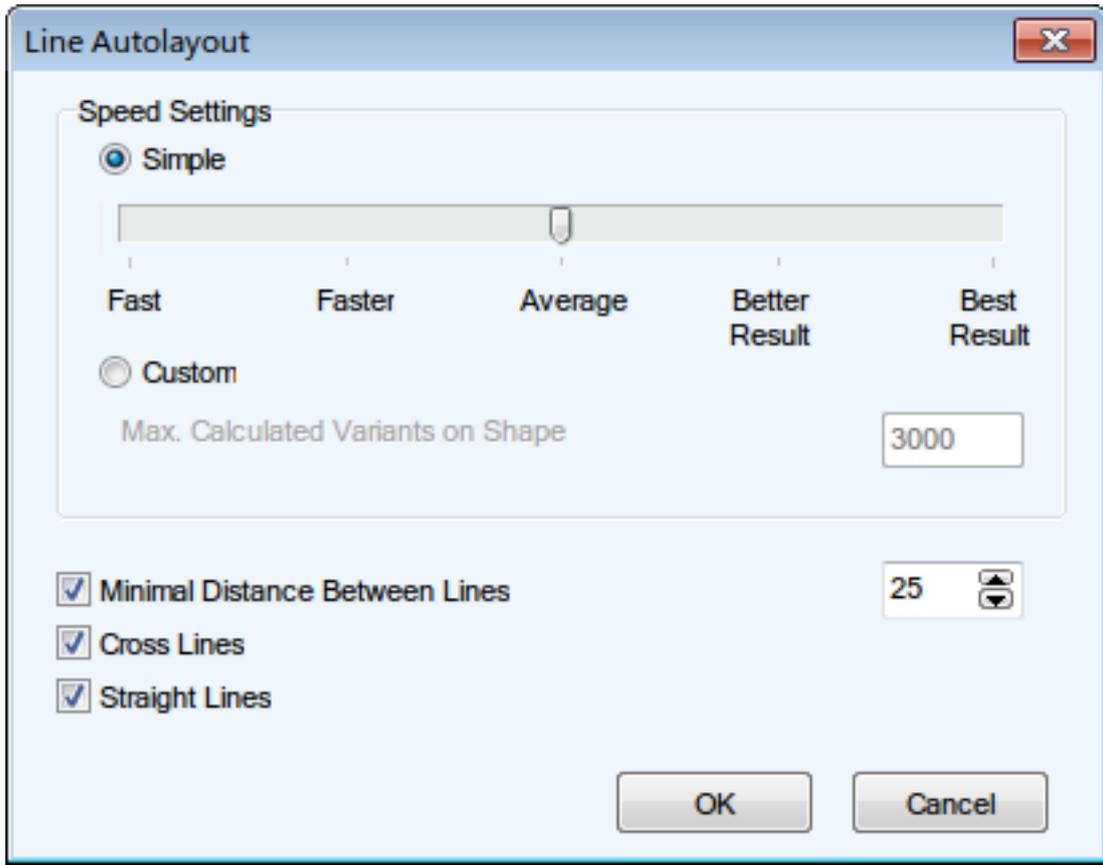
Toad Data Modeler provides you with an useful tool that is capable of automatically organizing your lines in your model diagram.

### How to use Line Autolayout

1. On workspace select lines which you want to organize (or don't select any to organize all lines).
2. Go to **Layout Menu | Line Autolayout** or click  on **Layout Toolbar**.



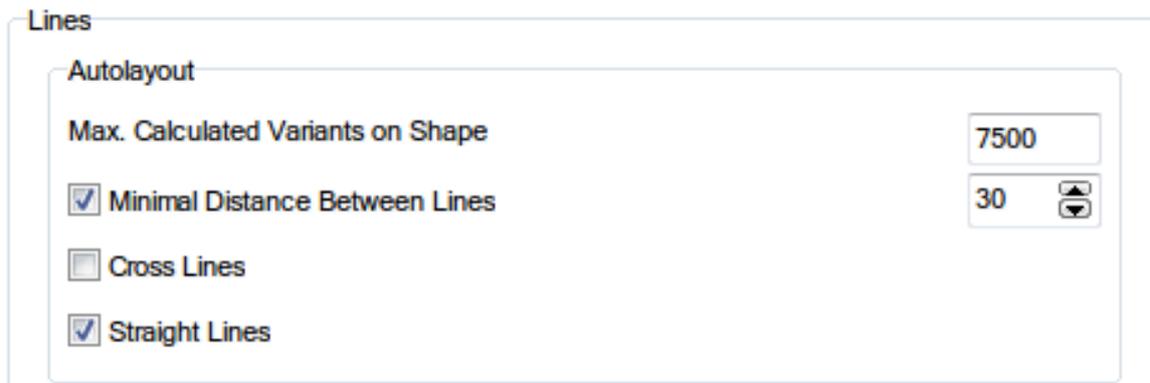
3. **Line Autolayout** dialog displays. You can now customize its settings.



Option	Description
Speed Settings	<p>Line Autolayout function tries to find the most optimal result from a set generated number of line variations. Generally, the more variants, the better the result and the more time Line Autolayout takes.</p> <p>You can select the <b>Simple</b> option and use the slider to set the number of generated variations. Or select the <b>Custom</b> option and enter the maximum number of calculated variants manually.</p> <p>Note that the bigger the number of variants and shapes on workspace is, the more time the process takes.</p>
Minimal Distance Between Lines	Determines the distance between lines on a shape edge.
Cross Lines	When checked, variants where lines are crossed are preferred.
Straight Lines	When checked, variants where lines are straight are preferred.

### **Line Autolayout settings**

Go to **Settings Menu | Options | Graphics | Autolayout tab**.



In **Lines | Autolayout** section you can define the default settings of Line Autolayout function.

## **Select Colors for Attributes and Keys**

### **To define different colors for attributes on your Workspace**

Select **Settings | Options | Model** section | **Physical Model/Logical Model| Entity** tab | **Attribute Colors** area.

## Physical Model

General | Workspace | Shape | Note Line | Entity

**Default**

Display Level	Attributes
Align	<input checked="" type="checkbox"/>
Display Data Types	<input checked="" type="checkbox"/>
Display Dictionary Types as Data Types	<input type="checkbox"/>
Display Keys Graphically	<input checked="" type="checkbox"/>
Display Key and Index Marks	<input checked="" type="checkbox"/>
Display Indexes	<input checked="" type="checkbox"/>
Display Not Null Mark	<input checked="" type="checkbox"/>
Gradient Effect	<input checked="" type="checkbox"/>

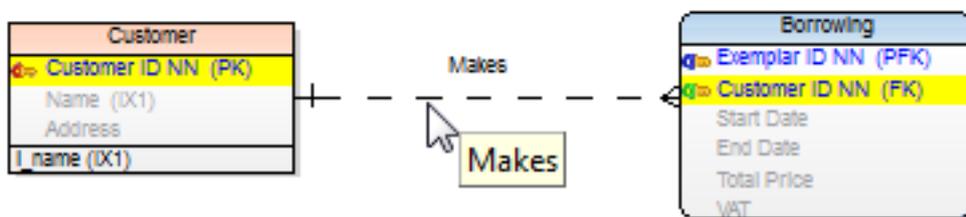
**Attribute Colors**

Primary Key	Red
Primary Foreign Key	Blue
Foreign Key	Green
Mandatory Attribute	Black
Optional Attribute	Medium Gray

**Form Settings**

Display Data Warehouse Type and Size	<input type="checkbox"/>
--------------------------------------	--------------------------

**i** TIP: When you point your mouse cursor at a relationship line, parent and child attributes are highlighted on the Workspace. You can set the color at: **Settings | Options | Graphics | Colors area | Highlight Color**.



# Show Grid and Grid Size

The **Grid** options are available directly from the toolbar or from the **View** menu.

## To show grid and set a grid size

1. Click  to show grid.
2. To change a grid size, click .
3. To snap objects of your ERD to grid, click .

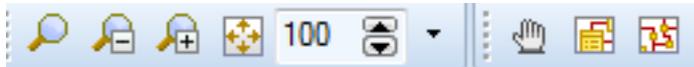
# Shift and Zoom in Your Model

Toad Data Modeler allows you to move around your large models quickly and views various parts of a model in various zoom levels.

For this purpose, the following functions can be used:

- Zoom (Zoom In, Zoom Out, Zoom Box)
- Loupe (magnifier glass)
- Overview (navigator)
- Move

All these functions are available on **Zoom Toolbar** or in **View** and **Windows** menus.



## Navigation Tips on Workspace

- CTRL + scroll mouse to zoom in/zoom out
- CTRL + Page Up/Page Down to zoom in/out
- Scroll mouse to move up/down on the Workspace
- SHIFT + scroll mouse to move right/left on the Workspace
- Holding down the middle mouse button to move around the entire page/Workspace
- Page Down, CTRL + down to move down
- Page Up, CTRL + up to move up
- CTRL + left to move left
- CTRL + right to move right
- Click  on **Zoom Toolbar** to adjust the zoom level so the entire ER diagram fits on screen.

## Objects on Workspace and keyboard arrows

- Move objects on Workspace using keyboard arrows.

**i** TIP: To set the move distance of one keypress, see **Settings Menu | Options | Graphics | Move Objects by (mm/10) (in tenths of millimeters)**.

- Select a shape on Workspace, hold down SHIFT and use the keyboard arrows to change size of the shape.

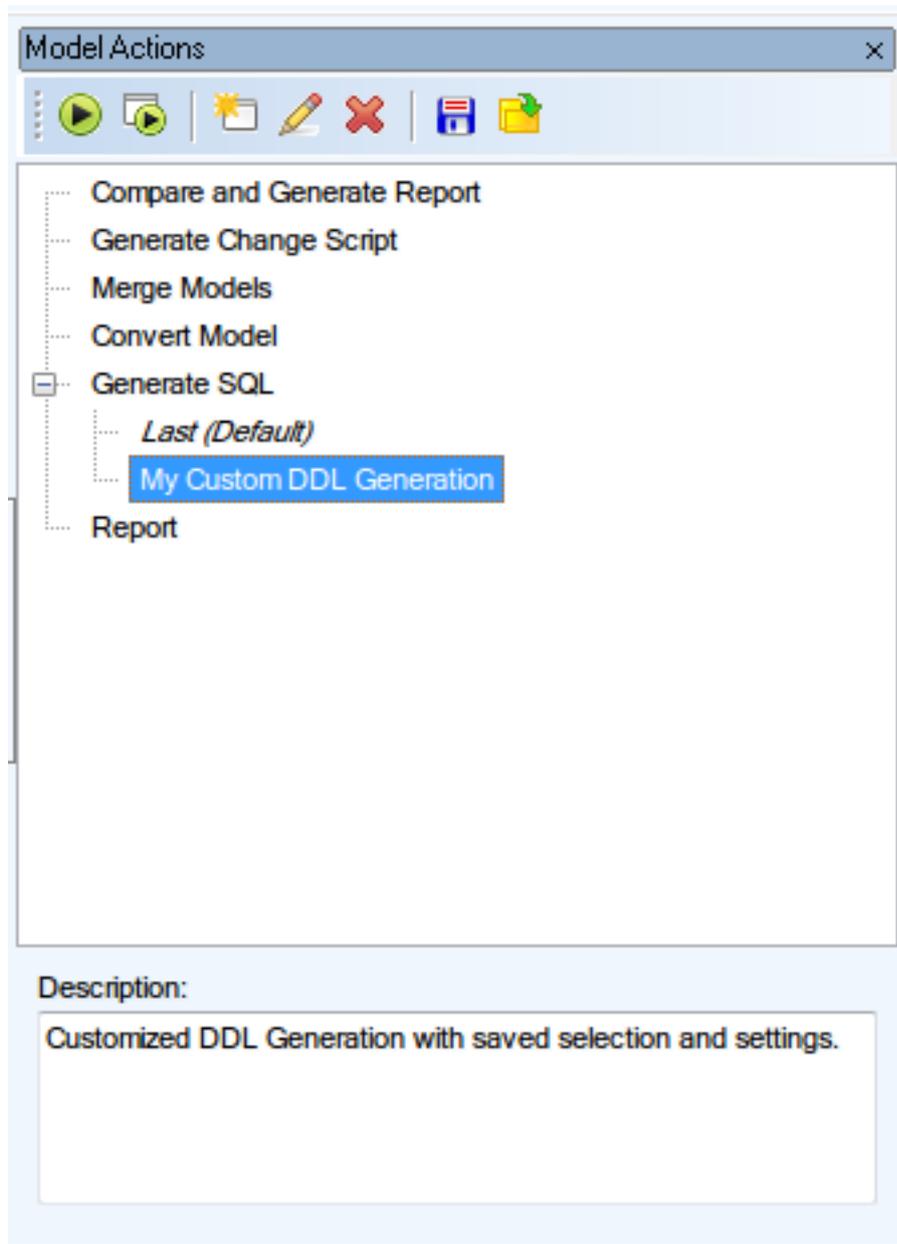
# Model Actions

## Basic Information

Model Actions can be opened by clicking **Window | Model Actions** or **Model | Manage Model Actions**.

This tool contains following Model features:

- Compare and Generate Report
- Generate Change Script
- Merge Models
- Convert Model
- Generate SQL
- Report



Using Model Actions helps you work more effectively. You can:

- Find the most used Model features all in one place
- Save Action Definitions for future use, even to a file
- Load Action Definitions to reduce time spent configuring Actions, even from a file
- Reduce the number of dialogs using **Run Promptly** button

**i** Note: Action Definition is a custom configuration of an Action (e.g. Convert Model). Instead of configuring Action every time from scratch, you are now able to save its Definition for later use (e.g. Convert Model from DB2 10.5 to DB2 10.1). In the future you can simply execute the Action Definition without worrying about its settings.

If you clicked through Action dialogs and forgot to save the Action Definition in progress, worry not. Every executed Action creates an Action Definition called *Last \*Action\**. It contains the settings of the last Action you ran.

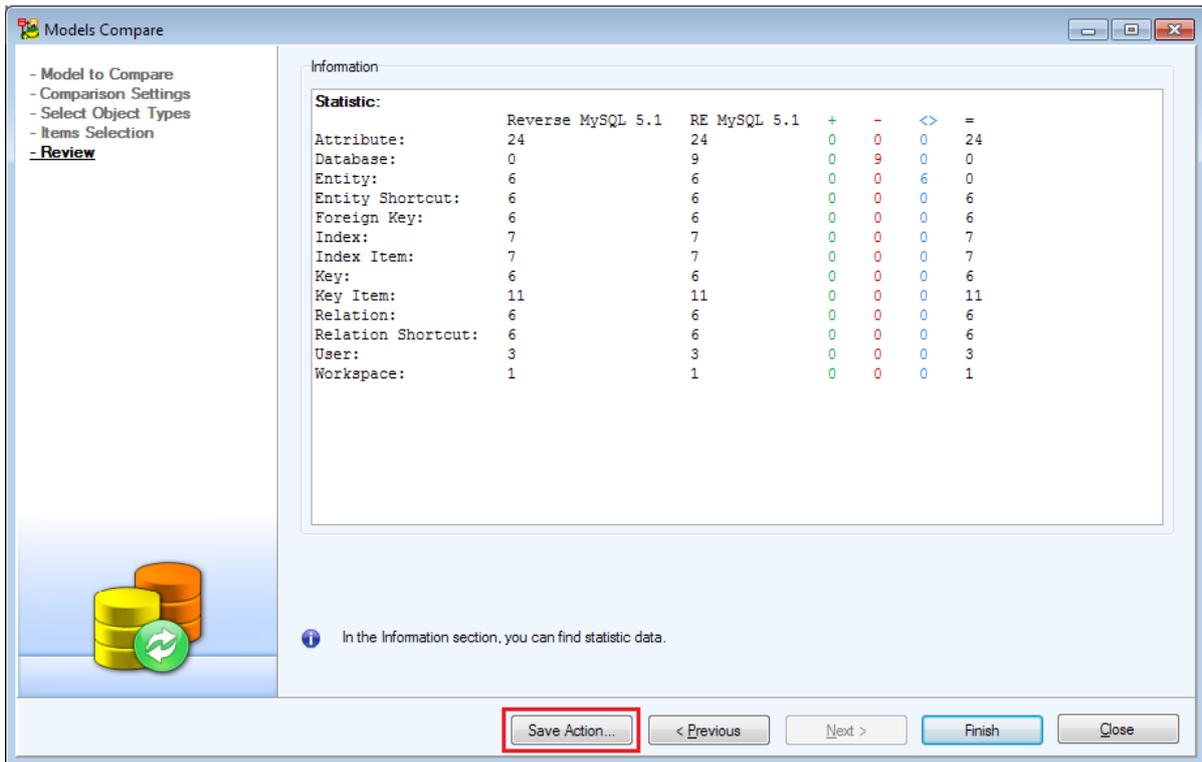
Also please note that all saved Action Definitions are part of the model. You can transfer Action Definitions from one Model to another by saving them to a file in original Model and loading them from the file in target Model.

## Model Actions Options

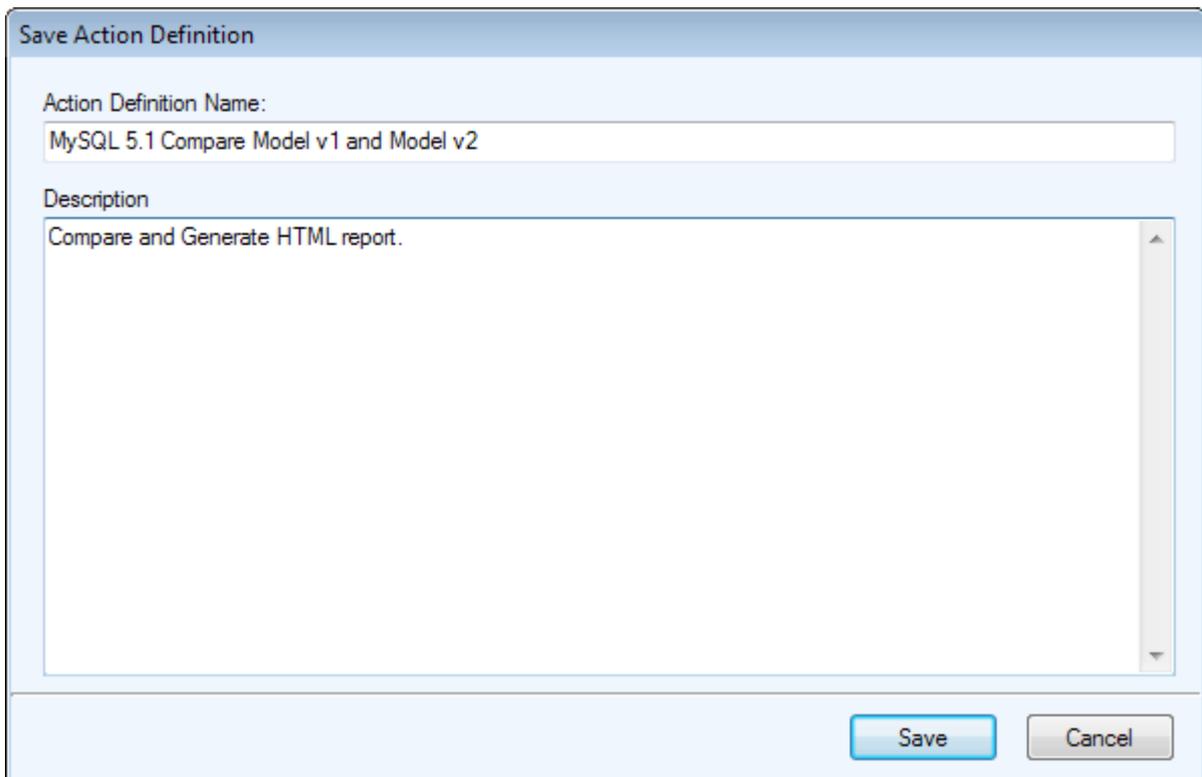
Option	Description
Run Action Definition Promptly	Allows you to execute Action Definition with minimum of dialogs shown.
Run Action Definition	Executes the selected Action/Action Definition. You can save the configuration as a new Action Definition in progress.
New Action Definition	Executes the selected Action/Action Definition. An extra dialog displays asking you to enter name and comment for new Action Definition.
Edit Action Definition	Allows you to go through the dialogs of an Action Definition and edit any options.
Delete Action Definition	Removes Action Definition from list.
Save Action Definition to file	Allows you to save selected Action Definition to a <b>.txad</b> file.
Load Action Definition from file	Allows you to load an Action Definition from <b>.txad</b> file.

## Saving Action Definition

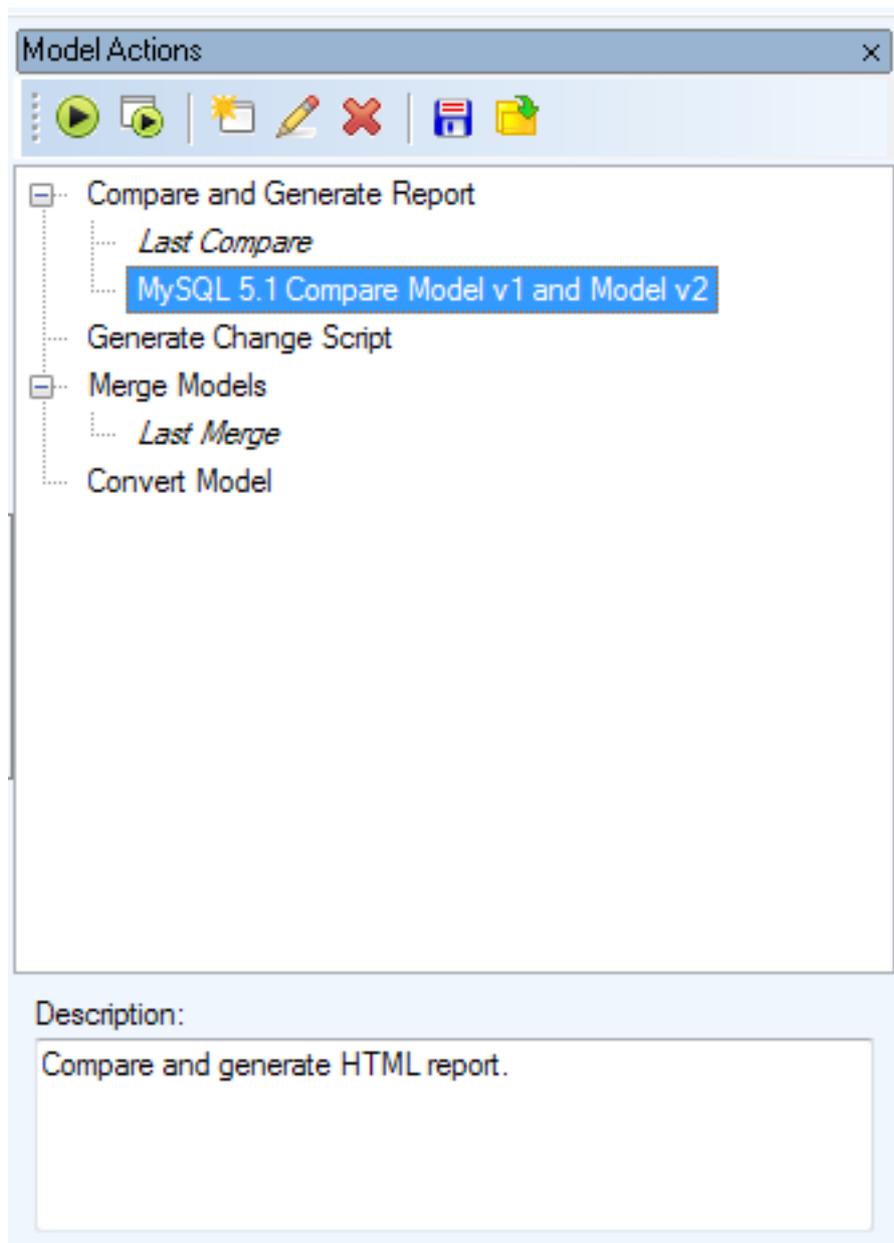
Action Definition can be saved by clicking the **Save Action** button which is located in the Review dialog (the last dialog every Action has).



Fill in the new Action Definition name and an optional description. Click **Save**.



The newly saved **Action Definition** will now show up in the list.



## Change Script

**Change Script** is useful for transferring changes done to your database or another model into the model you are currently working with. You can generate change script from a modified model or from a connection to a database or to a DDL script file. For executing the resulting script you will need another application such as Toad for Oracle. See [Toad for Oracle® as Default Editor](#) for more information.

For the list of databases for which **Change Script Generation** is supported in Toad Data Modeler see [Supported Databases](#). For those supported you can either generate [Simple Change Script](#) or complete change script using a **Generate Change Script** wizard:

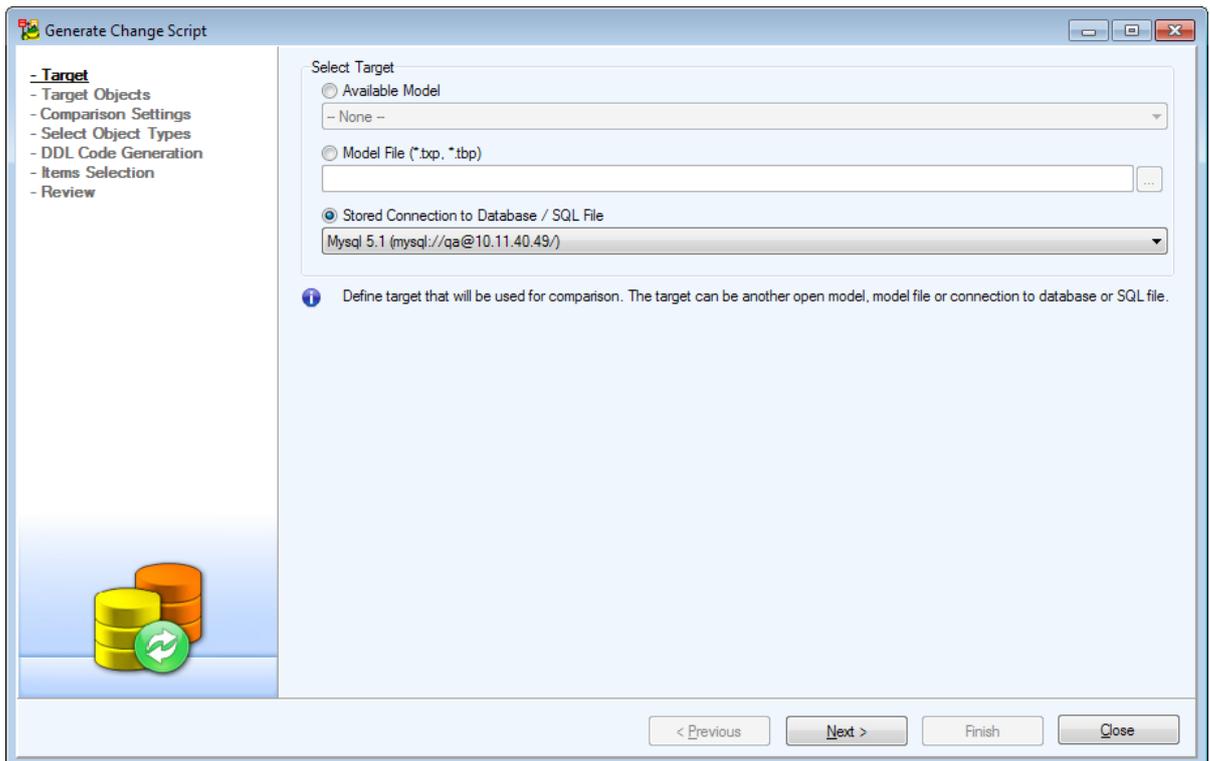
## To generate change script

1. Click **Run Generate Change Script** button or select **Model | Generate Change Script** or click **Model Actions | Generate Change Script**.



2. Select your target:

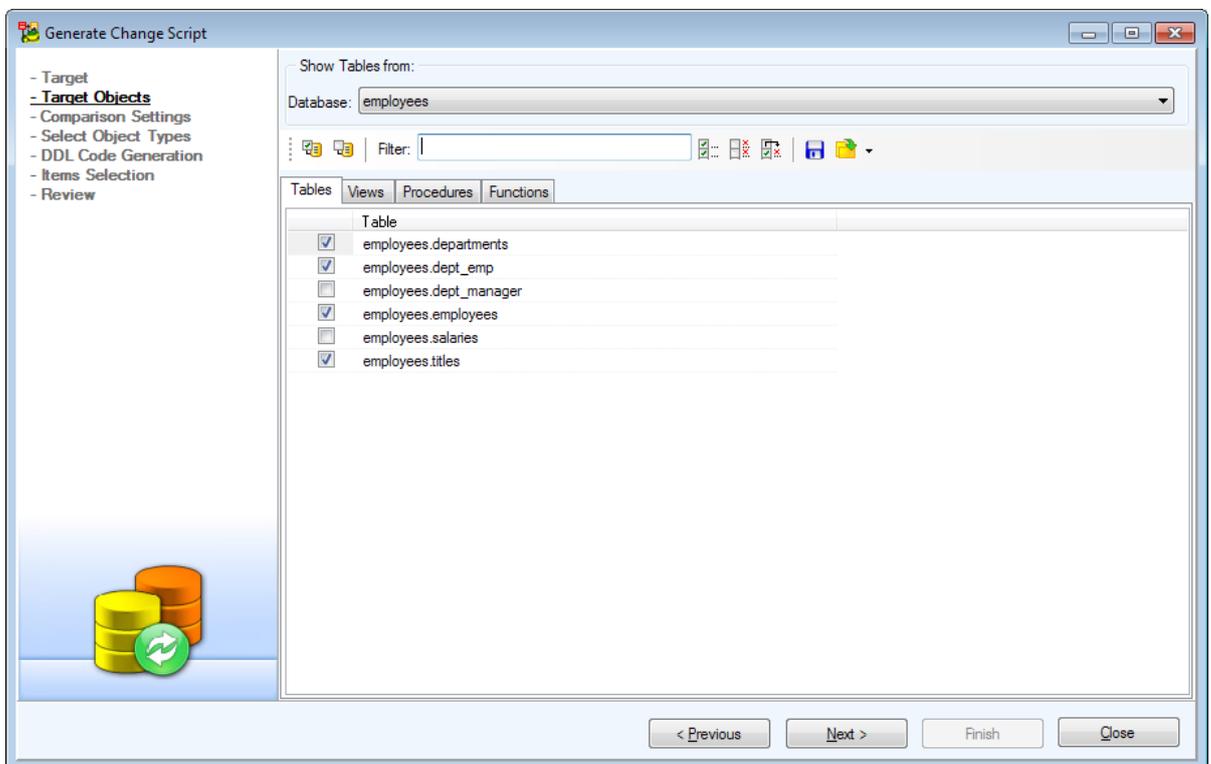
- **Available Model** - Select any open model of the same type
- **Model File** - Select and browse for any saved model file of the same type
- **Stored connection to Database/SQL file** - Select any saved connection. See [Connections](#) for more information



3. If you select a connection to a database or to a DDL file as the target you need to select the target objects you would like to include in your change script. The following buttons and the filter can be used to easily select objects:

Feature	Description
Schema	Select a specific schema you want to include in the target model, all tables or all selected tables.
Select All on All Tabs	Selects all objects on all tabs.
Deselect All on All Tabs	Deselects all objects on all tabs.
Filter	Type to filter objects.

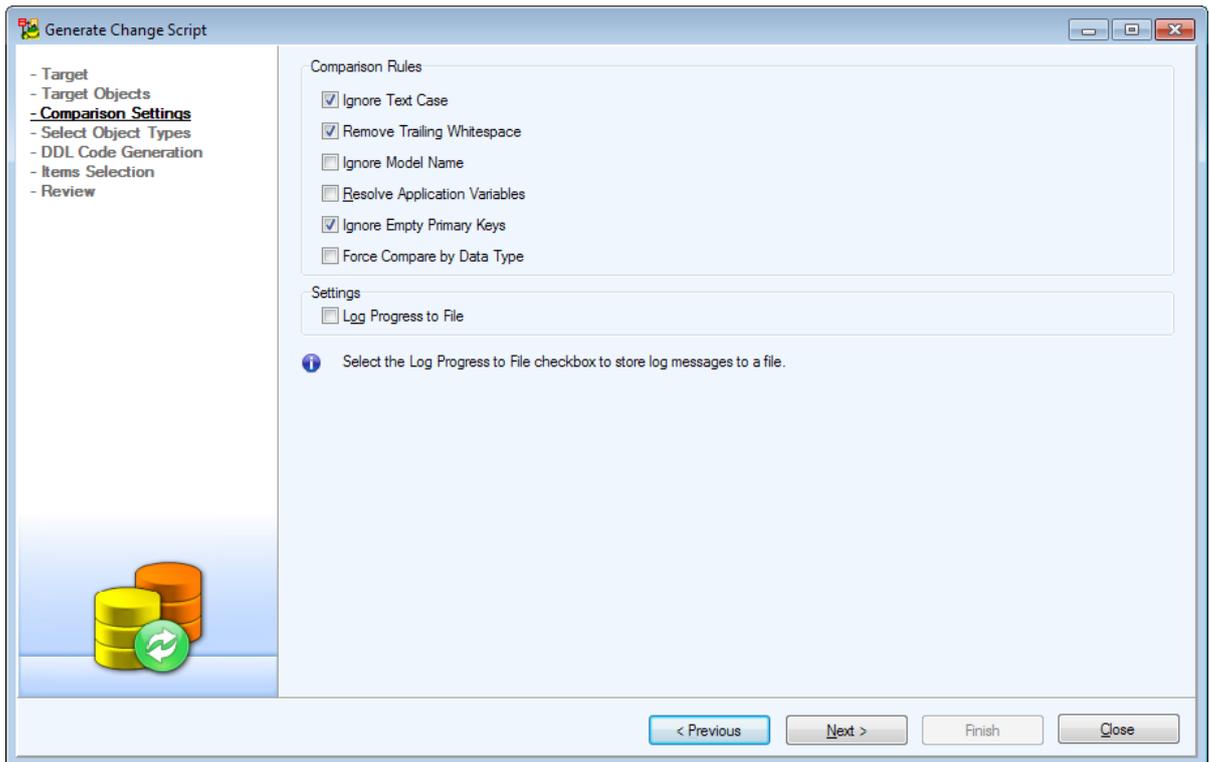
Feature	Description
	<p><b>i</b> TIP: Use wildcards - example: "*user" or "?ser".</p> <ul style="list-style-type: none"> <li>* - replaces unlimited number of characters</li> <li>? - replaces any single character</li> </ul>
Select All	Selects all object on the current tab.
Deselect All	Deselects all object on the current tab.
Invert Selection	Inverts selection on the current tab.
Select Parent Tables	Selects all parent tables of the currently selected tables.
Select Child Tables	Selects all child tables of the currently selected tables.
Select Parent and Child Tables	Selects all child and all parent tables of the currently selected tables.
Export Selection	Export selection as a *.wsxr file.
Import Selection	Imports a selection from *.wsxr file.



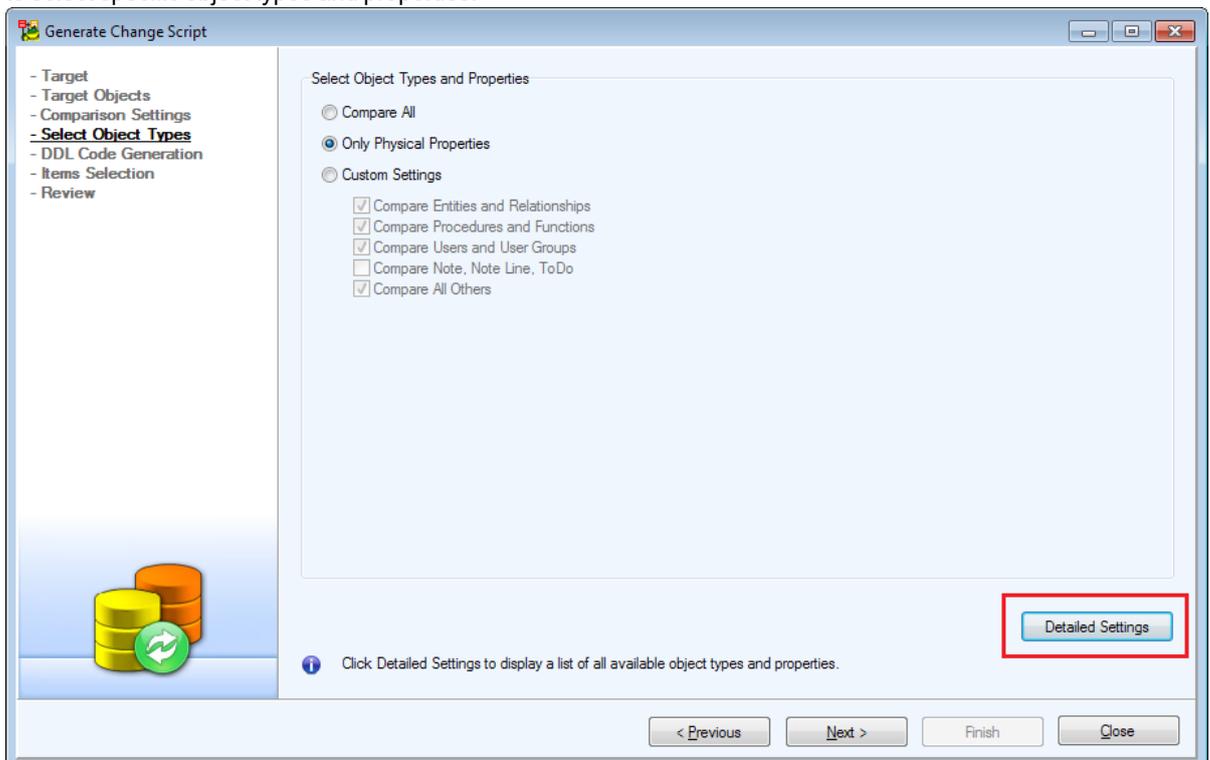
4. Adjust your comparison rules and settings:

Feature	Description
Ignore Text Case	Toad Data Modeler will ignore all differences in text case when pairing model objects. E.g. "Entity1", "eENTITY1", "ENTITY1" are treated as identical.

Feature	Description
Remove Trailing Whitespace	<p>Trailing whitespace, i.e. any whitespace characters at the end of a line including blank lines at the end of bodies of definitions, will be ignored during comparison.</p> <p>E.g. If checked, the following pieces of script will be treated as identical:</p> <pre>"Insert Into "Test" Values("aaa");"  "Insert Into "Test" Values("aaa");  "  "Insert Into "Test" Values("aaa");      "</pre>
Ignore Model Name	<p>Ignores the name of the model. Set the name in <b>Model Properties   Model</b>.</p>
Resolve Application Variables	<p>Resolves application variables during comparison. If left unchecked variables will be compared as variables.</p> <p><b>i</b> <b>NOTE:</b> In names, application variables are supported in the following objects: Relations, Keys, Check Constraints, and Indexes. For more information on application variables see <a href="#">Application Variables</a>.</p> <p>E.g. Your name is John Doe Your user name is "jdoe". An index is called "Index1_&lt;%Author%&gt;". If you check <b>Resolve Application Variables</b> the index will be treated as "Index1_jdoe". If unchecked it will be treated as "Index1_&lt;%Author%&gt;".</p>
Ignore Empty Primary Keys	<p>Primary keys with no attributes will be ignored during comparison.</p>
Pair Primary Keys Regardless of Names	<p>Primary keys will be mapped with no regard to their names.</p>
Force Comparison by Data Type	<p>Objects and properties will be compared also according to their data types.</p> <p>E.g. Source and target attributes in domains are named identically but they have different data types. If checked, they will be marked as different. If unchecked, they will be considered identical because their names and the names of the domains are identical.</p>
Log Progress to File	<p>Progress and errors will be logged to a file.</p>

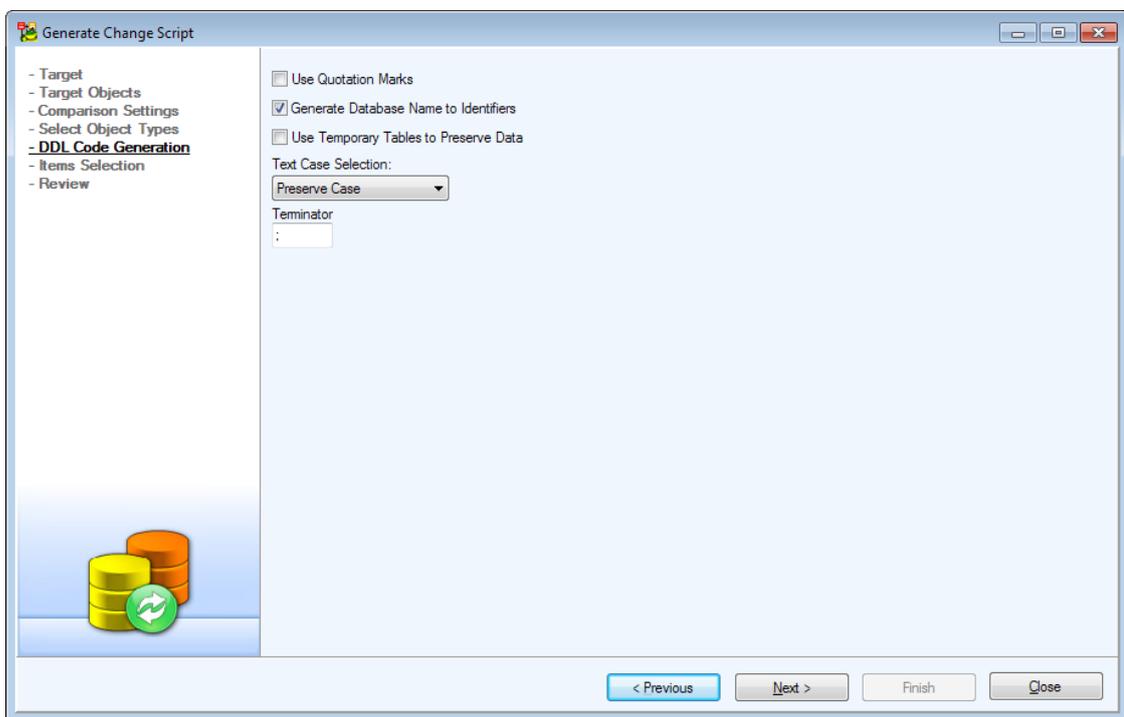


- By default **Simple Settings** are displayed. Select which object types and properties from several preset options will be compared between the models. Switch to **Detailed Settings** to display finer settings in order to select specific object types and properties.



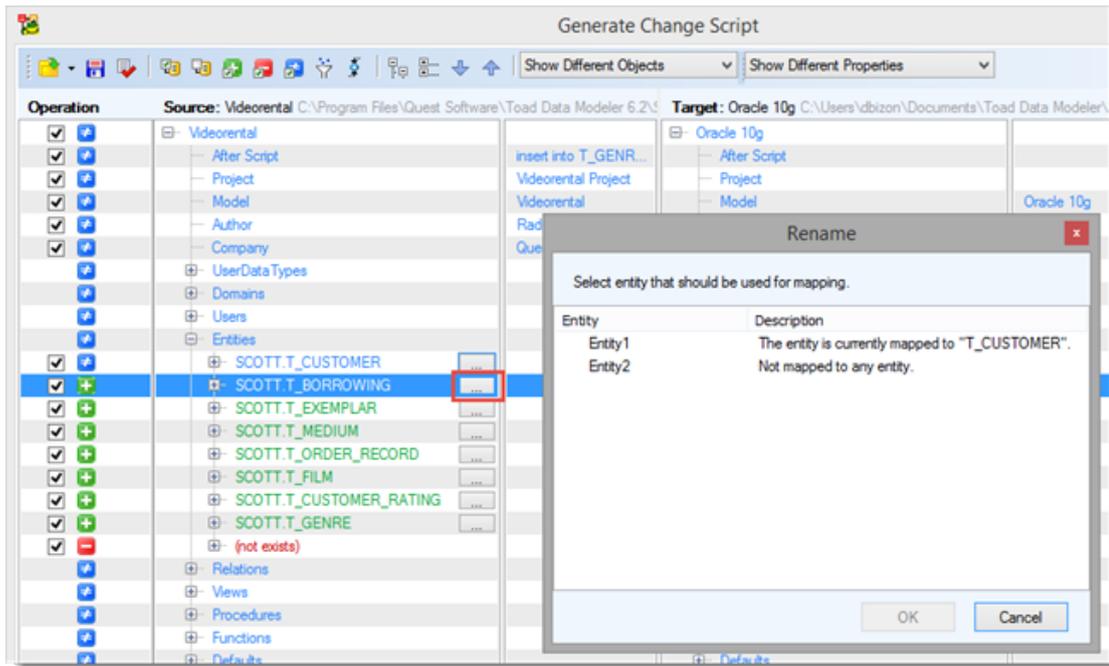
6. Adjust settings for **DDL Code Generation**. The following options for **DDL Code Generation** are common for more database types. The wording of the options is dependent on the database type selected.

Option	Description
<b>Use Quotation Marks</b>	Names of objects and properties will be generated in quotation marks (or as delimited identifiers etc.).
<b>Generate Database Name to Identifiers</b>	Names of objects and properties will be generated together with the name of the related database, user or schema or similar.
<b>Text Case Selection</b>	Select the case in which the change script will be generated: <ul style="list-style-type: none"> <li>• Preserve Case</li> <li>• Lower Case</li> <li>• Upper Case</li> </ul>



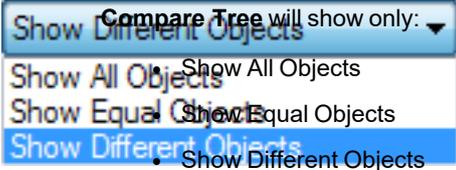
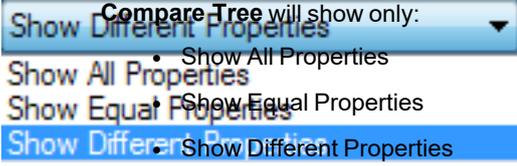
7. Map entities and attributes in the last step. Click the **Rename** button (  ) to map an entity or attribute to a target entity or attribute. The Description column in **Rename** dialog displays the current state of the object. Objects can be unmapped.

**Compare Tree** displays all differences between the two given models. The generated change script modifies the target model on the right so it matches the source model on the left. Check the changes you would like to execute.



### Compare Tree Overview

Option	Icon	Description
Import Selection		Import selection from a saved file.
Export Selection		Save selection to a file.
Verify		Runs the verification process.  The verification might return a warning  . Details are available in <b>Verification Log</b> .
Check All		Checks all items.
Uncheck All		Unchecks all items.
Check All to Add		Checks all items that exist in the Source model, but not in the Target model (CREATE).
Check All to Remove		Checks all items that do not exist in the Source model, but exist in the Target model (DROP).

Option	Icon	Description
Check All to Modify		Checks all items that exist in both models and are different (ALTER).
Wildcard Filter		Opens the <b>Wildcard</b> Dialog where you can define settings for bulk selection/deselection of the <b>Action</b> box of the items listed on page <b>Select Items</b> .
Refresh Necessitated Items		<p>Some objects are related together (e.g. entity and domain, entity and relationship). In case you uncheck an object or property in <b>Select Object Types</b> step and a related object or property is selected, the unchecked object or property will be automatically selected too.</p> <p>E.g. You uncheck a domain in <b>Select Object Types</b> but you keep an attribute of the domain type checked for conversion. In the next screen the domain will be selected for conversion (and highlighted in gray). This is because of its relationship with the attribute, which cannot exist without the domain.</p> <p>If you uncheck the attribute, the domain will still be checked for conversion. This is where you use this button. It inspects all checked objects and removes the domain highlighted in gray since the attribute is no longer checked. That means the domain is no longer necessary, since it has no relationships with currently checked objects and you unchecked it in <b>Select Object Types</b> step.</p>
Display options for objects		<p>Compare Tree will show only:</p> <ul style="list-style-type: none"> <li>Show All Objects</li> <li>Show Equal Objects</li> <li>Show Different Objects</li> </ul>
Display options for properties		<p>Compare Tree will show only:</p> <ul style="list-style-type: none"> <li>Show All Properties</li> <li>Show Equal Properties</li> <li>Show Different Properties</li> </ul>
Source	-	The updated or modified model.
Target	-	The model for which you want to generate change script.
Operation	-	<p>Check this checkbox to generate change script for the difference. Uncheck this checkbox to not generate change script for the difference.</p> <p>Default selection: Default selection of the <b>Action</b> checkboxes depend on your settings on page <b>Settings</b> in the <b>Options for Default Selection of Items</b> area. If you select all the options in this area, the <b>Action</b> checkboxes will be selected for all changes (CREATE, DROP and ALTER) by default.</p>

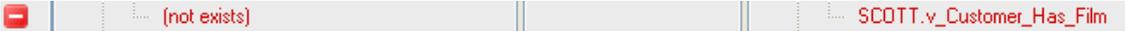
### Compare Tree Icons



Difference between original model and Target model.



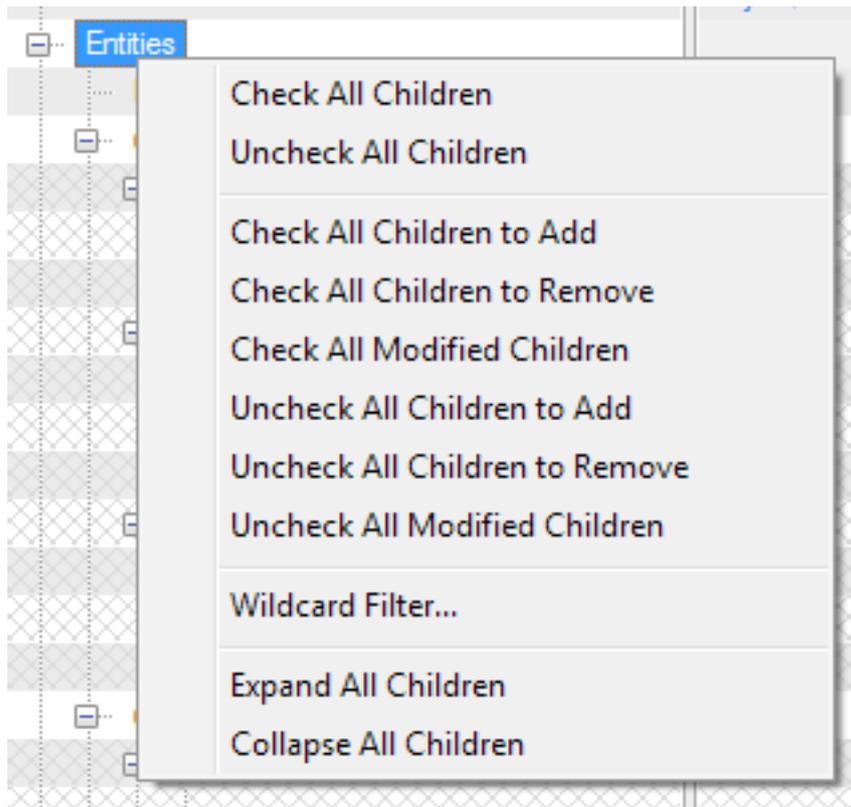
This object exists in original model but does not exist in Target model.



This object is missing in original model but exists in Target model.

### Right-click menu

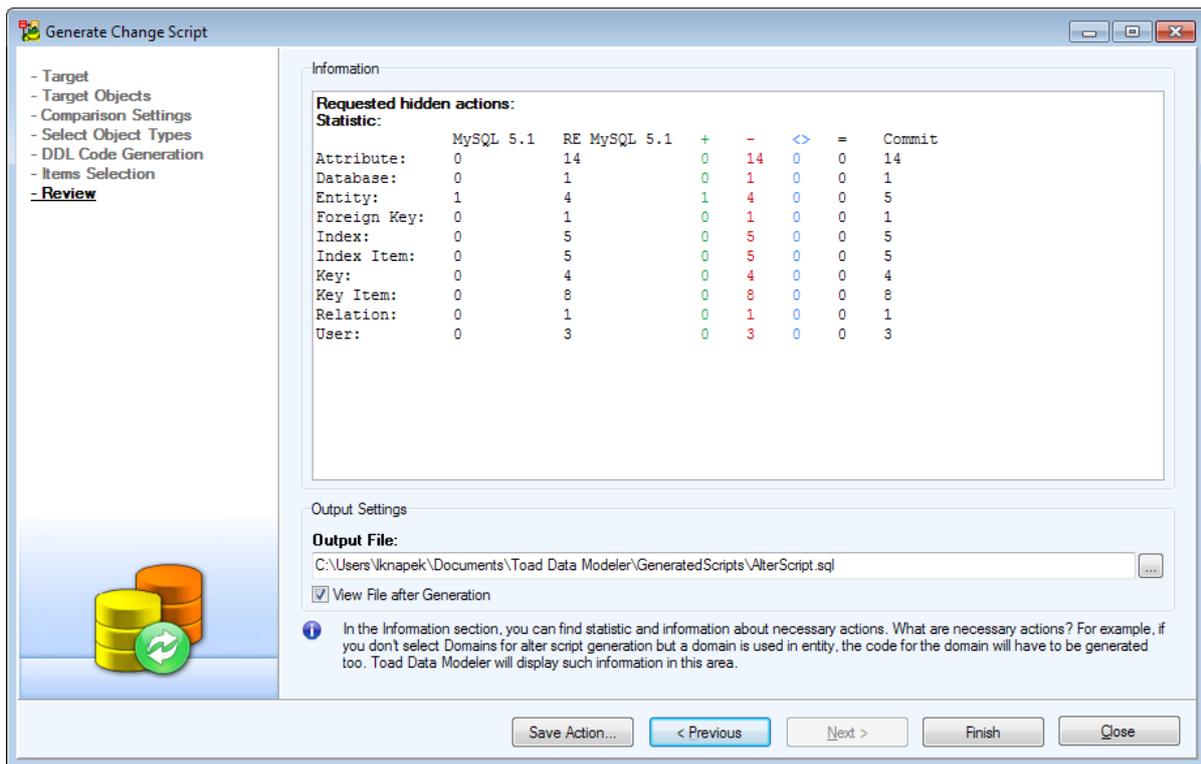
Right-click any item to display the following menu:



Option	Description
Check All Children	Checks the <b>Operation</b> checkbox of all children items.
Uncheck All Children	Unchecks the <b>Operation</b> checkbox of all children items.
Check All Children to Add	Checks the <b>Operation</b> checkbox of children items that exist in Source model but not in the Target model (  items) to generate the CREATE statement.

Option	Description
Check All Children to Remove	Checks the <b>Operation</b> checkbox of children items that are missing Source model and exist in Target model (  items) to generate the DROP statement.
Check All Modified Children	Checks the <b>Operation</b> checkbox of children items where properties differ (  items) to generate the ALTER statement.
Uncheck All Children to Add	Unchecks the <b>Operation</b> checkbox of children items that exist in Source model but not in the Target model (  items) to not generate the CREATE statement.
Uncheck All Children to Remove	Unchecks the <b>Operation</b> checkbox of children items that are missing Source model and exist in Target model (  items) to not generate the DROP statement.
Uncheck All Modified Children	Unchecks the <b>Operation</b> checkbox of children items where properties differ (  items) to not generate Change Script for this change.
Wildcard Filter	Opens the <b>Wildcard Dialog</b> where you can define settings for bulk selection/deselection of the <b>Operation</b> box of the items listed on page <b>Select Items</b> .
Expand All Children	Expands all sub-items of the selected item.
Collapse All Children	Collapses all sub-items of the selected item.

- Review your comparison. You can set your own path for change script output file here. Click **Save Action** to save the comparison as Model Action for repeated use. See [Model Actions](#) for more information. Click **Finish** to generate the script.



## Change Script Generation - Temporary Tables

You can decide whether or not to use Temporary Tables during Change Script generation. The settings are available in **Settings | Options | Physical Model | \*Specific Database\* | Change Script Settings**.

- **Use Temporary Tables to Preserve Data** – If you uncheck this checkbox, a sequence of commands DROP TABLE / CREATE TABLE will be generated instead of temporary tables. (The exception are the changes in data types where ALTER TABLE command is generated.)
- **Use Temporary Tables to Preserve Data on Data Type Change** – This option is only available with **Use Temporary Tables to Preserve Data** option checked. Unchecking this option results in generating ALTER commands where there are changes in data types instead of using Temporary Tables.

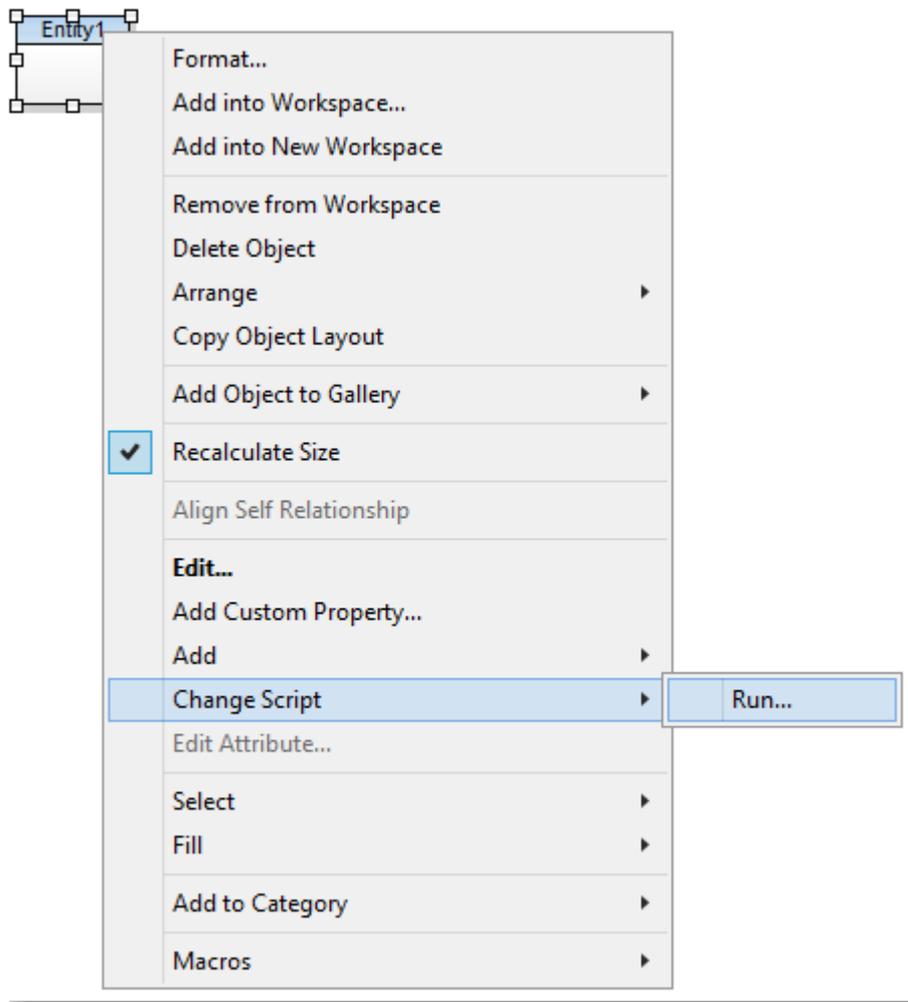
**i Note:** For Microsoft Azure SQL Database, Microsoft SQL Server, and PostgreSQL, if you uncheck the **Use Temporary Tables to Preserve Data** checkbox, temporary UDT/DictType will not be generated either. (They are created together with Temporary Tables.)

## Simple Change Script

**Simple Change Script** is generated for one entity. You might find it useful when you want to transfer minor changes in your model to your database.

### To generate Simple Change Script

- Right-click the entity you want to generate change script for and select **Change Script | Run**



- Follow the steps in [Change Script](#) to create simple change script

## Merge Models

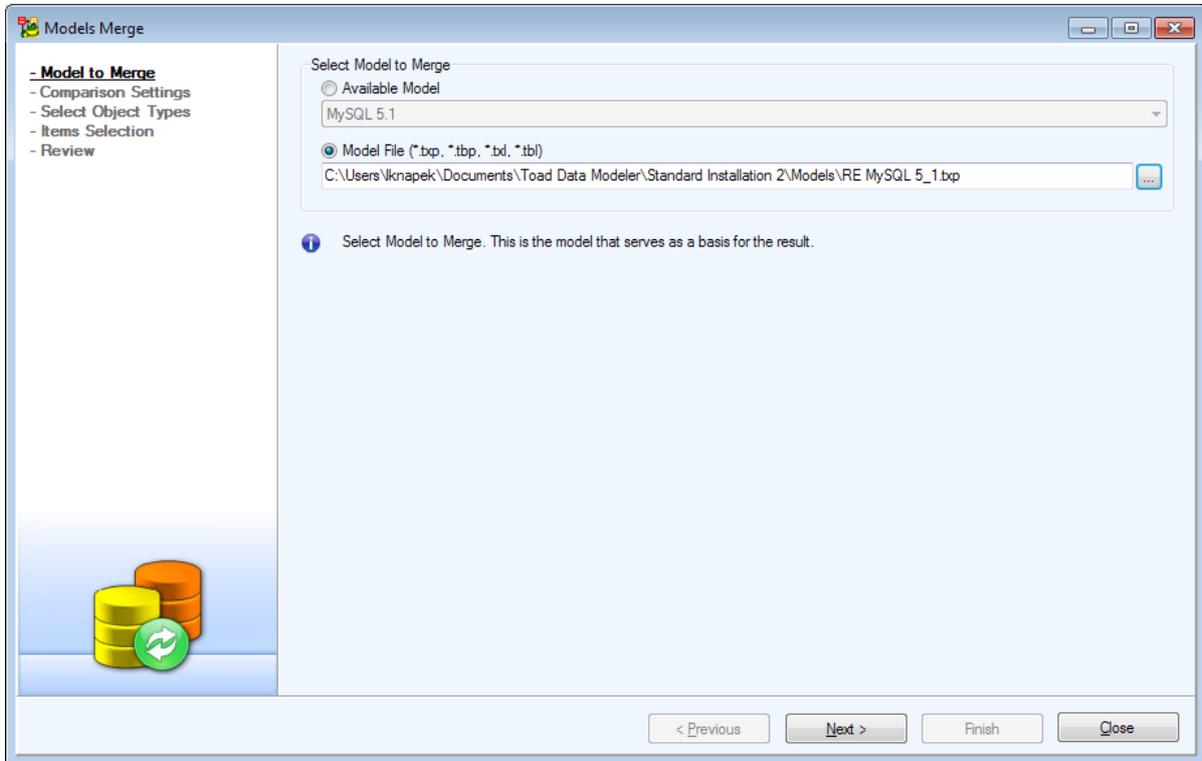
Toad Data Modeler offers you two options to merge your models:

- **Simple Model Merge** - allows you to merge physical models quickly due to skipping several settings dialogs and Model Comparison. The target model will always be overwritten.
  - **Simple Model Merge** is available in **Model Menu | Simple Merge**.
  - [Simple Model Merge](#)
- **Model Merge** in the **Model Actions** - allows you to see differences between two models, select particular items to merge and merge the models either to already existing model or to a new model.

- Example: You work with several models. Use **Model Merge** to compare the models and merge them either to an existing model, or to a new model altogether.

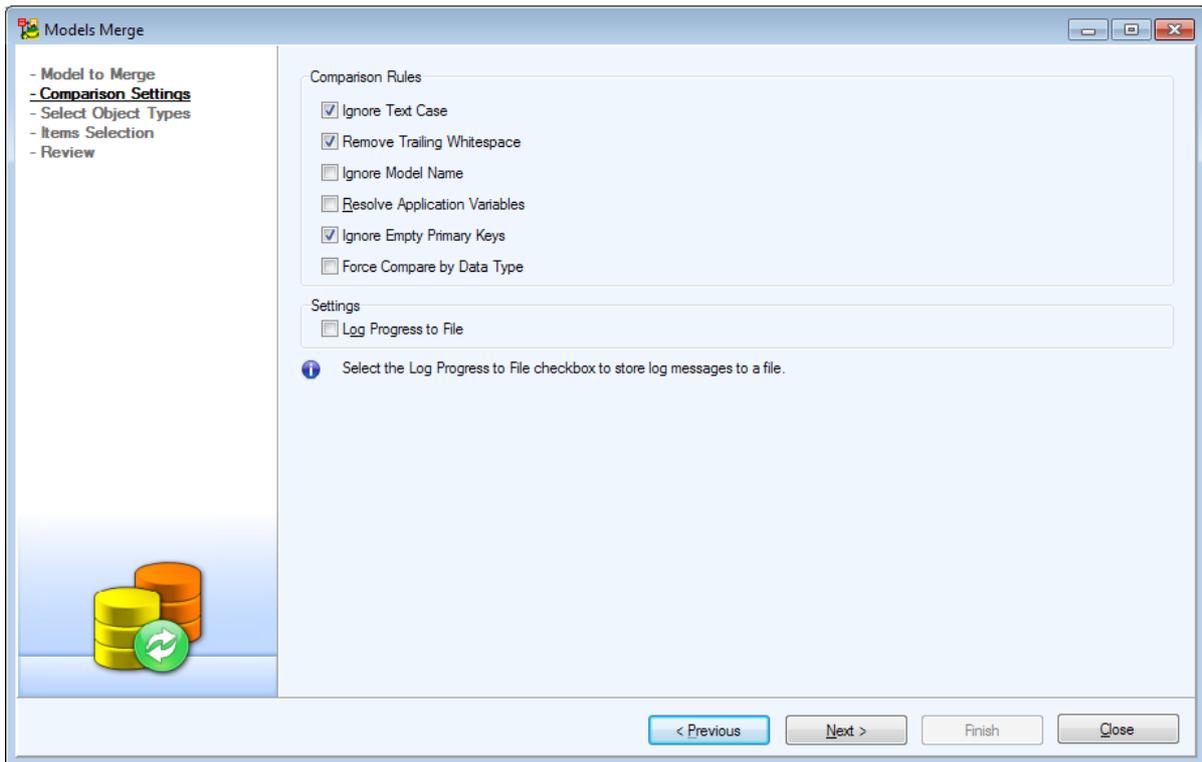
## How to Merge Models

Model Merge can be found in **Model Actions (Window Menu | Model Actions)**, or in **Model Menu | Merge Models | Run**.

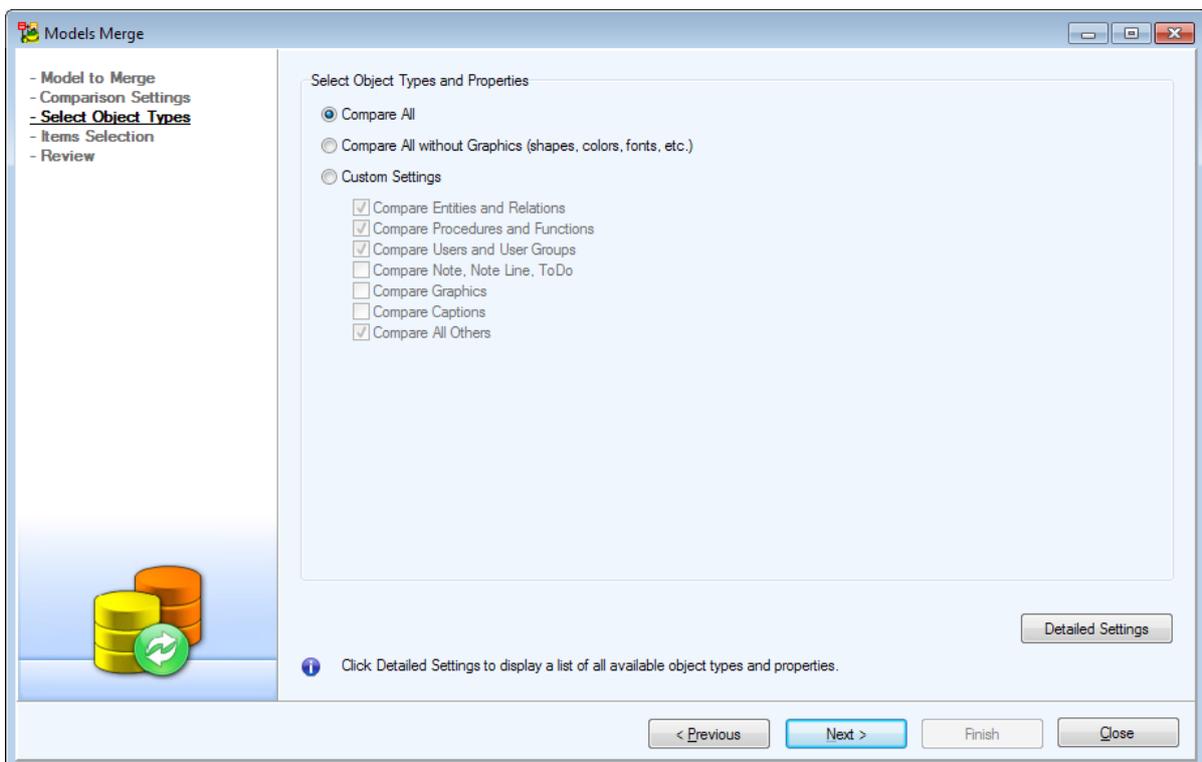


- In the first step select either an opened model or a model from a file.

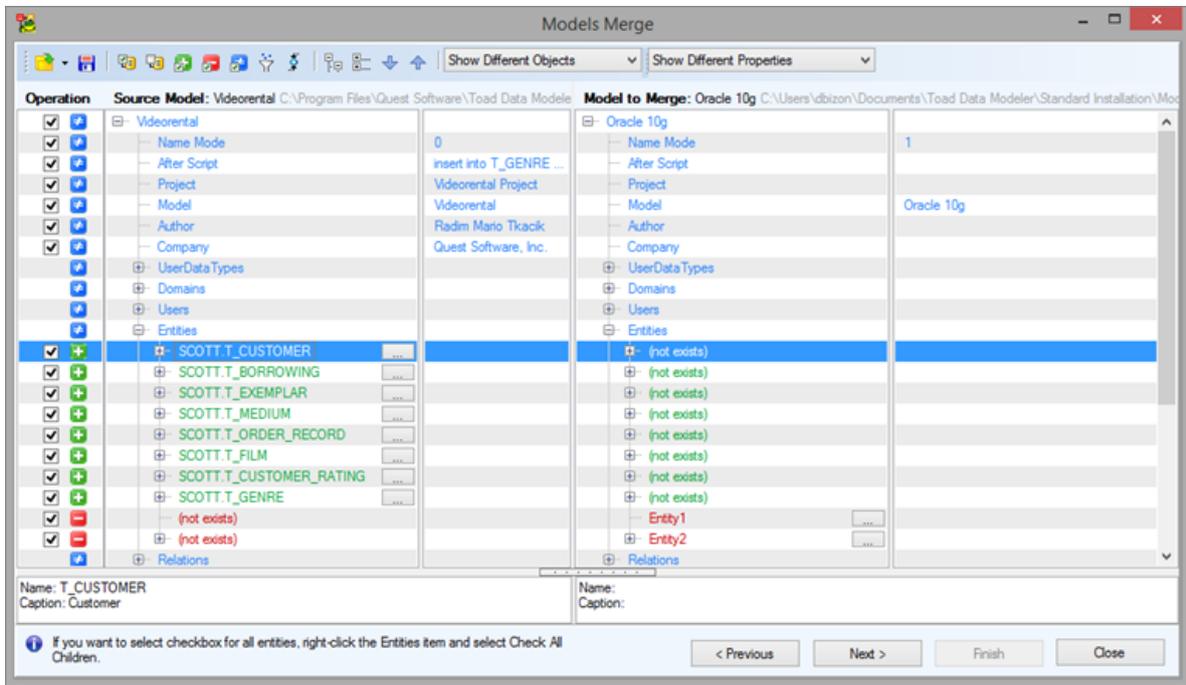
**i** note: The selected model has to be of the same database platform and version. You cannot compare DB2 10.5 and MySQL 8.0 models.



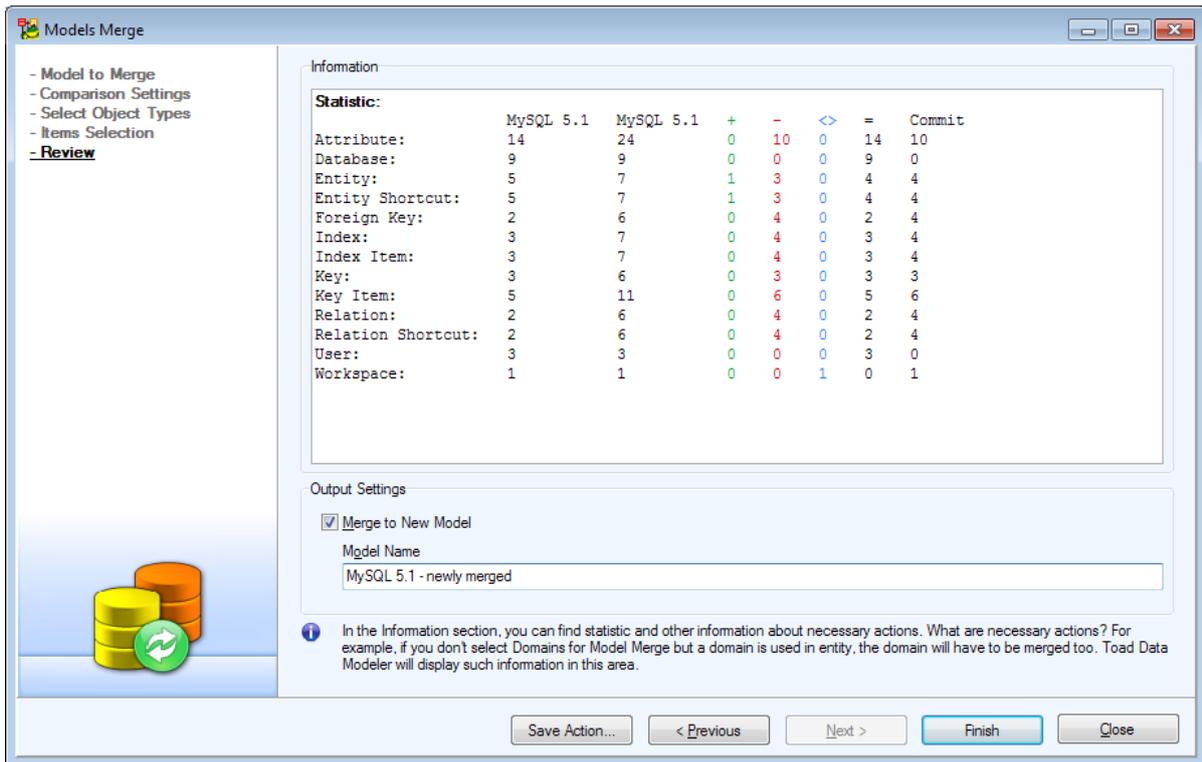
- The next dialog presents you with a couple of setting related to **Model Comparison**. When you're done editing the options, click on Next.



- Now you have to choose what types of objects will be included in the comparison. If you wish to select objects in more detail, click on **Detailed Settings**.
  - Entities and attributes can be easily mapped here in the last step
  - Double-click the button (...) to map an entity or attribute to a target entity or attribute
  - The Description column displays the current state of the object. Objects can be unmapped



- You are now presented with Compare Tree dialog displaying all differences between the two given models. Check or uncheck items to select what objects will be merged.



- Check the Review screen which show you the changes for each object type. If you want to merge models into a new model, check **Merge to New Model** and enter its name.

**i** TIP: If you plan to do this action again in the future, you might want to click the **Save Action** button to save this **Action Definition**. See **Model Actions** for more information.

### Compare Tree Overview

Option	Icon	Description
Import Selection		Import selection from a saved file.
Export Selection		Save selection to a file.
Verify		Runs the verification process. The verification might return a warning  . Details are available in <b>Verification Log</b> .
Check All		Checks all items.

Option	Icon	Description
Uncheck All		Unchecks all items.
Check All to Add		Checks all items that exist in the Source model, but not in the Target model (CREATE).
Check All to Remove		Checks all items that do not exist in the Source model, but exist in the Target model (DROP).
Check All to Modify		Checks all items that exist in both models and are different (ALTER).
Wildcard Filter		Opens the <b>Wildcard</b> Dialog where you can define settings for bulk selection/deselection of the <b>Action</b> box of the items listed on page <b>Select Items</b> .
Refresh Necessitated Items		<p>Some objects are related together (e.g. entity and domain, entity and relationship). In case you uncheck an object or property in <b>Select Object Types</b> step and a related object or property is selected, the unchecked object or property will be automatically selected too.</p> <p>E.g. You uncheck a domain in <b>Select Object Types</b> but you keep an attribute of the domain type checked for conversion. In the next screen the domain will be selected for conversion (and highlighted in gray). This is because of its relationship with the attribute, which cannot exist without the domain.</p> <p>If you uncheck the attribute, the domain will still be checked for conversion. This is where you use this button. It inspects all checked objects and removes the domain highlighted in gray since the attribute is no longer checked. That means the domain is no longer necessary, since it has no relationships with currently checked objects and you unchecked it in <b>Select Object Types</b> step.</p>

Display options for objects

Compare Tree will show only:

- Show Different Objects
- Show All Objects
- Show Equal Objects
- Show Different Objects

Display options for properties

Compare Tree will show only:

- Show Different Properties
- Show All Properties
- Show Equal Properties
- Show Different Properties

Source	-	The updated or modified model.
Target	-	The model for which you want to generate change script.
Operation	-	Check this checkbox to generate change script for the difference.

Option	Icon	Description
--------	------	-------------

Uncheck this checkbox to not generate change script for the difference.  
 Default selection: Default selection of the **Action** checkboxes depend on your settings on page **Settings** in the **Options for Default Selection of Items** area. If you select all the options in this area, the **Action** checkboxes will be selected for all changes (CREATE, DROP and ALTER) by default.

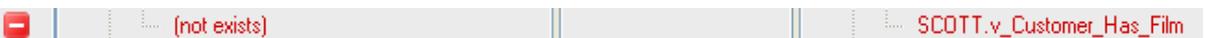
### Compare Tree Icons



Difference between original model and Target model.



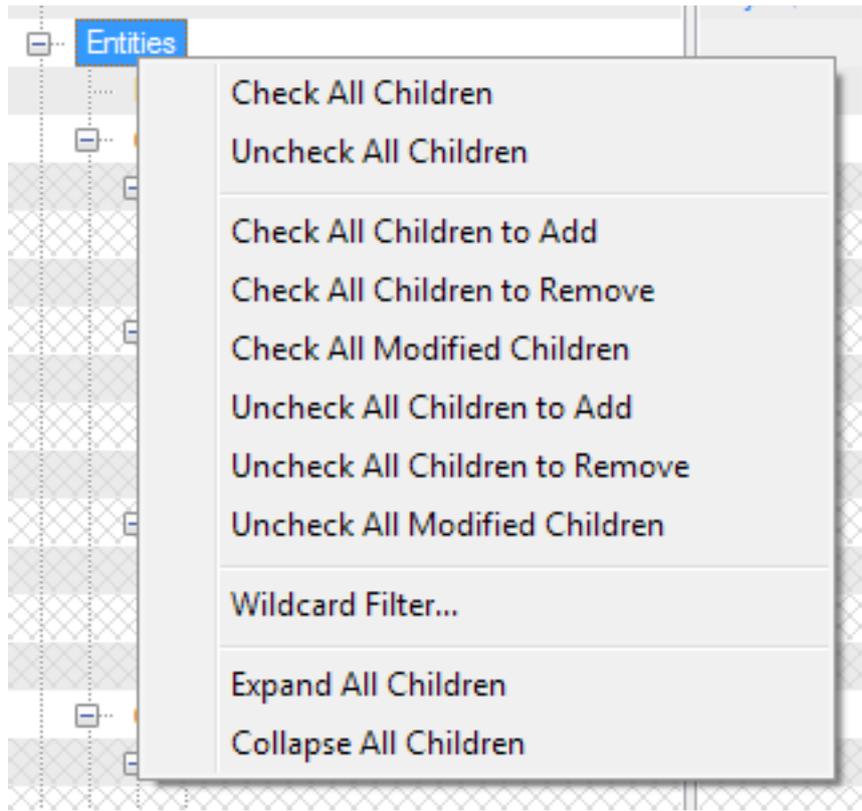
This object exists in original model but does not exist in Target model.



This object is missing in original model but exists in Target model.

### Right-click menu

Right-click any item to display the following menu:



Option	Description
Check All Children	Checks the <b>Operation</b> checkbox of all children items.
Uncheck All Children	Unchecks the <b>Operation</b> checkbox of all children items.
Check All Children to Add	Checks the <b>Operation</b> checkbox of children items that exist in Source model but not in the Target model (  items) to generate the CREATE statement.
Check All Children to Remove	Checks the <b>Operation</b> checkbox of children items that are missing Source model and exist in Target model (  items) to generate the DROP statement.
Check All Modified Children	Checks the <b>Operation</b> checkbox of children items where properties differ (  items) to generate the ALTER statement.
Uncheck All Children to Add	Unchecks the <b>Operation</b> checkbox of children items that exist in Source model but not in the Target model (  items) to not generate the CREATE statement.
Uncheck All Children to Remove	Unchecks the <b>Operation</b> checkbox of children items that are missing Source model and exist in Target model (  items) to not generate the DROP statement.
Uncheck All Modified Children	Unchecks the <b>Operation</b> checkbox of children items where properties differ (  items) to not generate Change Script for this change.
Wildcard Filter	Opens the <b>Wildcard Dialog</b> where you can define settings for bulk selection/deselection of the <b>Operation</b> box of the items listed on page <b>Select Items</b> .
Expand All Children	Expands all sub-items of the selected item.
Collapse All Children	Collapses all sub-items of the selected item.

## Simple Model Merge

This feature allows you to merge two physical models very quickly. However, in comparison to the model merge in the **Sync & Convert Wizard**:

- You cannot select particular items for the model merge (e.g. not to merge particular entity). You can select only Object Types and Properties for the model merge.
- The target model will always be overwritten.

### **To use the Simple Model Merge feature**

1. Open both models that you want to merge.
2. Make the modified (updated) model active in the Application Window (source model).
3. Select **File | Synchronization | Simple Model Merge**.

4. From the **To Model** box, select a model that you want to update (target model). - This model will be overwritten.
5. See other options.

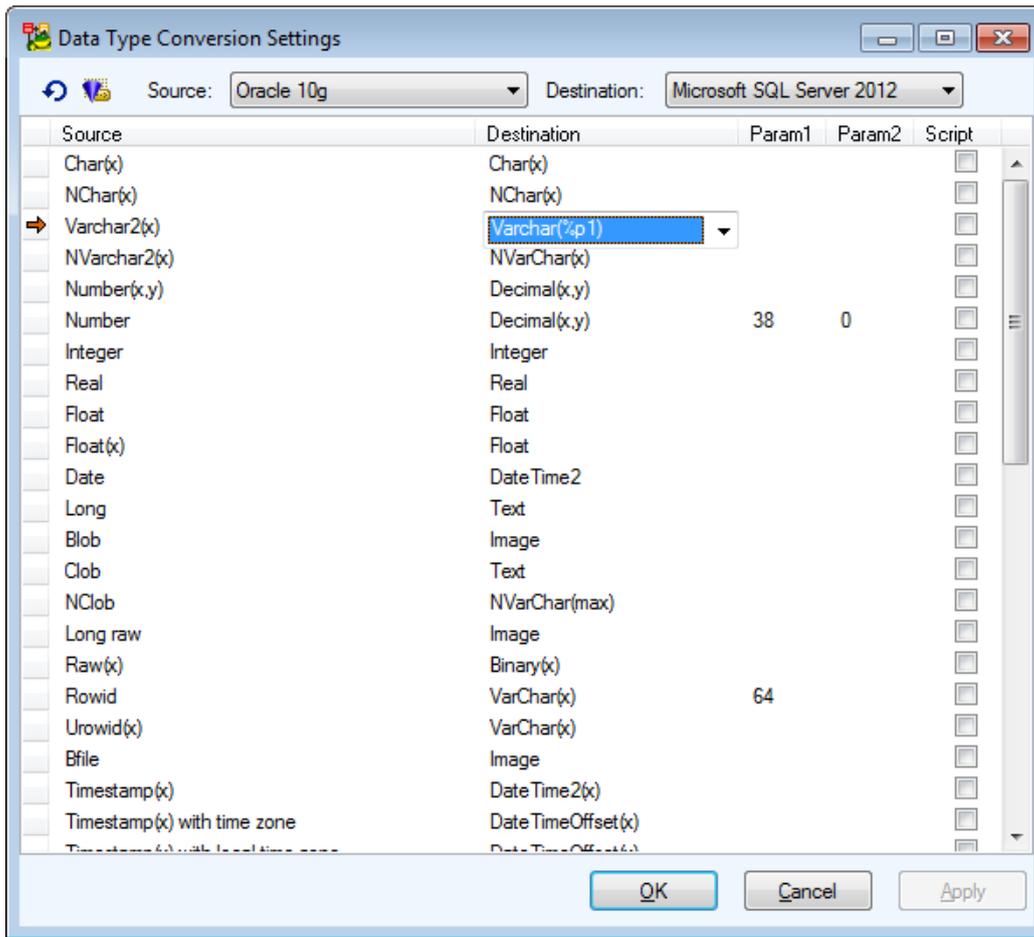
Option	Description
To Model	Select a target model.
More>>	Click this button to see and modify Object Types and Properties for the model merge. <a href="#">Object Types and Properties - OTPs</a>
Close after Merge	Closes the <b>Merge</b> dialog after the process is finished.
Merge	Executes the process of model merge.
Close	Closes the <b>Merge</b> dialog.

6. Click **Merge**.

## Data Type Conversion Settings

Before you start the conversion process, you can set conversion settings for data types.

1. Enable Expert mode: Select **Settings | Options | General** | check the **Expert Mode** checkbox.
2. From the main menu, select **Expert Mode | Expert Mode Settings | Data Type Conversion Settings**.



3. In the **Source** and **Destination** boxes choose the desired databases.
4. Press  to display available data types. Click  to view only exceptions.

You can see a list of data types of source model that will be converted to data types in destination model.

**Example:** Oracle **Blob** data type will be converted to Microsoft SQL Server 2012 model as **Image** data type. You can change the destination model data type.

**i** Note: The modified database conversion settings are saved to **DataTypeConversion.txn** file stored by default in: C:\Users\\*username\*\AppData\Roaming\Quest Software\Toad Data Modeler\\*Installation name\*\Configs. If you want to restore the original conversion settings, you can simply delete this .txn file.

5. Confirm **OK**.

### Execute Script

If you want to use your own script for data type conversion, click the **Go to Exception Script** button. A dialog offer you to create a new script. Confirm and the newly created script will be shown in **Script Editor**, where you can edit it as you like.

**Example:**

```

function Convert() {
    var Log = System.CreateObject('Log');
    Log.Information('Konverze');
    Log.Information('Input MD: '+ InputModelDef.Abbrev );
    Log.Information('Output MD: '+ OutputModelDef.Abbrev );

    Log.Information(InputDataType.ConversionID.Code);

    var outDT
    if (InputDataType.ConversionID.Code == 'C_X'){
        Log.Information('HERE');
        for( i = 0; i < OutputDataTypes.Count; i++ )
        {
            outDT = OutputDataTypes.GetObject( i );
            Log.Information( outDT.Name );
            if (outDT.Name=='Box') {
                return(i);
            }
        }
    }

    return(0);
}

```

## PER - PER Conversion Information

See some basic information on what is converted and how during the PER to PER conversion.

- Data Types are converted.
- Permissions are compared by names.
- SQL code is copied and commented not to generate any code.

See other details here:

Item to Convert	Model A (Source Model)	Model B (Destination Model)	Result
<b>Data Types</b>			
	Supported data type	Supported data type	Same data type
Example*:	Integer	Integer	Integer
	Supported data type	Equivalent data type	Equivalent data type
Example**:	Float	Real	Real
<b>Permissions</b>			
	Supported Permission	Supported Permission	Same Permission
Example:	SELECT	SELECT	SELECT
	Supported Permission	Unsupported Permission	--None--
Example:	DROP (MySQL 8.0)	--None-- (Microsoft SQL 2019)	--None-- (Microsoft SQL 2019)
	Unsupported Permission	Supported Permission	Model B Permission
Example:	--None-- (Microsoft SQL 2019)	DROP (MySQL 8.0)	DROP (MySQL 8.0) (DROP preserved in already existing model.)
<b>Deny Permission</b>			
	Deny Permission Supported	Deny Permission Supported	Deny Permission Converted
	Deny Permission Supported	Deny Permission Unsupported	--None--
	Deny Permission Unsupported	Deny Permission Supported	Model B Deny Permission
Example:	MySQL 8.0	Microsoft SQL	Microsoft SQL

Item to Convert	Model A (Source Model)	Model B (Destination Model)	Result
		2019	2005 (Deny permission preserved in already existing model.)
<b>Grantor</b>			
	Grantor Supported	Grantor Supported	Grantor Converted
	Grantor Supported	Grantor Unsupported	--None--
	Grantor Unsupported	Grantor Supported	Grantor is empty.
Example:	<i>MySQL 8.0</i>	<i>Microsoft SQL 2019</i>	Microsoft SQL 2019 (Grantor preserved in already existing model.)
<b>Users and User Groups</b>			
	Supported	Supported	Converted successfully
	Supported	Unsupported	--None--
	Unsupported	Supported	--None--
Example:	Users unsupported in PostgreSQL	Users supported in Microsoft SQL 2019	--None--

**Model A** - Currently opened model that you want to convert.

**Model B** - Model to which you want to convert Model A.

Complete Model A is converted to Model B.

### Other Information

- Conversion between PostgreSQL and Microsoft SQL: Serial and BigSerial data types in PostgreSQL are converted to Identity in Microsoft SQL. Identity in Microsoft SQL is converted to combination of sequence and default "nextval" in PostgreSQL.

- Conversion between PostgreSQL and MySQL: Serial data type in PostgreSQL is converted to Autoincrement in MySQL. Autoincrement in MySQL is converted to combination of sequence and default "nextval" in PostgreSQL.
- Conversion between PostgreSQL and Oracle: Serial and BigSerial data types in PostgreSQL are converted to combination of sequence and a sequence selected for attribute in Oracle. A sequence selected for attribute in Oracle is converted to default "nextval" in PostgreSQL (the sequence is converted automatically).
- Conversion from Oracle to Microsoft SQL (and Microsoft SQL to Oracle): Automatic conversion of Identity in Microsoft SQL to Sequence in Oracle (and back).
- Conversion from Oracle to MySQL (and back): Sequence in Oracle are converted to Autoincrement in MySQL (and back).
- Conversion from Microsoft SQL to MySQL (and back): During conversion of Identity (Microsoft SQL) to Autoincrement (MySQL) and back the new IdentitySeed/Initial Autoincrement is taken into consideration. (In version 3.4, only conversion between checkbox Identity and Autoincrement was possible.)
- Model Conversion from MySQL to Microsoft SQL and Oracle: MySQL Enum data type is converted to Char data type, a check constraint for the attribute is created, the parameter is preserved (see the **SQL** tab of the **Check Constraint Properties** dialog).

\* Data types conversion examples - conversion from MySQL 8.0 model to SQL Server 2019 model.

\*\* See the equivalent data types for conversion in the **Settings** menu | **Data Type Conversion Settings**.

## PER - LER Conversion Information

See the following basic information on what is converted during PER to LER and LER to PER conversion and how.

Item to Convert	Notes:	PER - LER Conversion	LER - PER Conversion
<b>Data Types</b>	Similar to PER to PER conversion.	The conversion rules should be defined in the <b>Data Types Conversion Settings</b> dialog.	The conversion rules should be defined in the <b>Data Types Conversion Settings</b> dialog.
<b>Self Relationship</b>	In PER model, only non-identifying self relationship is supported.	Self relationship is converted properly.	Identifying self relationship will change to non-identifying self relationship.
<b>Cardinality</b>	In PER model, cardinality of one side of relationship is 1..n.	Cardinality is converted properly.	E.g. 2..5 cardinality in LER model is converted to 1..5 in PER model.
<b>Parent Key</b>	Parent key can be defined in LER model. In LER model, open the <b>Relationship Properties</b> dialog   <b>General</b> tab   <b>Foreign Unique Identifier</b> box.	Parent key in PER model is different than PK (alternate key, unique attribute or index). -> Parent key defined in PER model is converted to LER model properly.	Selected UI of LER model is converted to PER model (PK and appropriate alternate keys are created).

Item to Convert	Notes:	PER - LER Conversion	LER - PER Conversion
	(In PER model, Parent Key is defined in the <b>Relationship properties</b> dialog   <b>Foreign Keys</b> tab.)		
<b>Foreign Keys</b>	In LER model, keys are not transferred from parent to child entity.	No FKs are displayed in child entity in LER model.	FKs that are not displayed in LER model are visible in child entity in PER model.
<b>Primary Keys</b>		PK (Primary key) in PER -> PUI (Primary unique identifier) in LER	PUI (Primary unique identifier) in LER -> PK (Primary key) in PER
<b>Alternate Keys</b>	You can select alternate key as a parent key.	AK (Alternate key) in PER -> UI (Unique identifier) in LER	UI (Unique identifier) in LER -> AK (Alternate key) in PER
<b>NN versus M Attributes</b>	NN - Not Null in PER model. M - Mandatory in LER model. The values can be displayed in ER diagram.	NN -> M	M -> NN
<b>Inheritance</b>	Inheritance is not supported in PER model.	--	Conversion of inheritance to PER model will be executed by the rules set in the <b>Inheritance</b> dialog   <b>Generation</b> tab.
<b>Valid Values in Attribute</b>	Valid values are supported only in LER model. They can be defined for the following data types: BigInt, Float, Integer, VarChar.	--  (Check constraints from PER model are not converted to LER model.)	Default values in LER model -> Check constraints in PER model.
<b>Defaults for Attributes and Domains</b>		Converted properly.	Converted properly.
<b>Rules for Attributes and Domains</b>	In LER model, attributes and domains can have rules.	Attribute check constraint has rules in PER model. -> Rules for this attribute are	Attribute has rules in LER model. -> Check constraint with this rule for

Item to Convert	Notes:	PER - LER Conversion	LER - PER Conversion
	In PER model, attributes and domains have check constraints and these check constraints can have rules.	converted to LER model.	the attribute is available in PER model.

### LER to PER Conversion - Self-Relationship

If there is a self-relationship in LER model, the entity has two columns, both of the same name (primary key), in converted PER model. Other modifications are necessary.

Possible solution: You can define a name for the propagated attributes in LER model before the conversion. Open the **Attribute Properties** dialog | **General** tab | enter the name to the **Propagated Name** box.

If this box is empty, Toad Data Modeler will behave standardly (two columns of the same name in PER model).

## Convert Model

Toad Data Modeler allows you to convert:

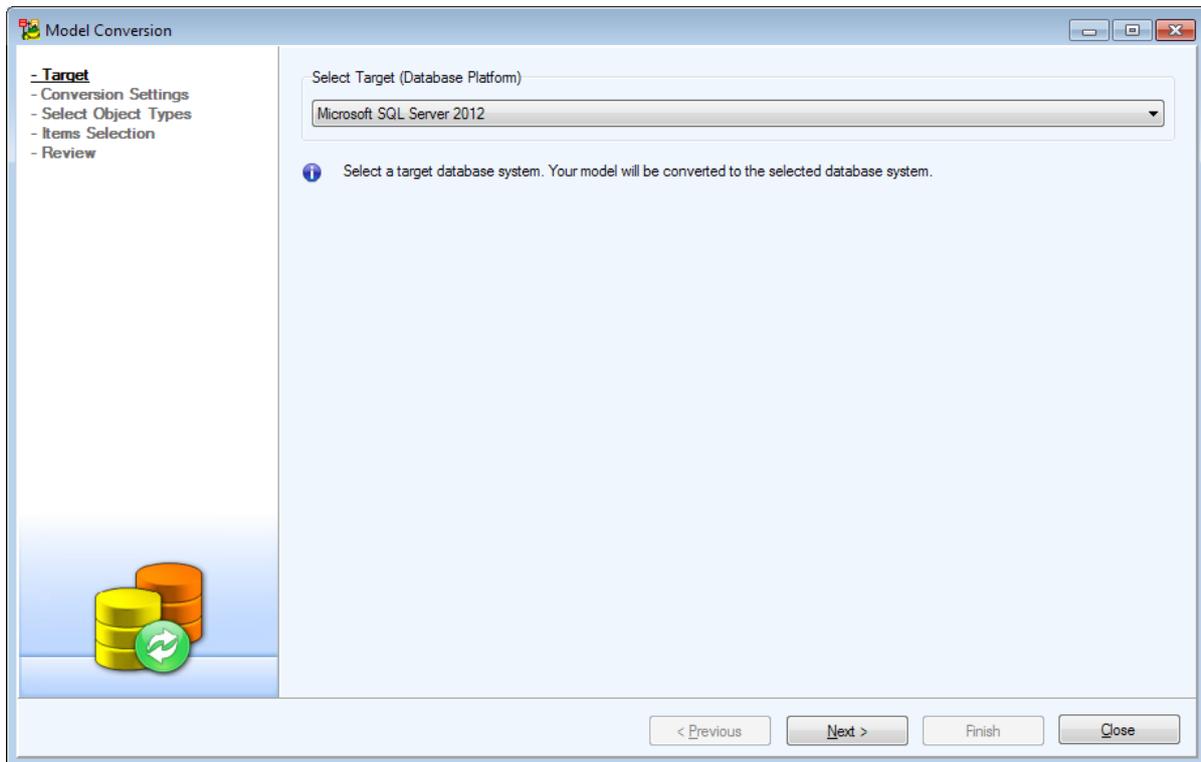
- Physical models from one database platform to another (e.g. Oracle 10g model can be converted to SQL Server 2008 model)
- Physical model to Logical model
- Logical model to Physical model

The conversion can be executed via:

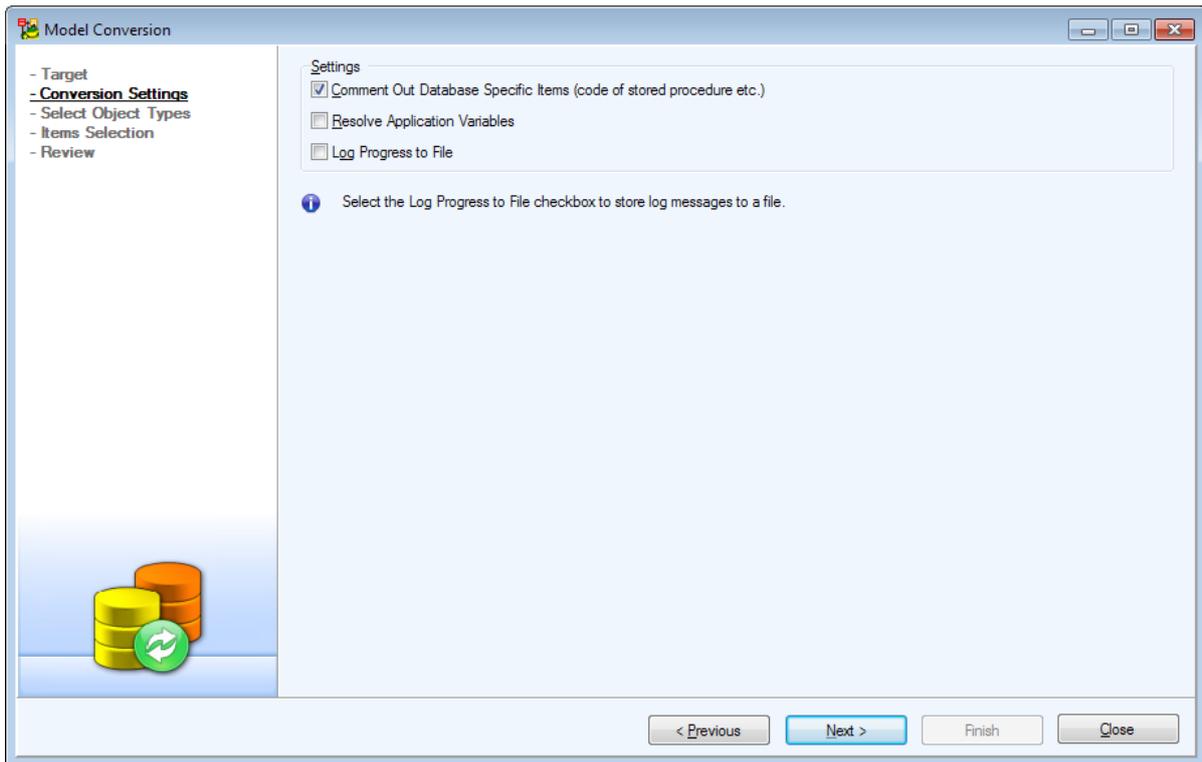
- **Model Actions** | **Convert Model** or **Model Menu** | **Convert** | **Run**
- **Simple Model Conversion** in **Model Menu** | **Convert**
  - [Simple Model Conversion](#)

## How to Convert Model

### Physical to Physical

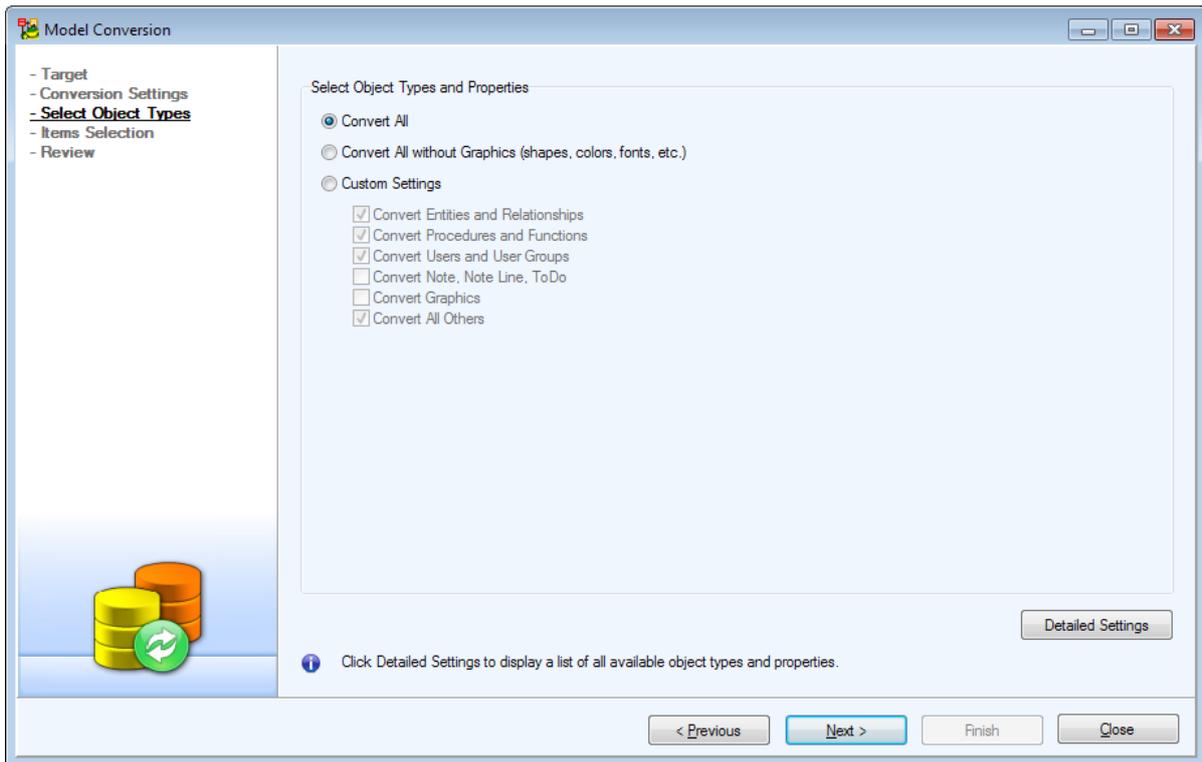


- Select your desired database platform.

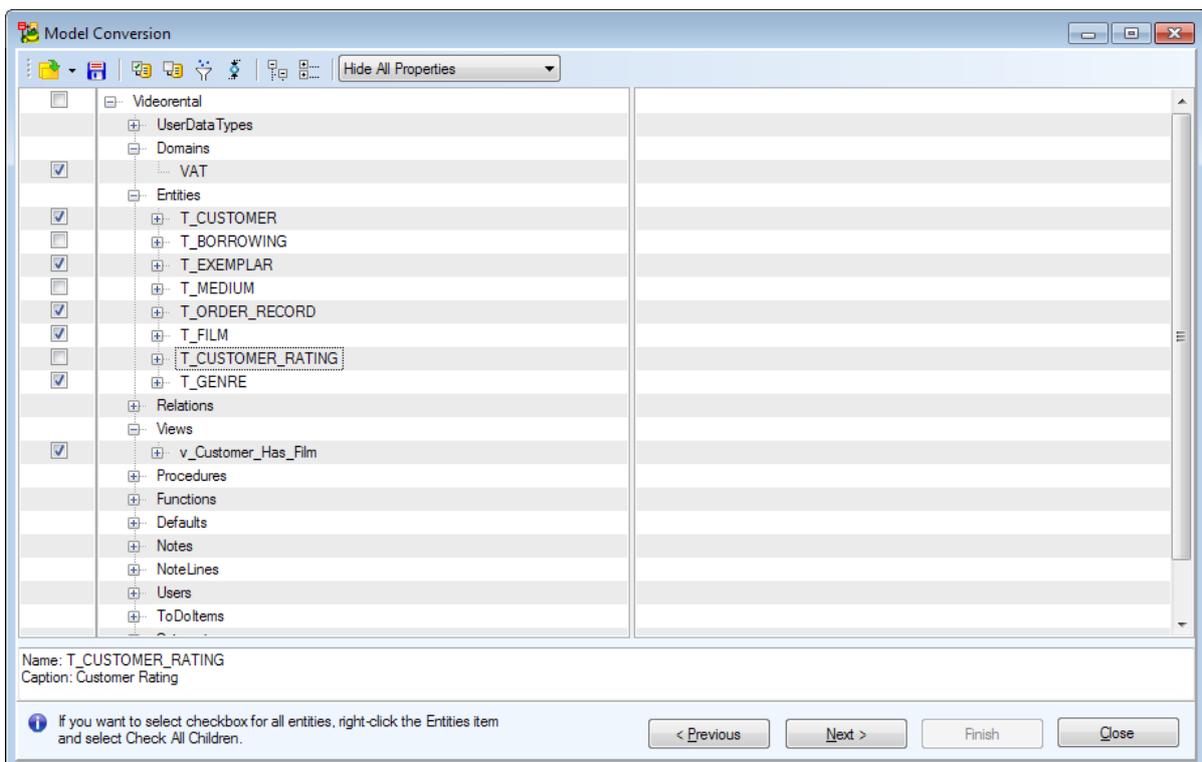


- Pay attention to the Conversion Settings dialog.

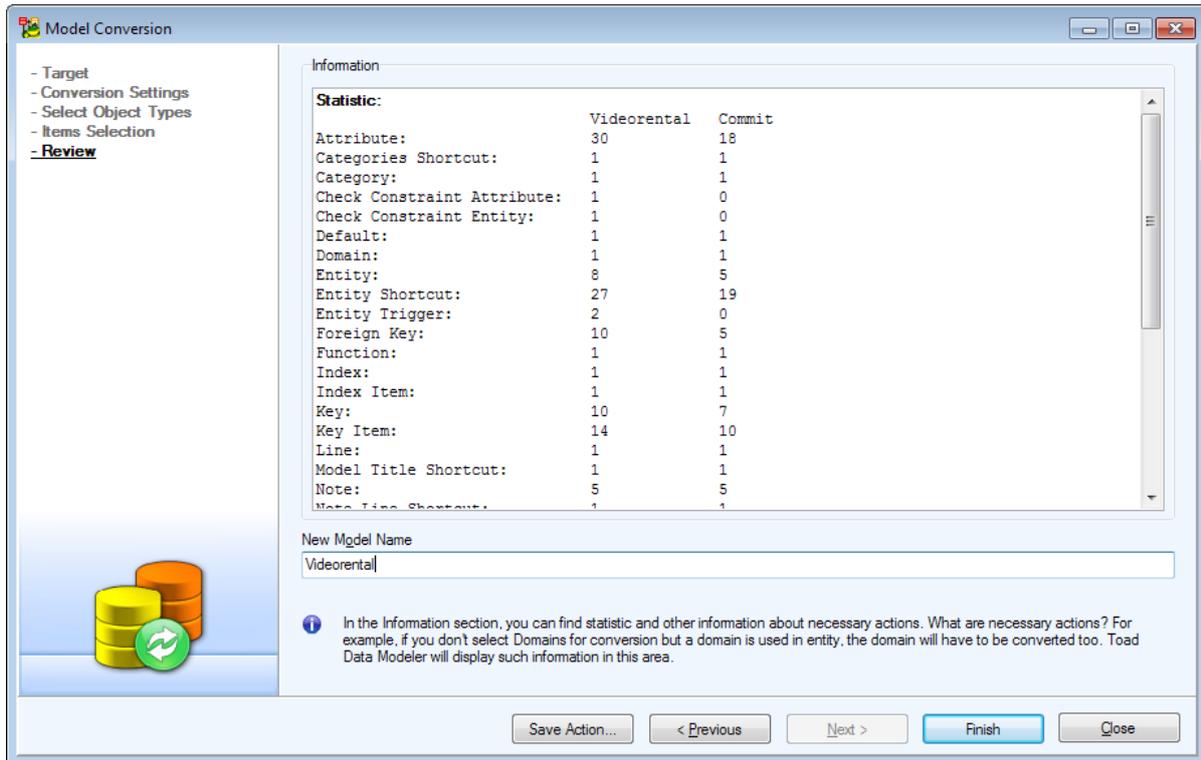
**i** Note: Converting between different platforms means that some items, which contain SQL code, may not be converted correctly (e.g. After Scripts, Before Scripts, Procedures...). In this case, it is recommended to check the **Comment Out Database Specific Items** checkbox. This way, SQL code, that cannot be successfully converted, will be commented out. You can review these parts of code later and fix them on your own.



- Choose what object types will be converted. You can access full list of Objects and Properties by clicking on **Detailed Settings**.



- Check items you want to convert to another model. For easier item management use buttons located on the top.

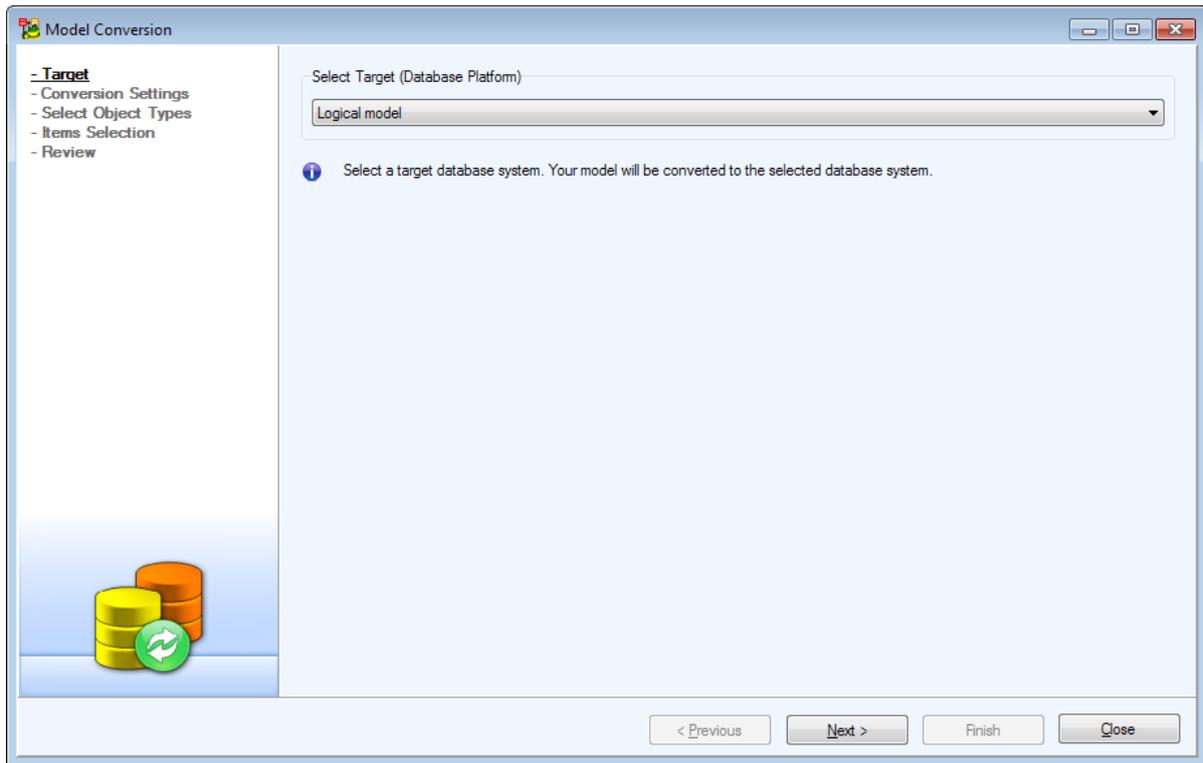


- A review dialog shows with some statistics about how many and what kinds of objects will be converted. Once you're done reviewing, click on Finish.

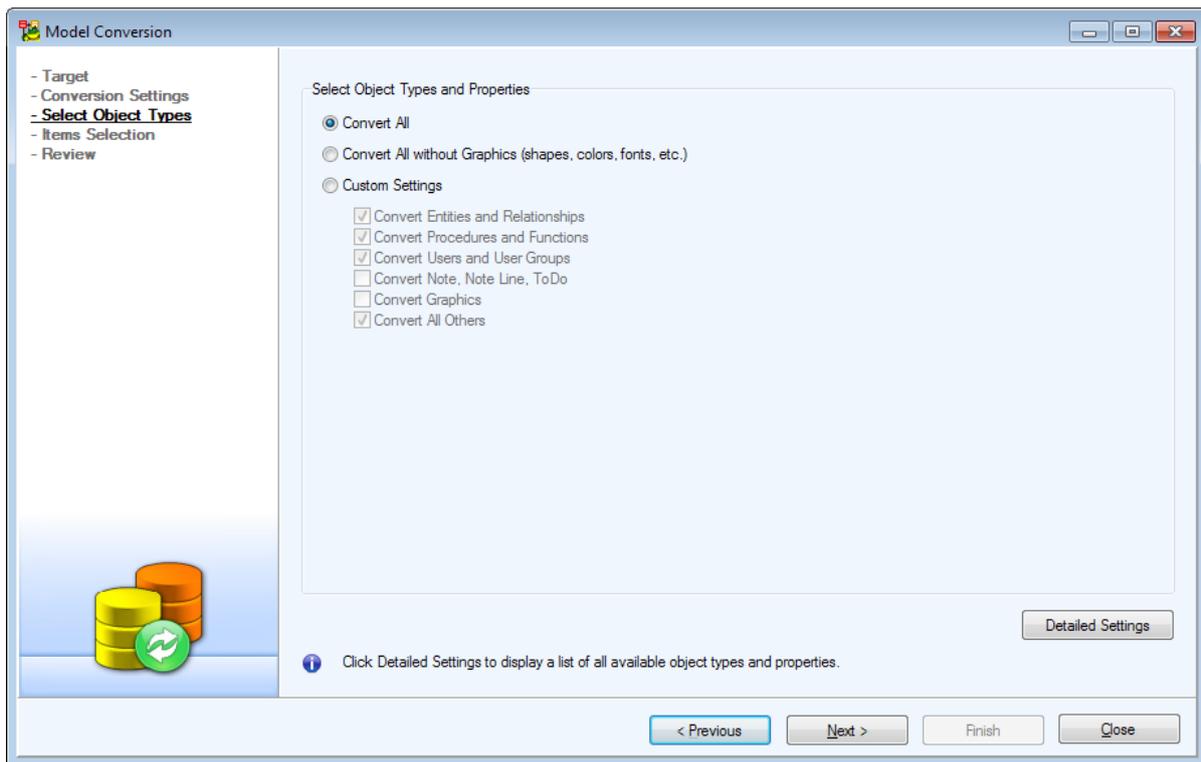
**i** TIP: If you plan to do this action again in the future, you might want to click the **Save Action** button to save this **Action Definition**. See **Model Actions** for more information.

- The conversion process occurs and your model is now converted to another database platform.

# Physical to Logical

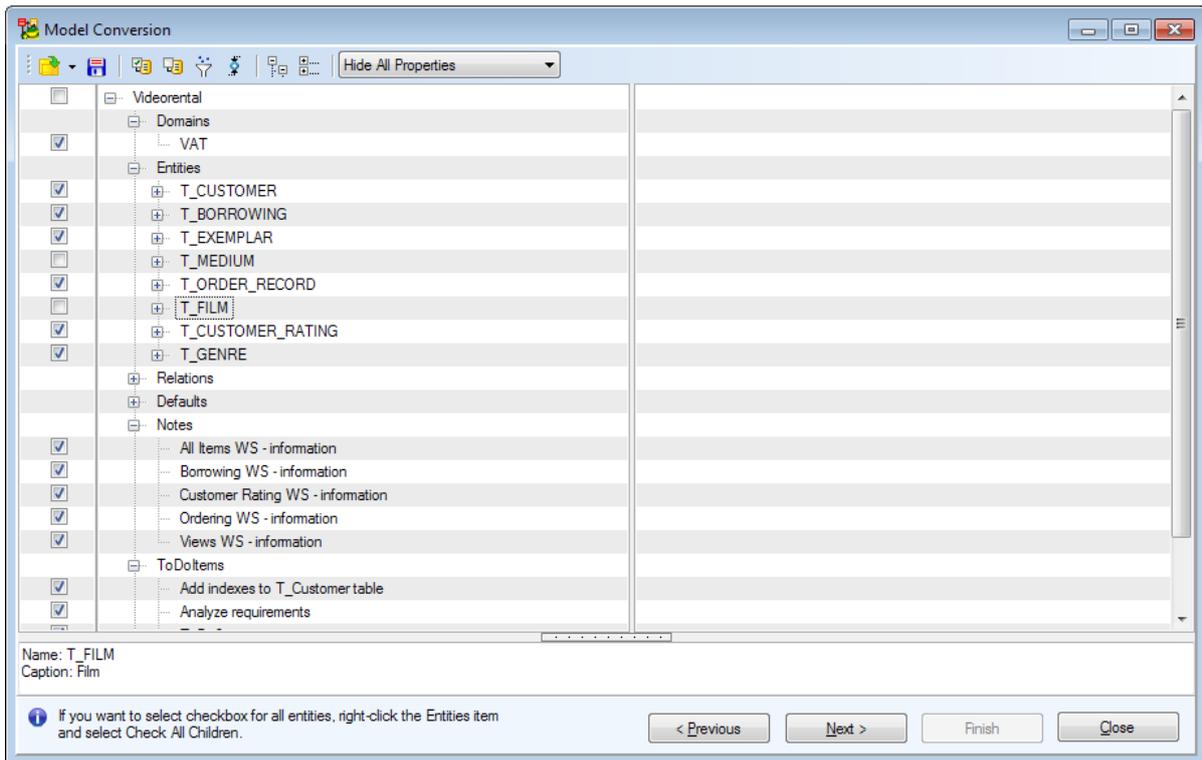


- Select **Logical model** from the list of available database platforms.

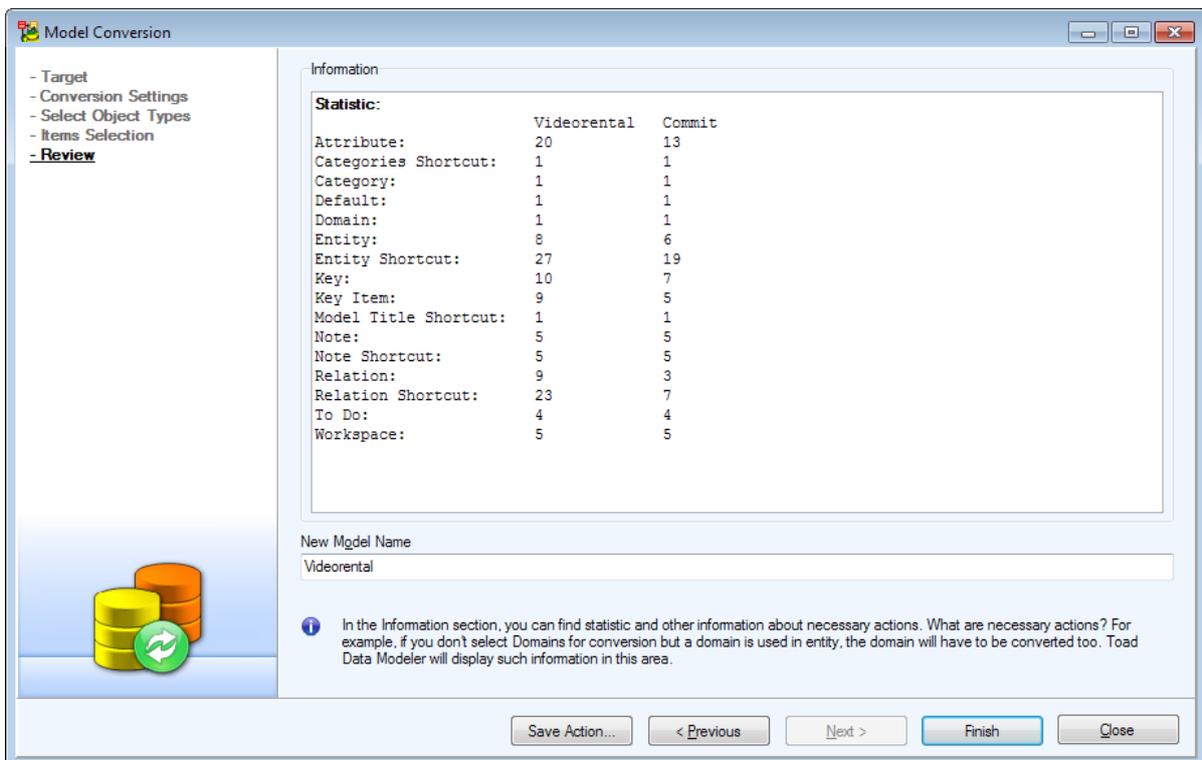


- Pay attention to the Conversion Settings dialog.

**i** Note: Since Logical Model doesn't have any objects which contain SQL, all your procedures, scripts and functions will be lost.



- Check items you want to convert to another model. For easier item management, use buttons located on the top.



- Review the statistic and when you're done, click the **Finish** button.
- After a short while your Physical model will be converted to Logical model.

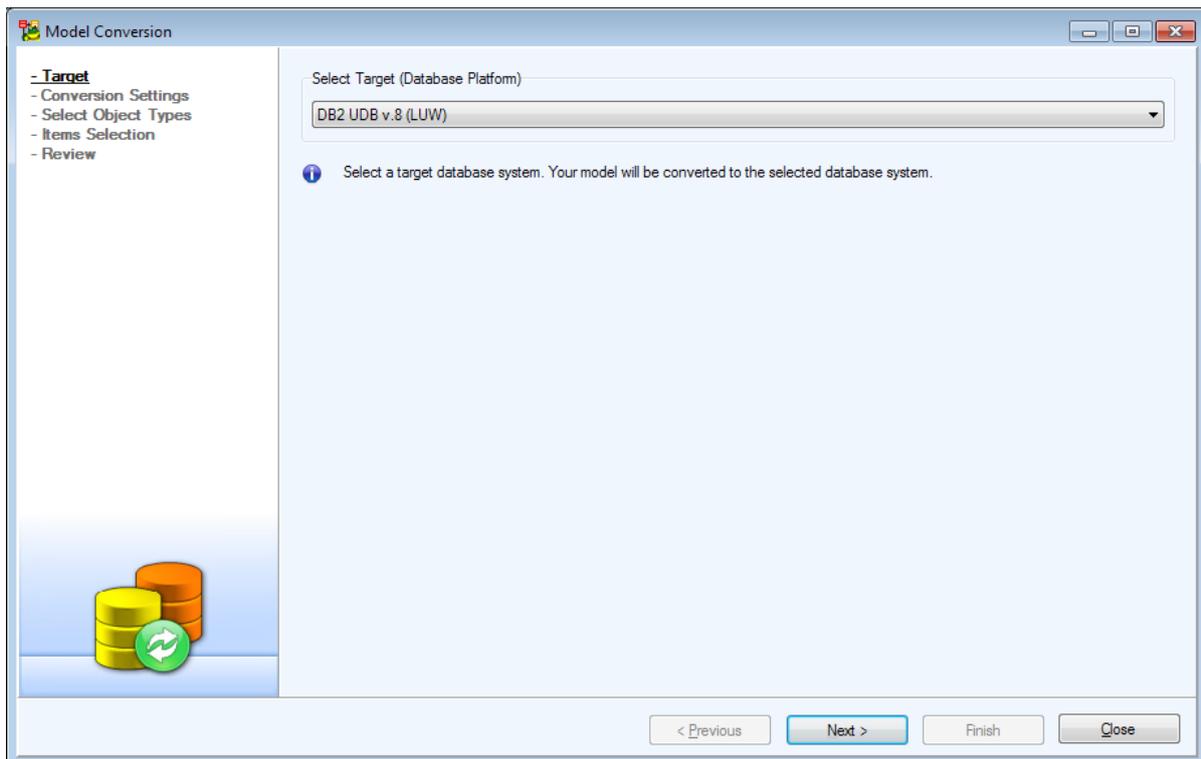
**i** TIP: If you plan to do this action again in the future, you might want to click the **Save Action** button to save this **Action Definition**. See **Model Actions** for more information.

## Logical to Physical

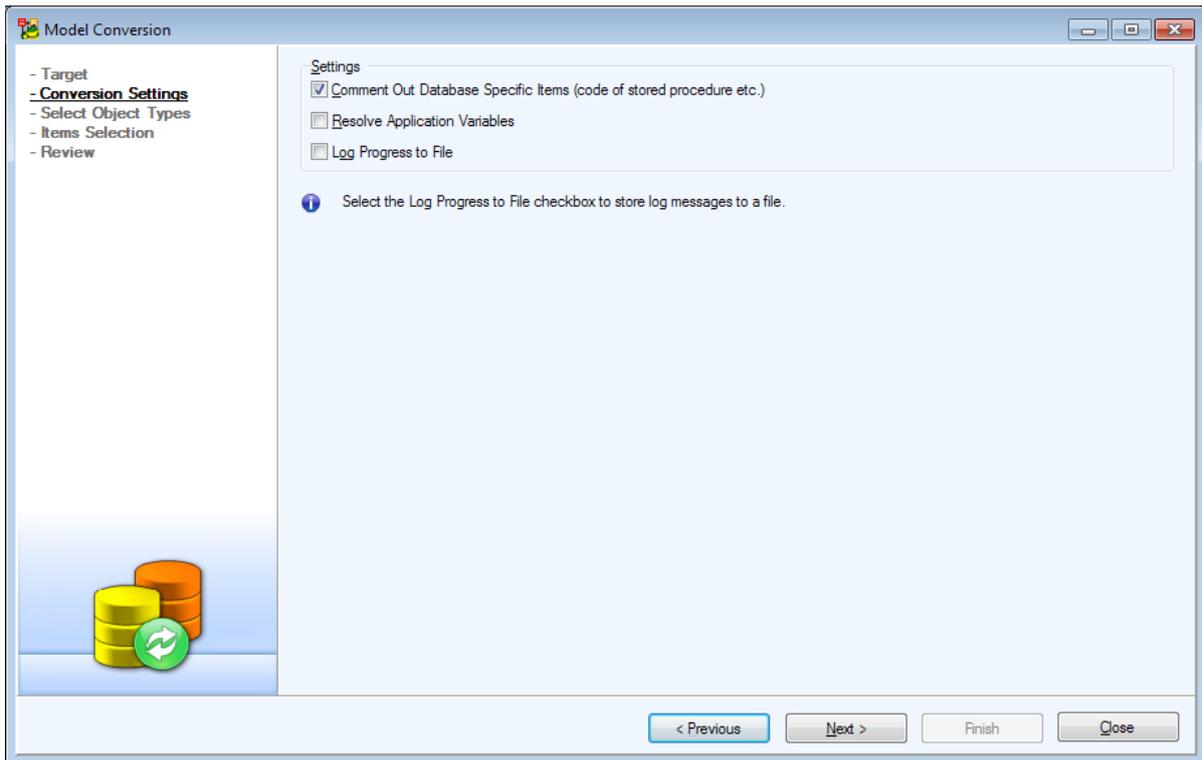
**i** Note:

Before you convert your Logical model to Physical model, you should be aware of the following:

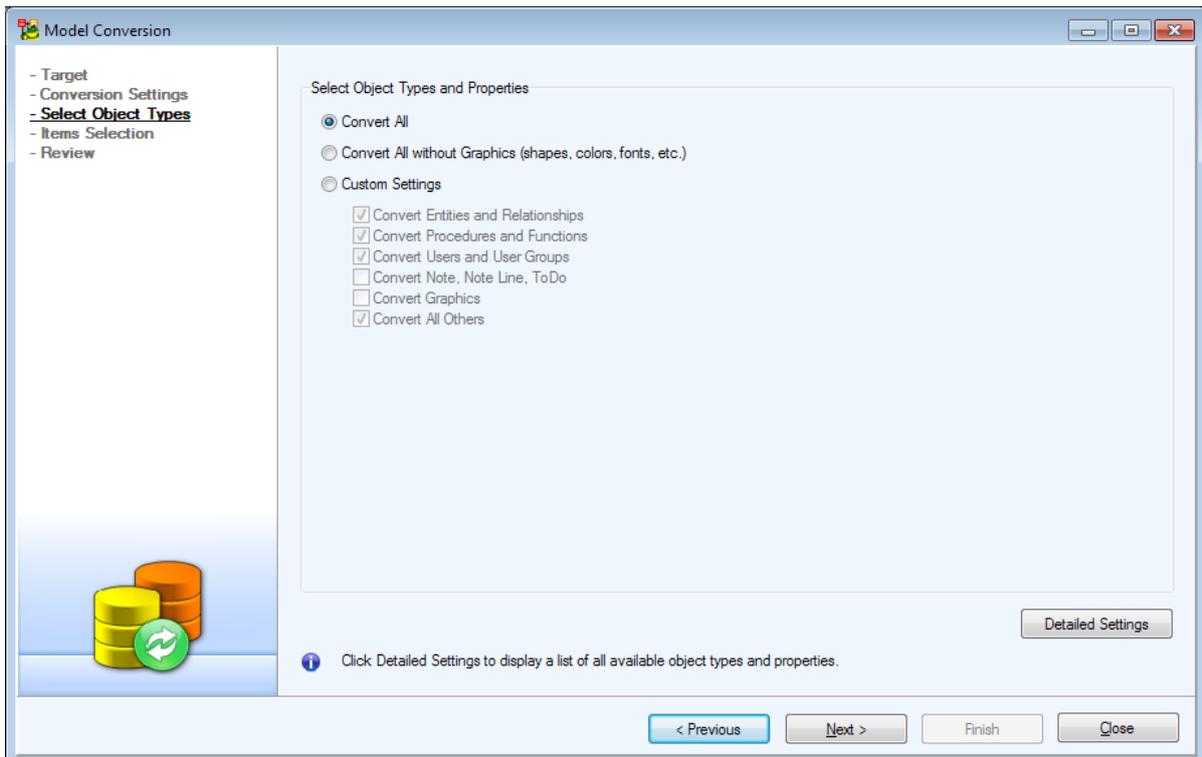
- Physical model supports only non-identifying self relationship.
- Inheritance is not supported in PER model. Toad Data Modeler solves this by converting Inheritance object into either **Single Table**, **N-1 Tables** or **N Tables**. See **Inheritance**.
- Keys in LER models do not migrate.
- You can select a linking method in LER model.
- M:N relationships are supported in both models.
- Before you start the conversion, you can set up the conversion rules in the **Data Type Conversion Settings** dialog. This option is available only if Expert Mode is enabled.
- Cycled relationships will be ignored during LER to PER conversion and will not be converted. A message informing you about this will be displayed in Message Explorer Log.



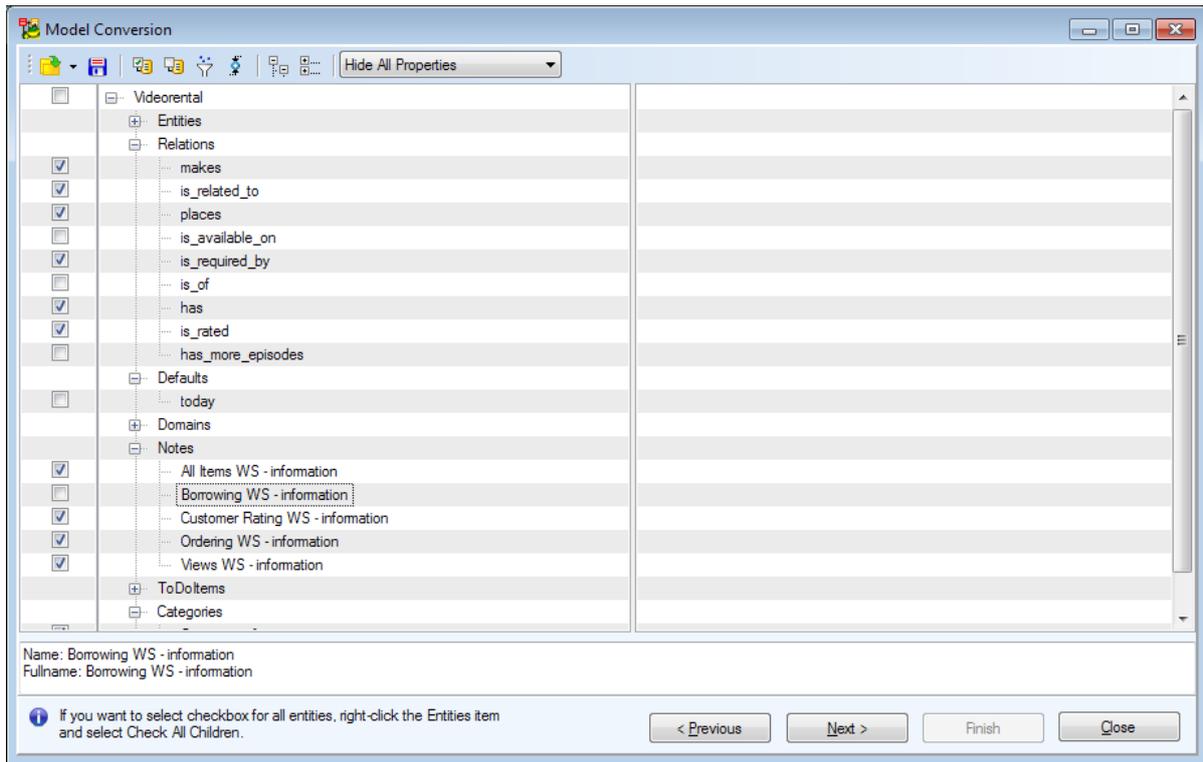
- Select your desired database platform.



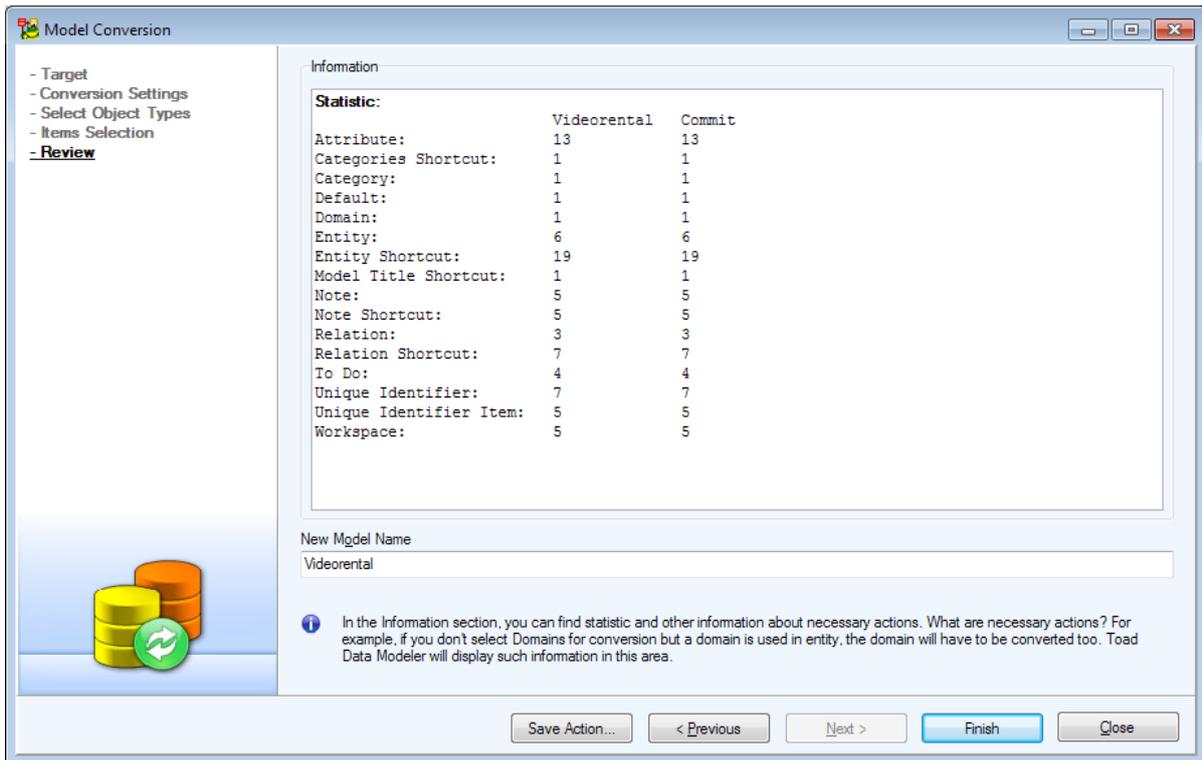
- Change any of the conversion settings, if necessary.



- Choose what object types will be converted. You can access full list of Objects and Properties by clicking on **Detailed Settings**.



- Check items you want to convert to another model. For easier item management use buttons located on the top.



- Review the statistic and when you're done, click the **Finish** button.
- After a short while your Logical model will be converted to Physical model of your desired database platform.

**i** TIP: If you plan to do this action again in the future, you might want to click the **Save Action** button to save this **Action Definition**. See [Model Actions](#) for more information.

## Selection Tree Overview

Option	Description
	Import selection from a saved file.
	Save selection to a file.
	Checks all items.
	Unchecks all items.
	Opens the <b>Wildcard Dialog</b> where you can define settings for bulk selection/deselection of the <b>Action</b> box of the items listed on page <b>Select Items</b> .
	<p>Refresh Necessitated Items</p> <p>Explanation: Some objects are related together (e.g. entity and domain, entity and relationship). Let's say you uncheck a Domain in <b>Select Object Types</b> dialog. However you keep an Attribute of the Domain type checked for conversion. In the next screen the Domain will be selected for conversion (and highlighted in gray), even if you don't want it to. This is because of its relationship with the Attribute, which cannot exist without the Domain.</p> <p>Now, if you uncheck the Attribute, the Domain will still be checked for conversion. This is where you use this button. It runs through all checked objects and removes the Domain highlighted in gray since the Attribute is no longer checked. That means the Domain is no longer necessary, since it has no relationships with currently checked objects and you unchecked it in <b>Select Object Types</b> dialog.</p>
	Expand All.
	Collapse All.



**Right-click an item to see the following options:**

Option	Description
Expand All Children	Expands all sub-items of the selected item.
Collapse All Children	Collapses all sub-items of the selected item.

# Simple Model Conversion

This feature allows you to convert your logical models to physical models and your physical models to models of another database system very quickly.

## Scenario

You want to convert your Oracle 10g model to Microsoft SQL Server 2005.

1. Open your Oracle 10g model.
2. Select **File | Synchronization | Simple Model Conversion**.
3. Define the settings in the **Conversion** dialog.

Option	Description
To Database	Select a target database system - Microsoft SQL Server 2005.
New Model Name	Define a name for the converted model.
More>>	Click this button to see and modify Object Types and Properties for the conversion. No modification is necessary. <a href="#">Object Types and Properties - OTPs</a>
Close after Conversion	Select it to close the <b>Conversion</b> dialog after the process is finished.
Convert	Executes the process of conversion.
Close	Closes the <b>Conversion</b> dialog.

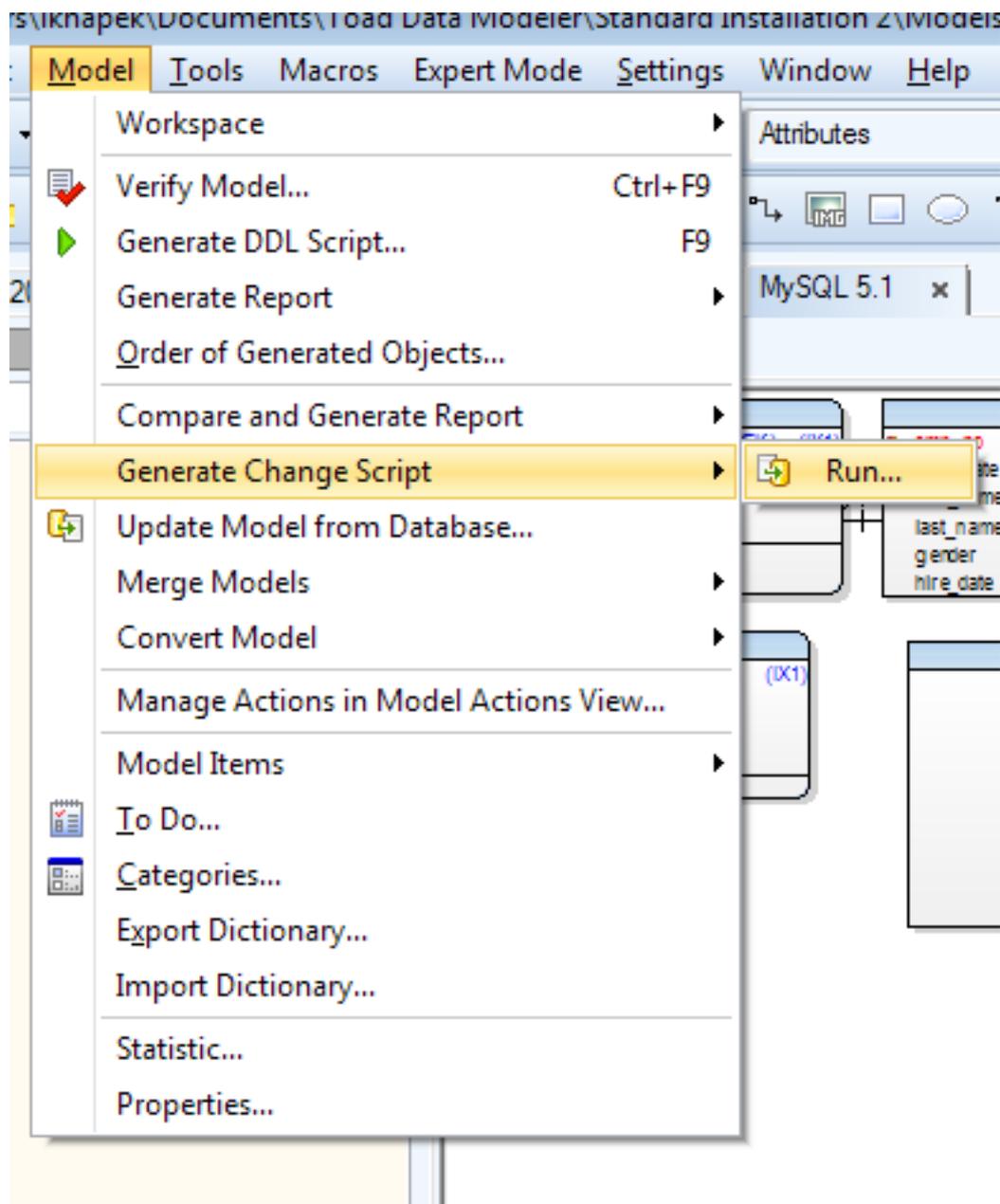
4. Click **Convert**.

# Compare Models, Generate Change Report

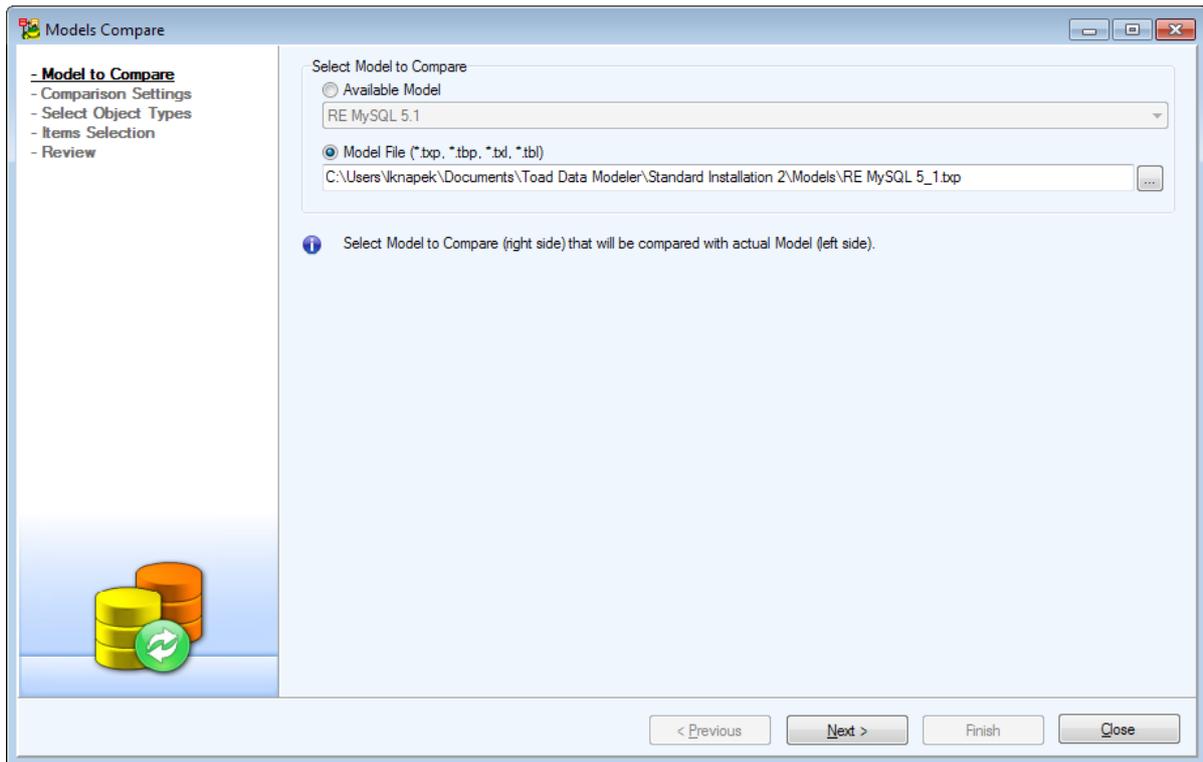
Toad Data Modeler allows you to view differences between two models and for physical models you can generate change reports in HTML, RTF and PDF formats. Change reports are not available in logical models.

You can compare your models and generate the change report via the **Model Actions**.

## How to Compare Models and Generate Change Reports

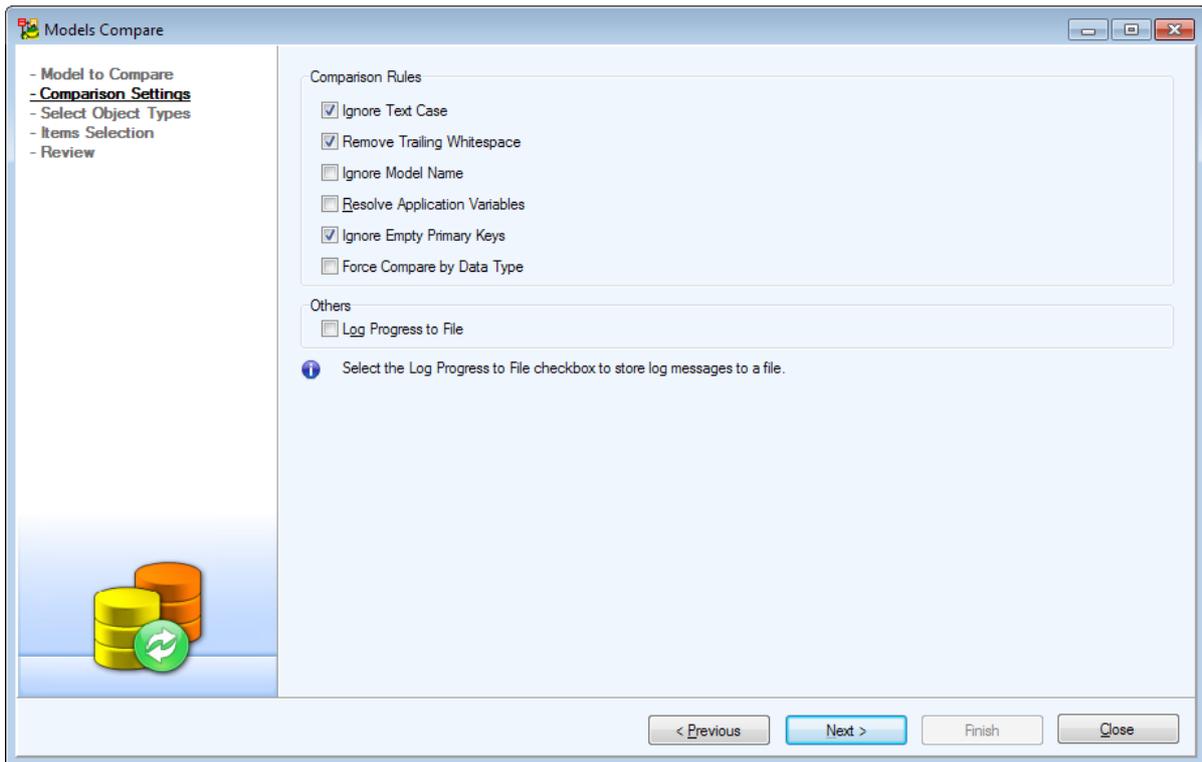


- Click on **Compare and Generate Report | Run** in **Model Menu** or Run **Compare and Generate Report** Action in **Model Actions**.

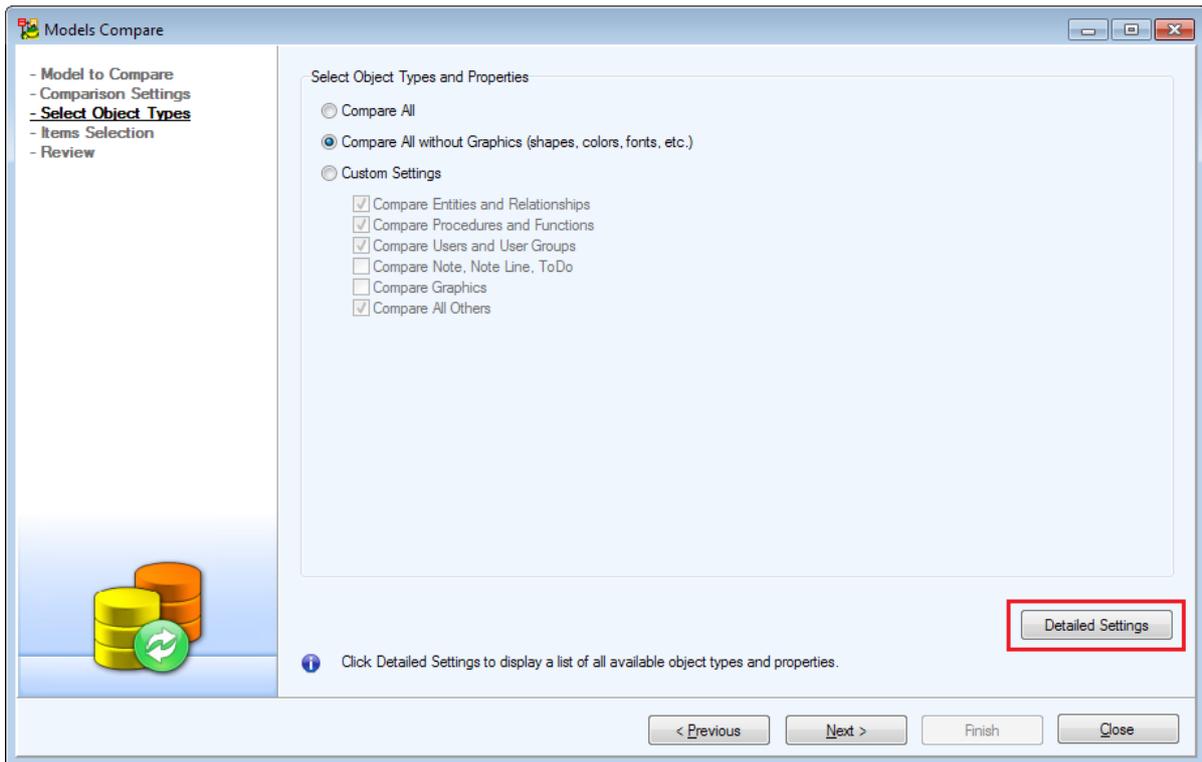


- Select another model for comparison. It can be either a model opened in Toad Data Modeler or a model loaded from **Model File**.

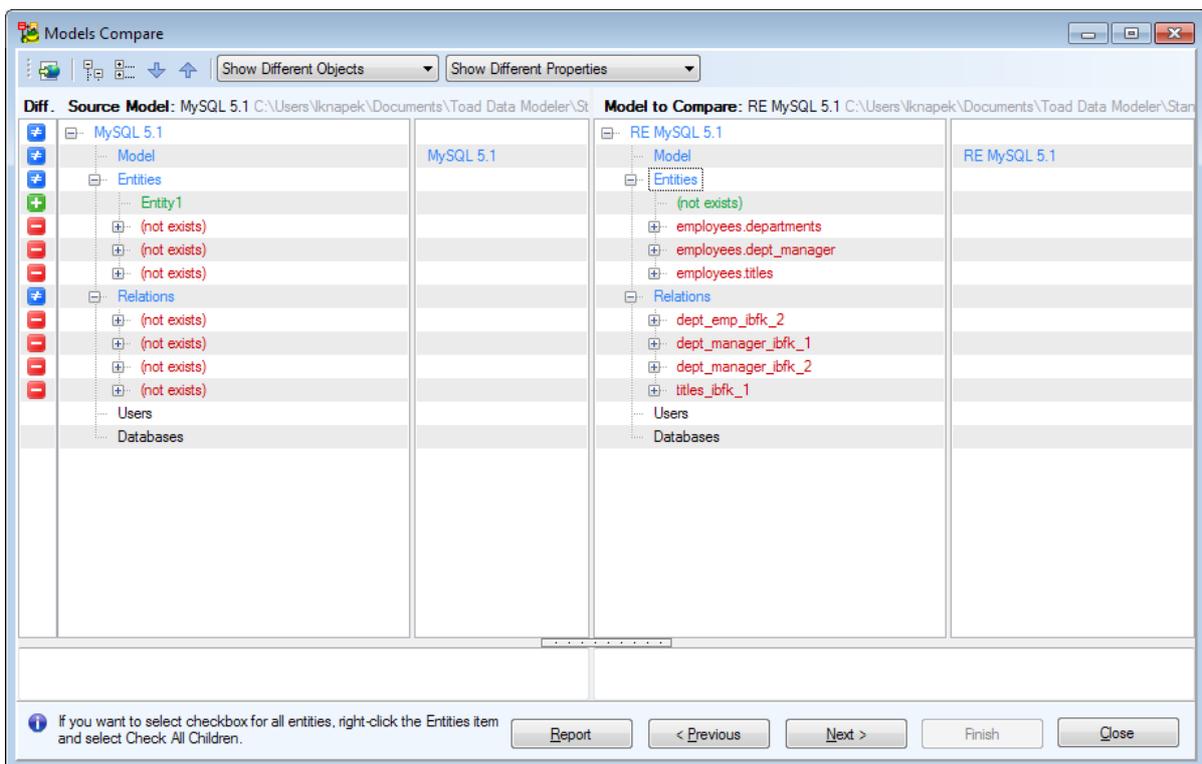
**i** **Note:** The selected model has to be of the same database platform and version. You cannot compare DB2 10.5 and MySQL 8.0 models.



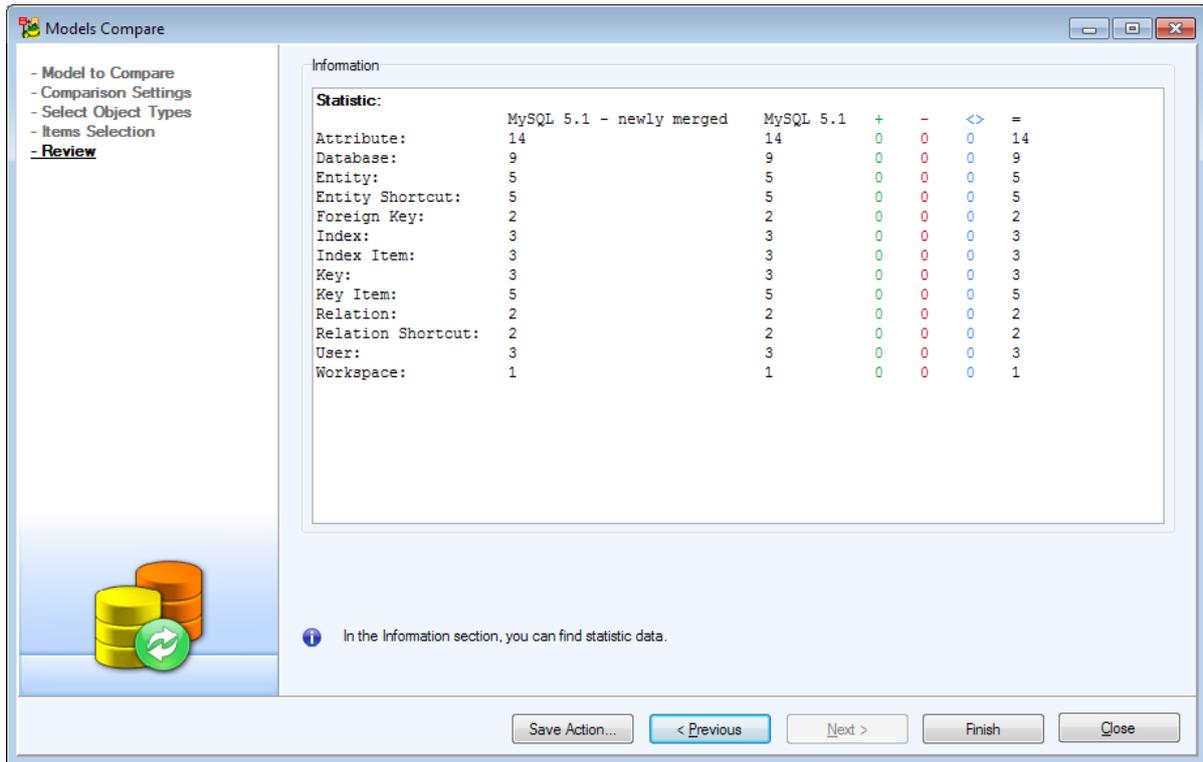
- Check **Resolve Application Variables** to resolve variables during generation including variables in names
- The next dialog presents you with a couple of setting related to **Model Comparison**. When you're done editing the options, click on Next.



- Now you have to choose what types of objects will be included in the comparison. If you wish to select objects in more detail, click on **Detailed Settings**.

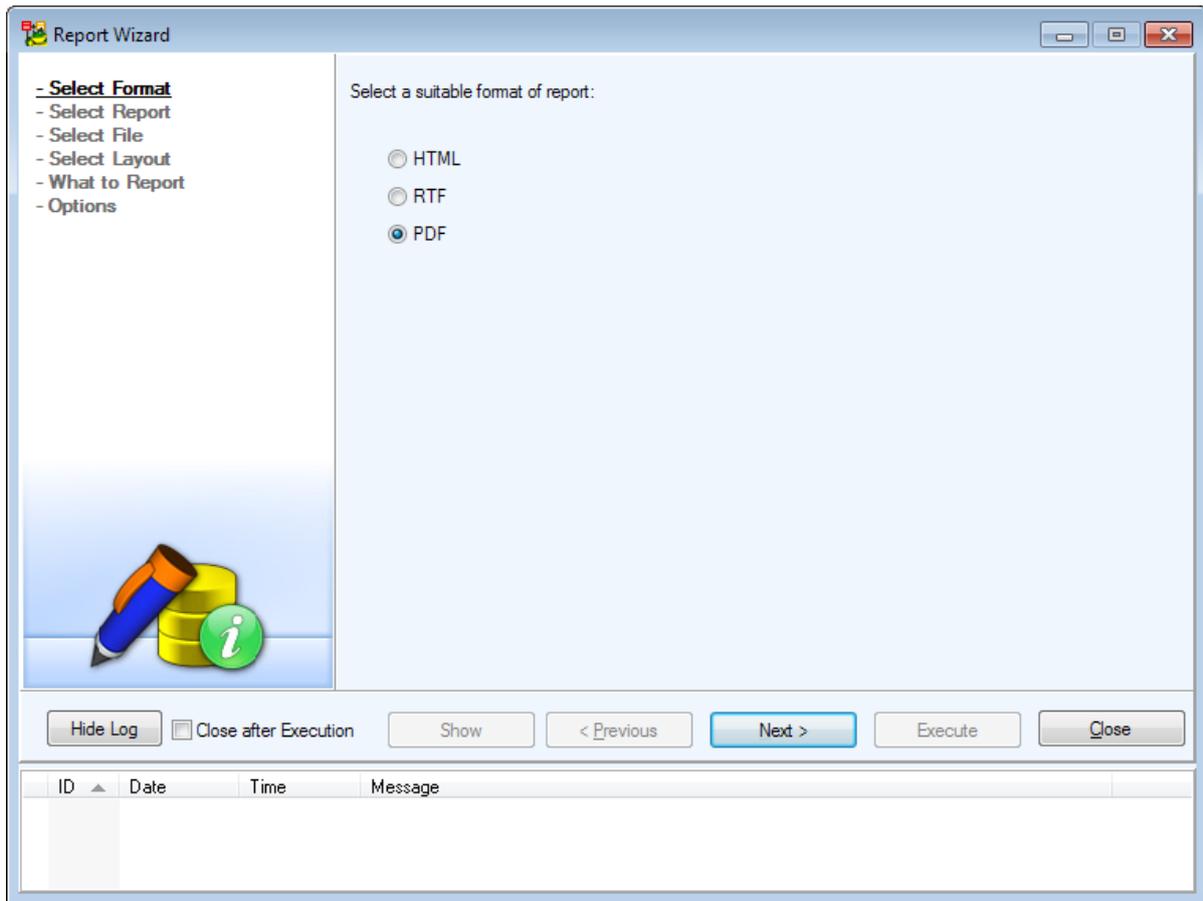


- You are now presented with Compare Tree dialog displaying all differences between the two given models. Now you have the option to generate a report by clicking on **Report** button.



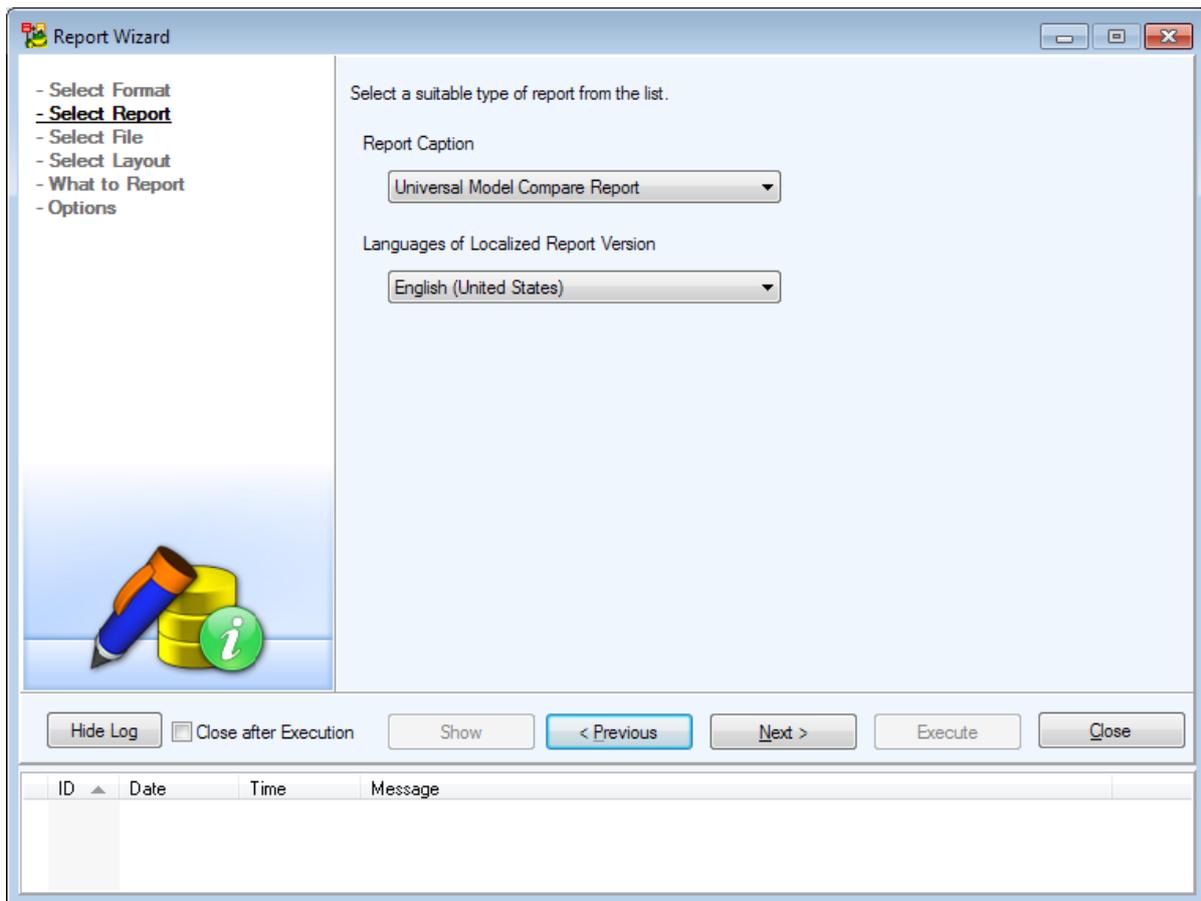
- Check the Review screen which shows you the final statistics and close Model Compare by clicking the **Finish** button.

**i** TIP: If you plan to do this action again in the future, you might want to click the **Save Action** button to save this **Action Definition**. See **Model Actions** for more information.



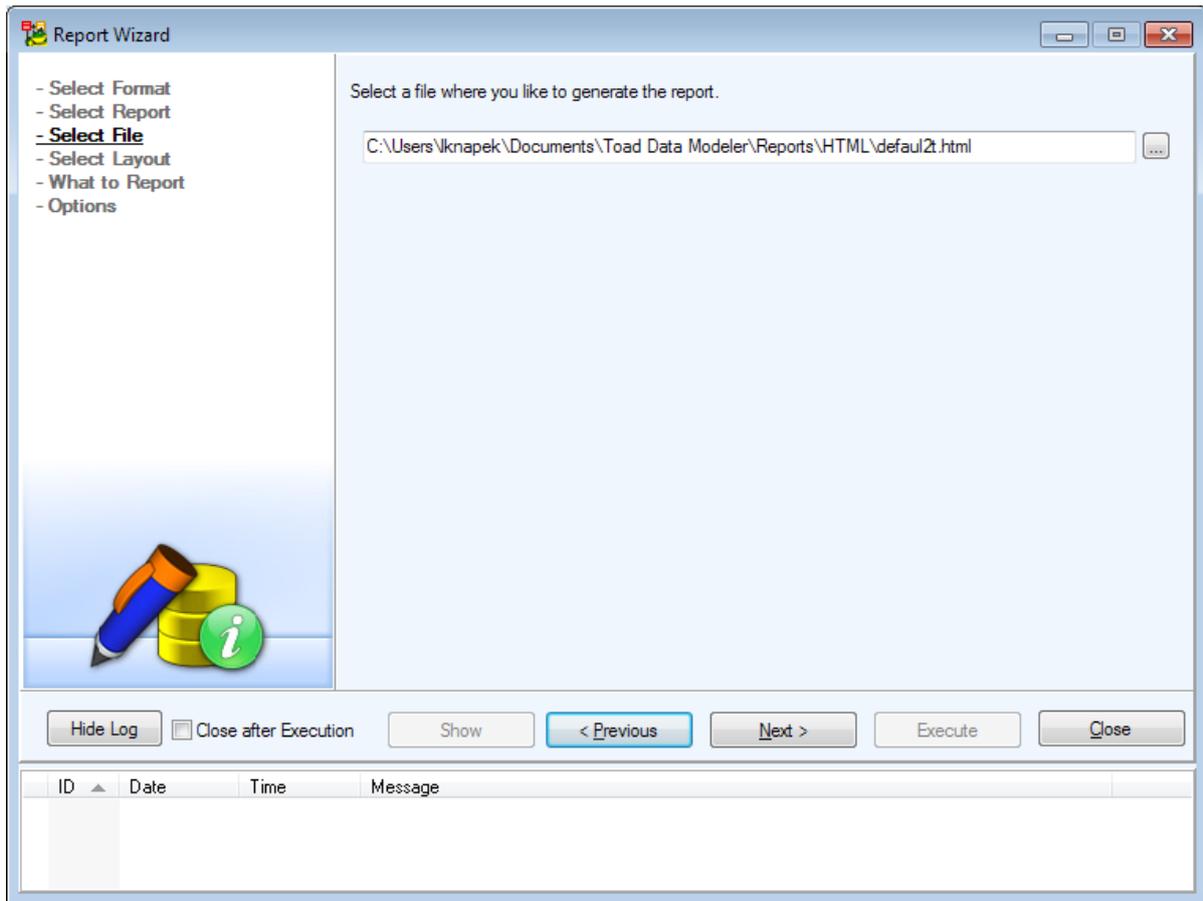
- Select Format of the generated report.
- There are two options available during the entire process of generating a report. **Hide Log** hides the bottom part of the dialogs which otherwise displays information related to the generation. Checking **Close after Execution** closes the dialog window after you're done creating your report.

## Generating HTML Reports

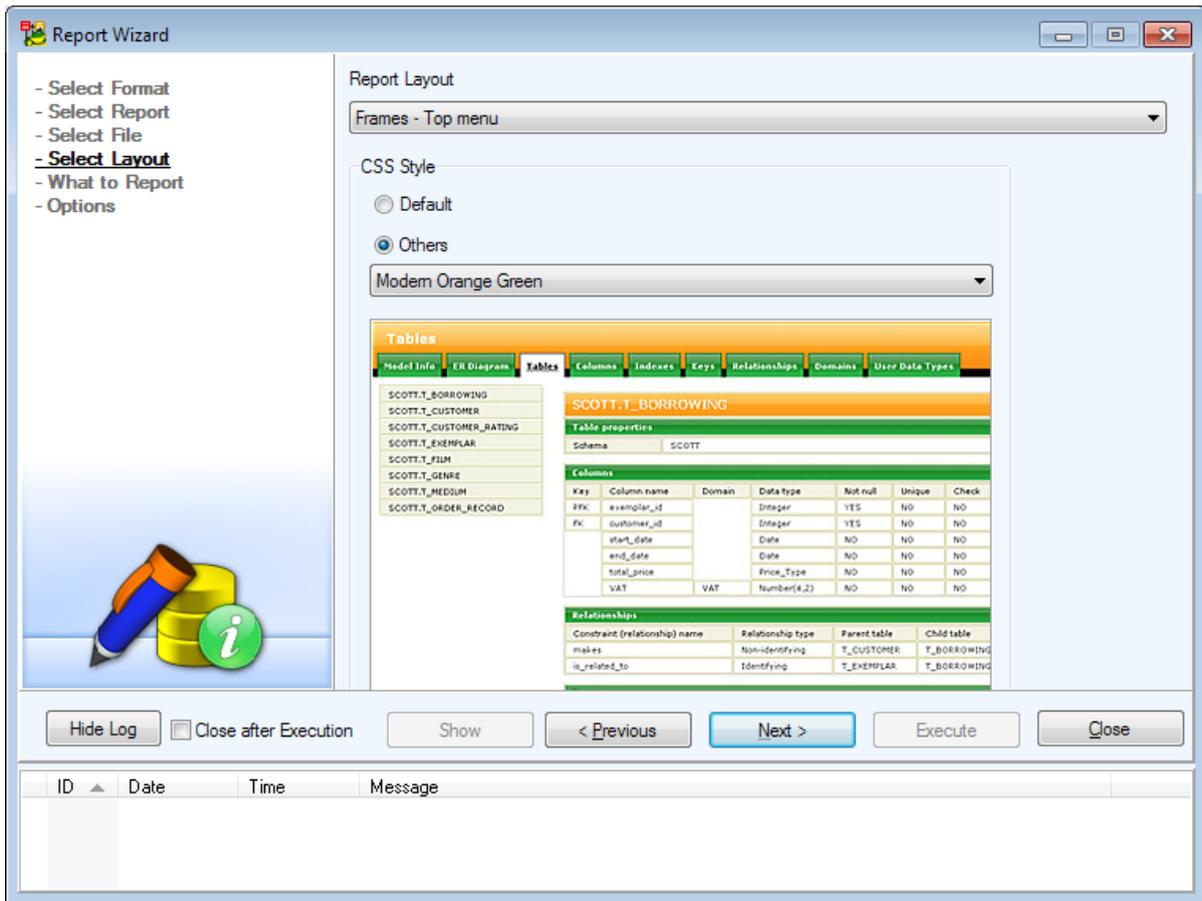


- After you select HTML from previous dialog and click on Next, you have the option to change **Report Caption** and **Language** of the report.

**i** nOTE: You can generate Reports in many languages provided that you have the appropriate Dictionaries installed. For more information see [Dictionaries](#).



- You are able to define your own path where the generated report will be saved, if you need to.



- In this dialog there are several options for customizing the look of the report.

### Select Layout Page

### Description

Report Layout

- Frameless (for Internet Explorer 7 and FireFox 2 users. Reports with Frameless layout is not displayed in older browsers correctly.)
- Frames - Top menu
- Frames - Left menu

**i** Note: For large models, *Frames - top menu* or *Frames - left menu* options are recommended. (Frameless report layout is not recommended as it uses Java script that goes through all objects, which takes too much time if your model is large.)

CSS Style

Available styles for HTML report:

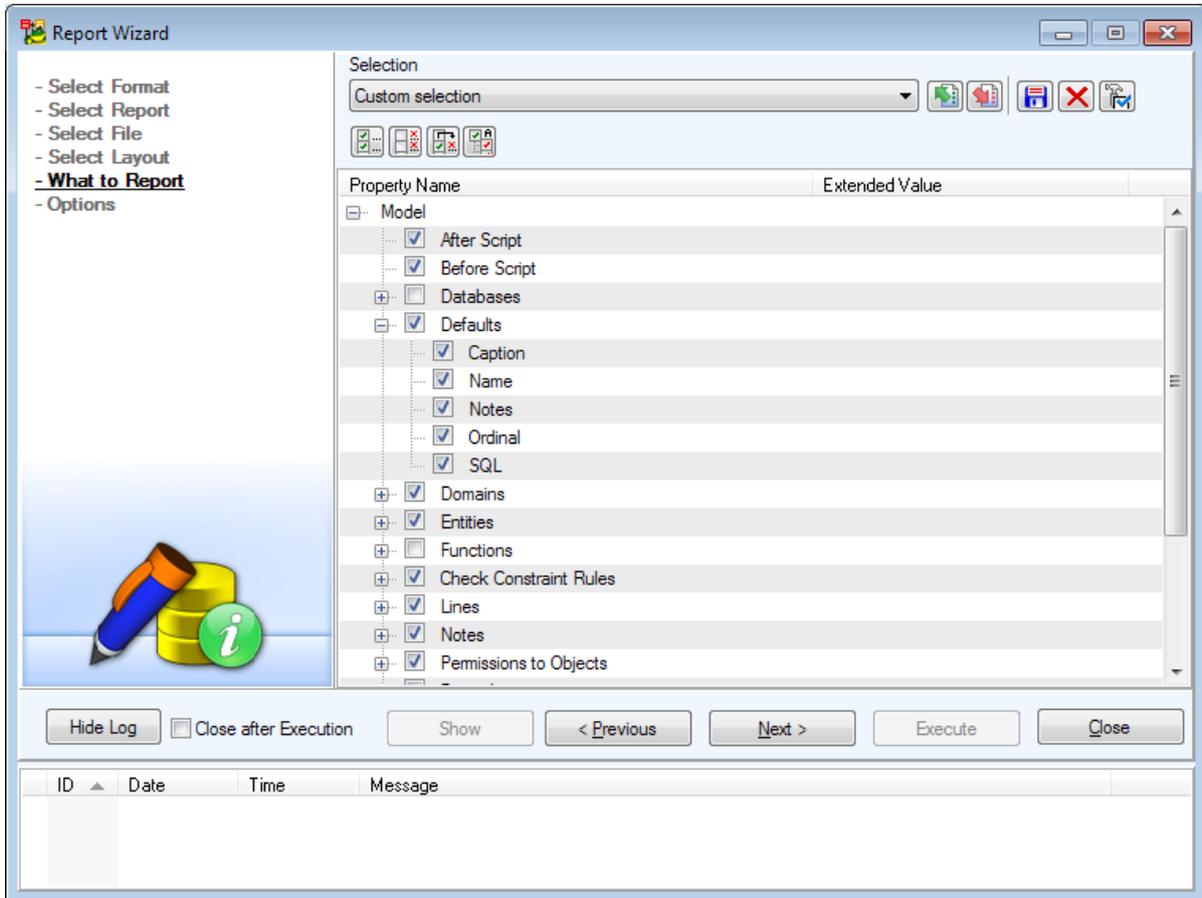
- Default

**Select Layout Page**

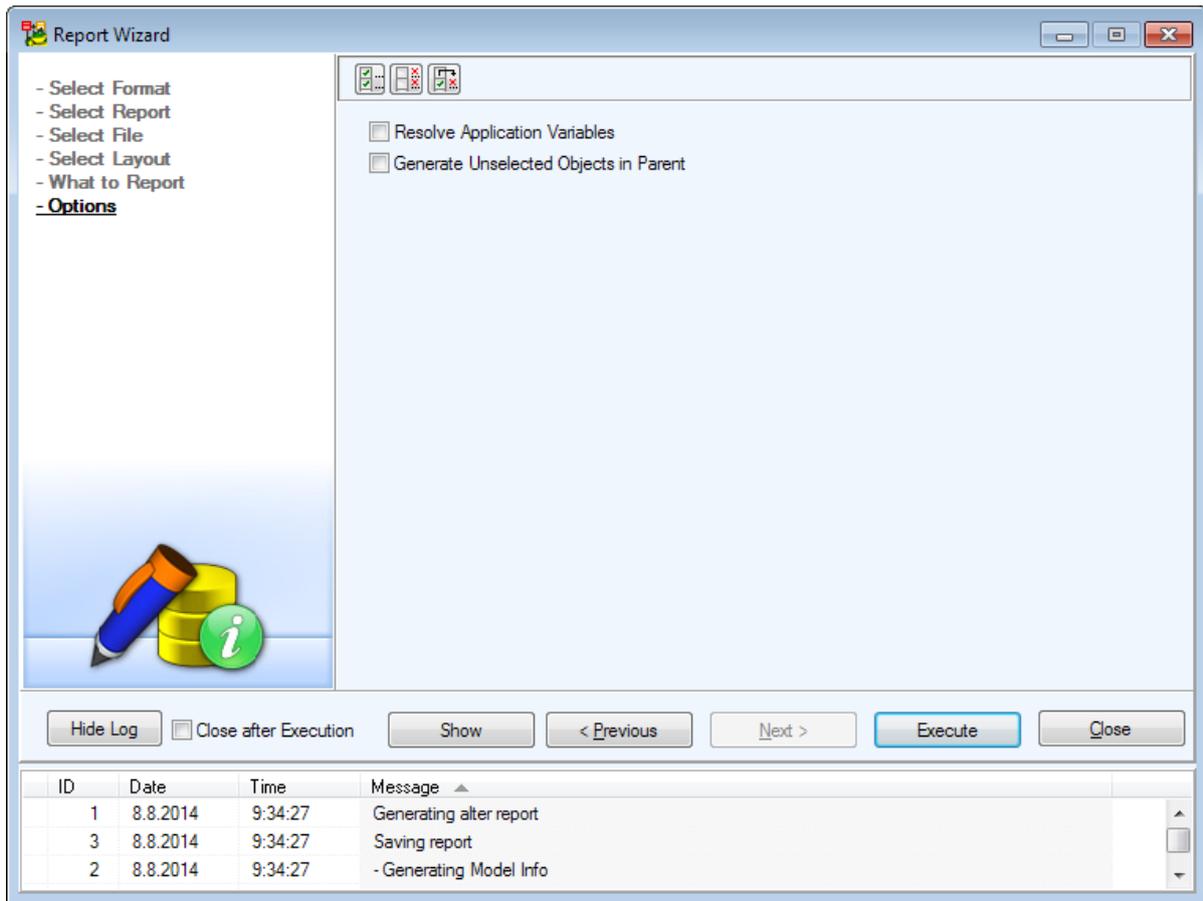
**Description**

- Others (select the most convenient for you)

**i** TIP: You can also create your own report styles using **CSS**. To integrate a custom style into Toad Data Modeler, you have to create a **.txs** file referencing the style resources and save it to **Style folder**.

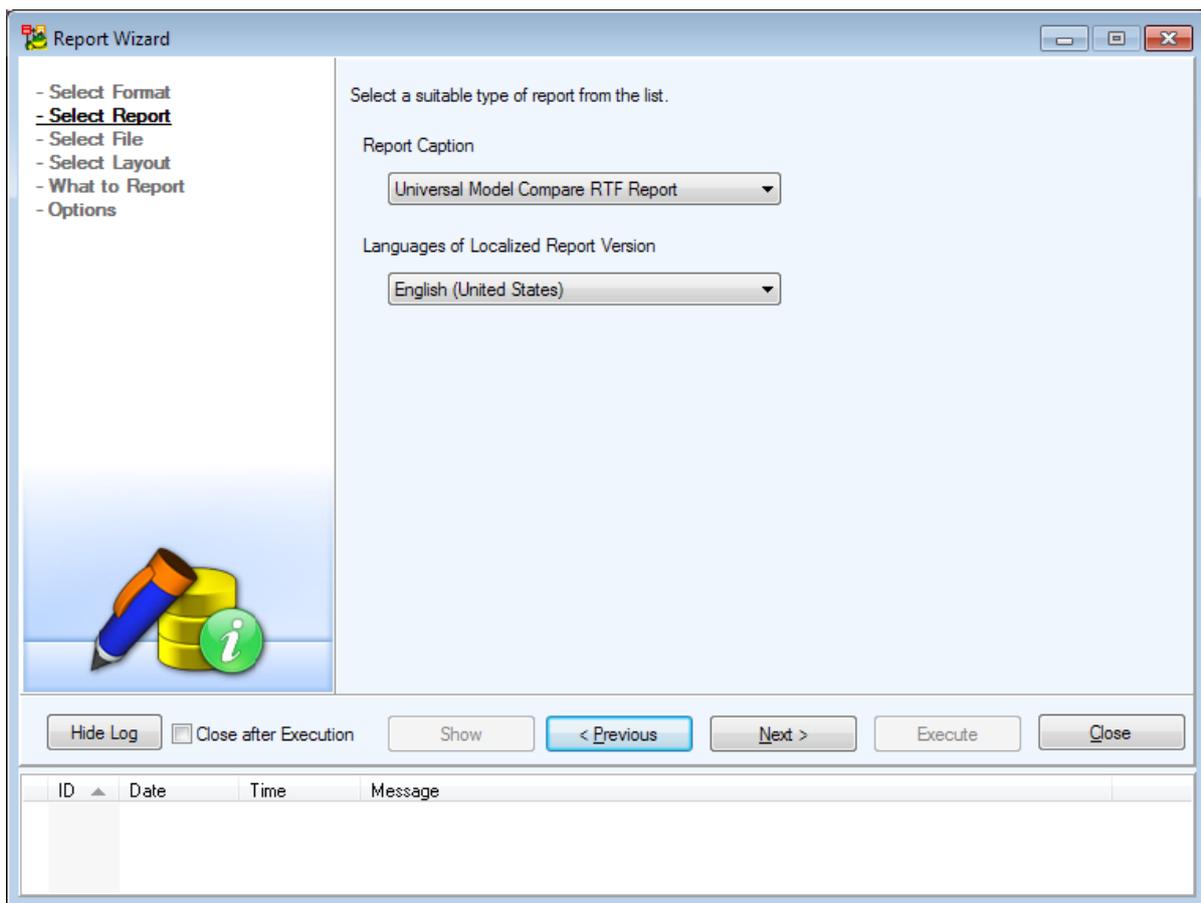


- Now is the time for you to decide what kind of information should the generated report contain. If you want to get a complete report, check all boxes. For easier management use buttons located on the top.



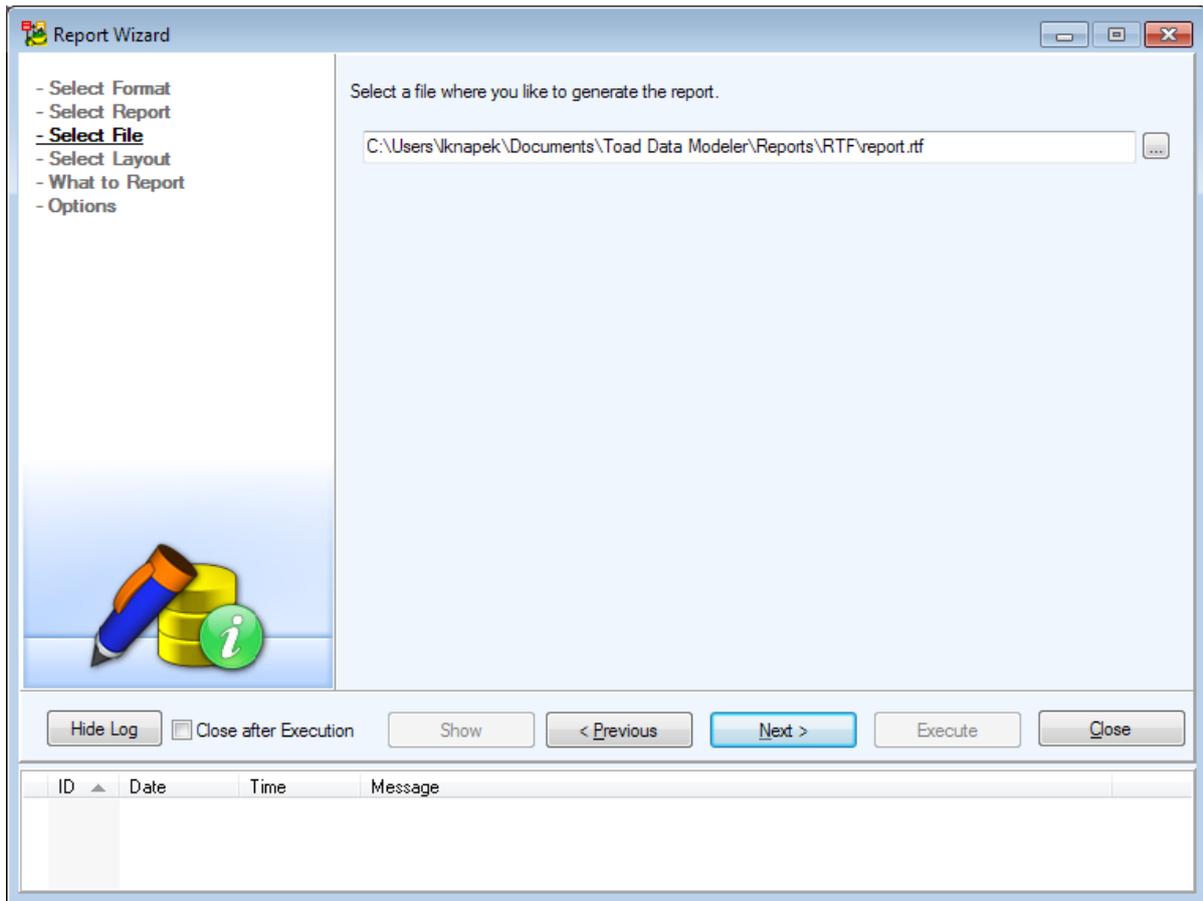
- The final dialog where you can set a few last options.
- Click on **Execute** to generate the report. A dialog window will inform you when the process is done.

## Generating RTF and PDF Reports

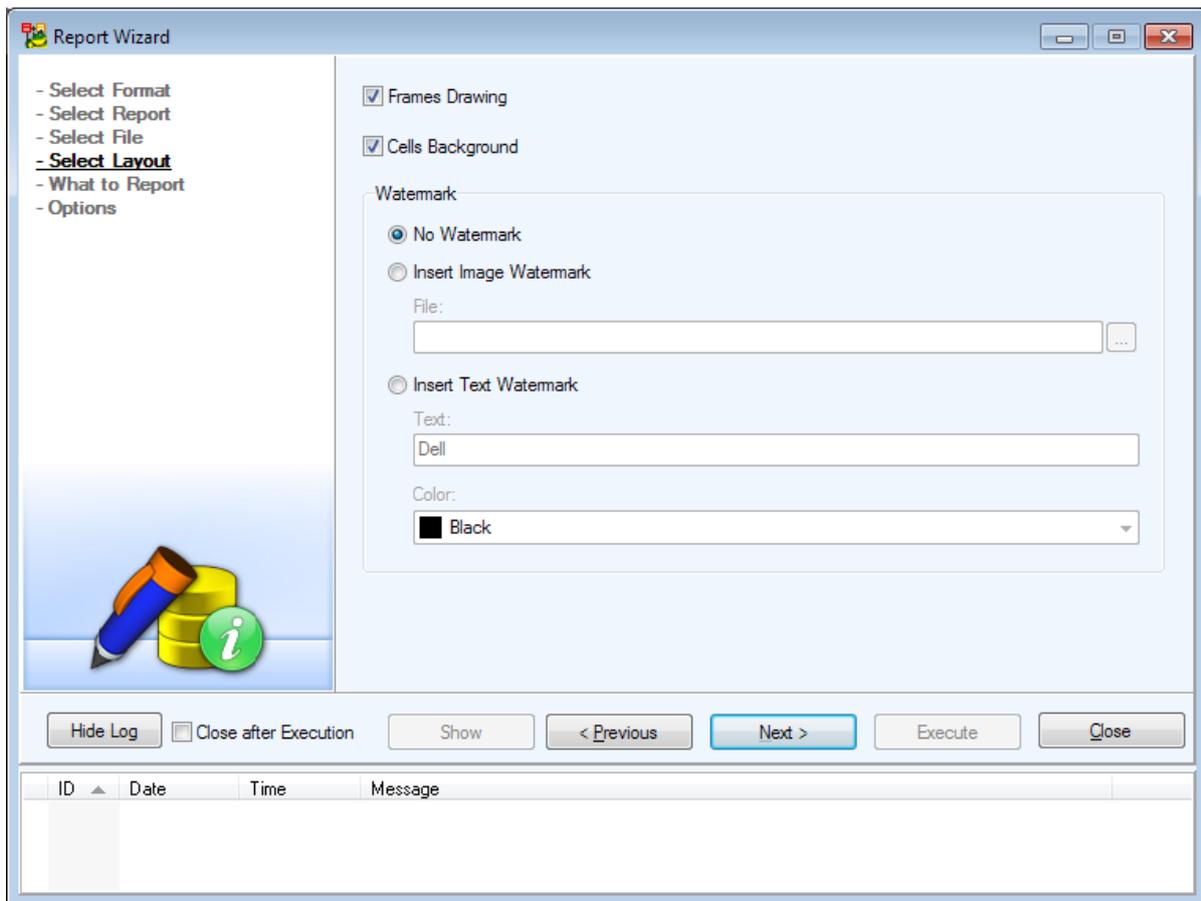


- On this dialog you are able to change the report caption and its language.

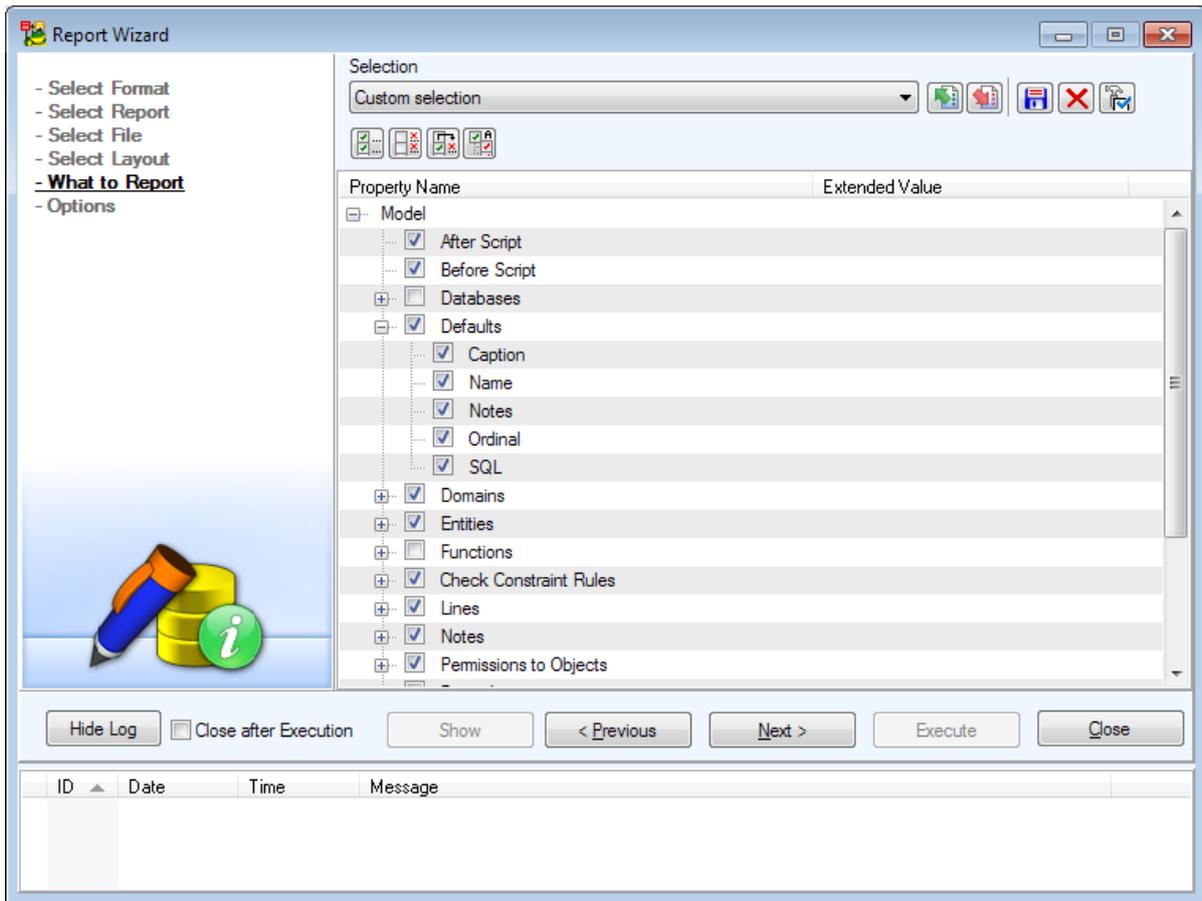
**i** NOTE: You can generate Reports in many languages provided that you have the appropriate Dictionaries installed. For more information see [Dictionaries](#).



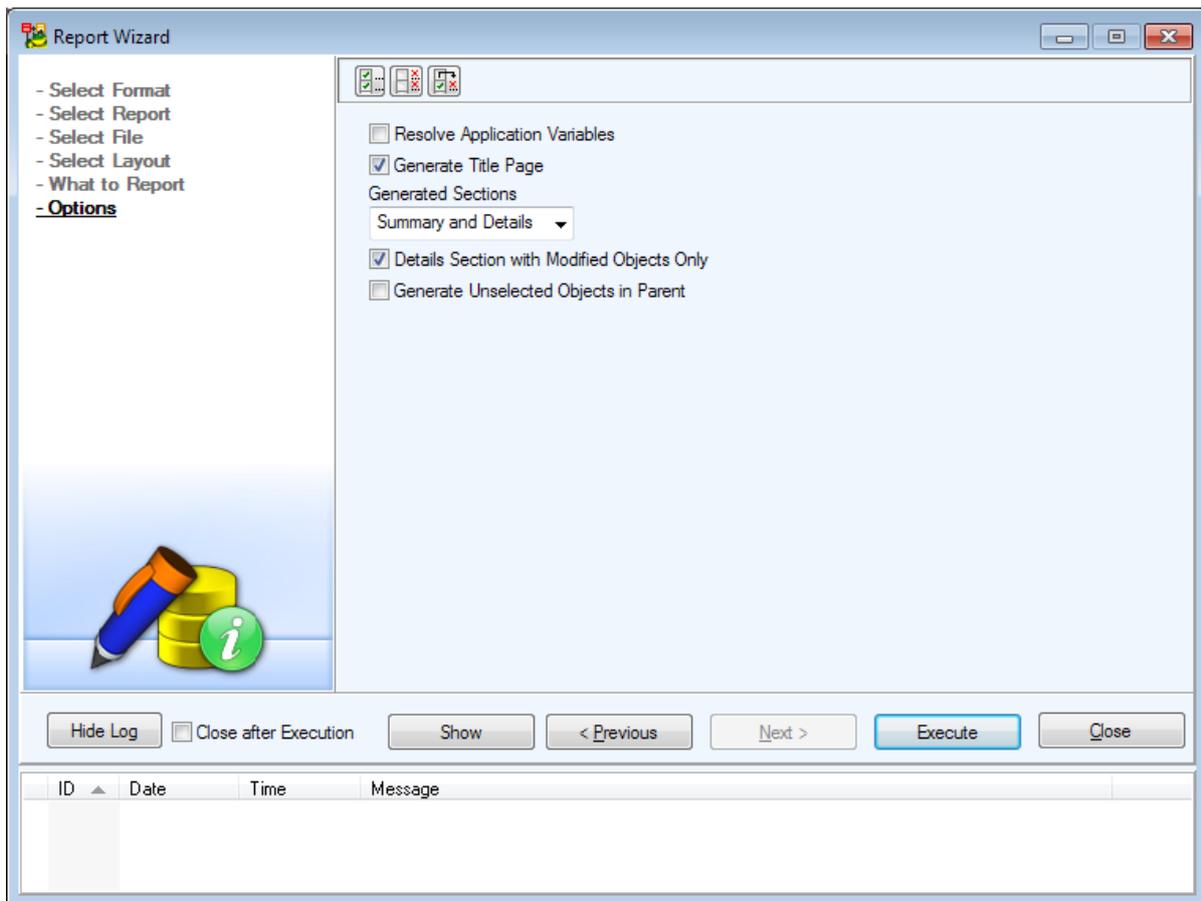
- You are able to define your own path where the generated report will be saved, if you need to.



- A few options regarding the report look are available here. Also note the option to insert text or image watermark.



- Now is the time for you to decide what kind of information should the generated report contain. If you want to get a complete report, check all boxes. For easier management use buttons located on the top.



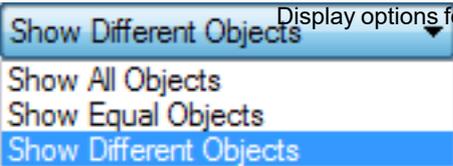
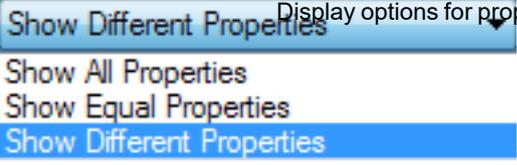
- A few final options are available here. For example, you can decide between generating Summary only, Details only, or both.
- Click on Execute to generate the report. A dialog window will inform you when the process is done.

### **What to Report Dialog Buttons**

<b>Option</b>	<b>Description</b>
Selection Menu	Allows you to save time by loading saved selection. For example, you may decide that you only want your report to cover Entities, some properties of Relationships and After Scripts. Selecting these items every time you generate a report would be frustrating. That's why you have the option to check these items once and then save the selection and load it every time you generate another report.
Save Selection	Saves the current selection. If a selection is chosen in the Selection Menu, it will be rewritten.
Load Selection	Load the selection chosen in the Selection Menu.

Option	Description
Save as New	Saves the current selection as a new selection. You will be prompted to enter a name.
Delete	Deletes the selection chosen in the Selection Menu.
Set as Default	Sets the selection chosen in the Selection Menu as default. Such selection is marked by an asterisk in Selection Menu.
Select All	Selects all items.
Deselect All	Deselects all items.
Inverse Selection	Inverts the selection.
Auto Check	On: Checking/Unchecking an item will also check all its sub-items. Off: Checking/Unchecking an item will not check its sub-items.

### Compare Tree Overview

Option	Description
	Generate a Change Report.
	Expand All.
	Collapse All.
	Go to next object in the tree.
	Go to previous object in the tree.
	Display options for objects.
	Display options for properties.
Source Model	The source model to which the second model is compared to.
Model to Compare	The model which is compared to the source model.

## Compare Tree Icons



Difference between Source model and Compared model.

**Right-click an item to see the following options:**

Option	Description
Expand All Children	Expands all sub-items of the selected item.
Collapse All Children	Collapses all sub-items of the selected item.

## Model Update

Toad Data Modeler allows you to update models to synchronize changes between your database and your model.

**Example:** You loaded the database structure of your database to Toad Data Modeler (Reverse Engineering). - Model A was created. Then you made some changes in your database and now you want to update the Model A. For this purpose, you can use the **Model Update** feature.

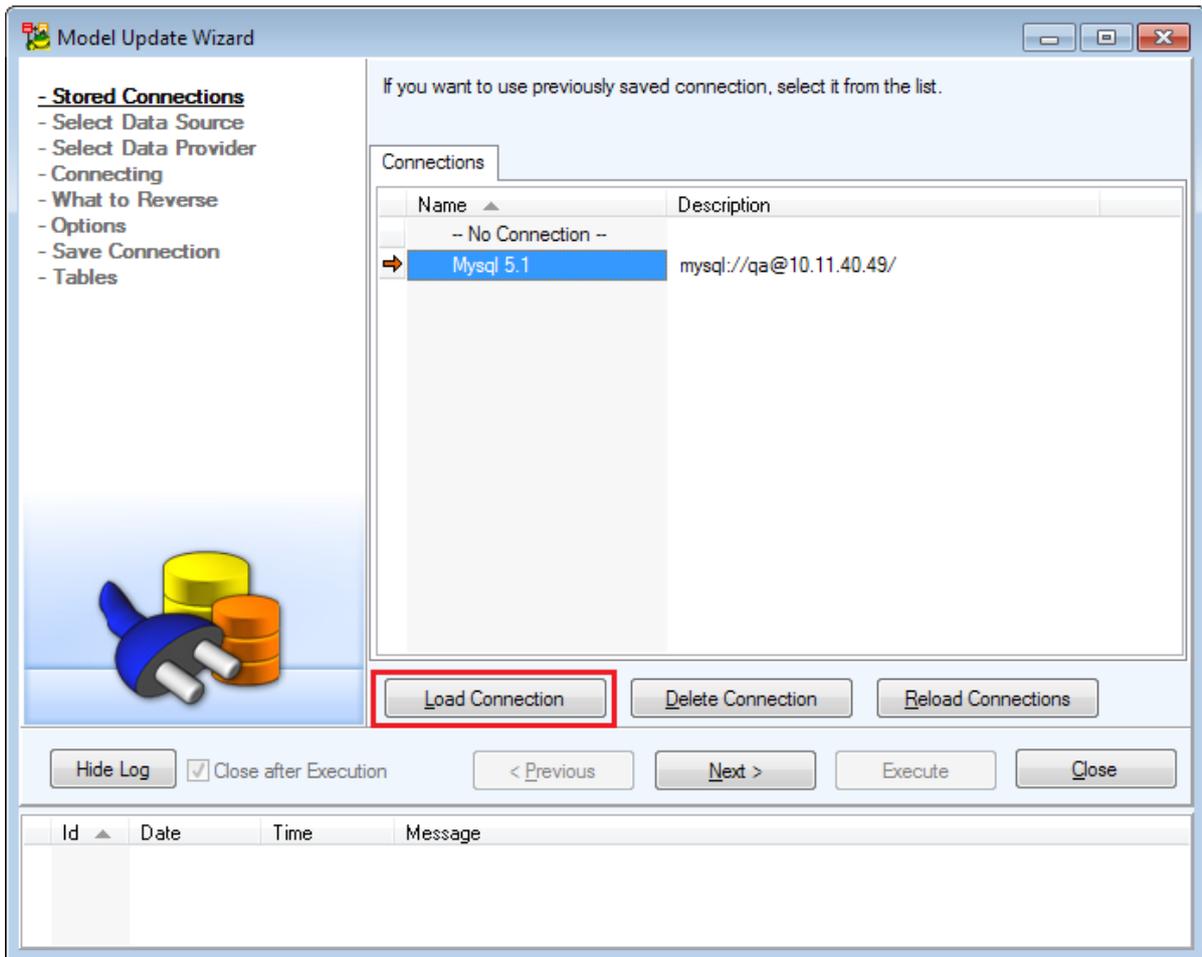
Model Update includes operations such as **Connections**, **Model Comparison** and **Model Merge**

### To update your model

- Open the model that you want to update.

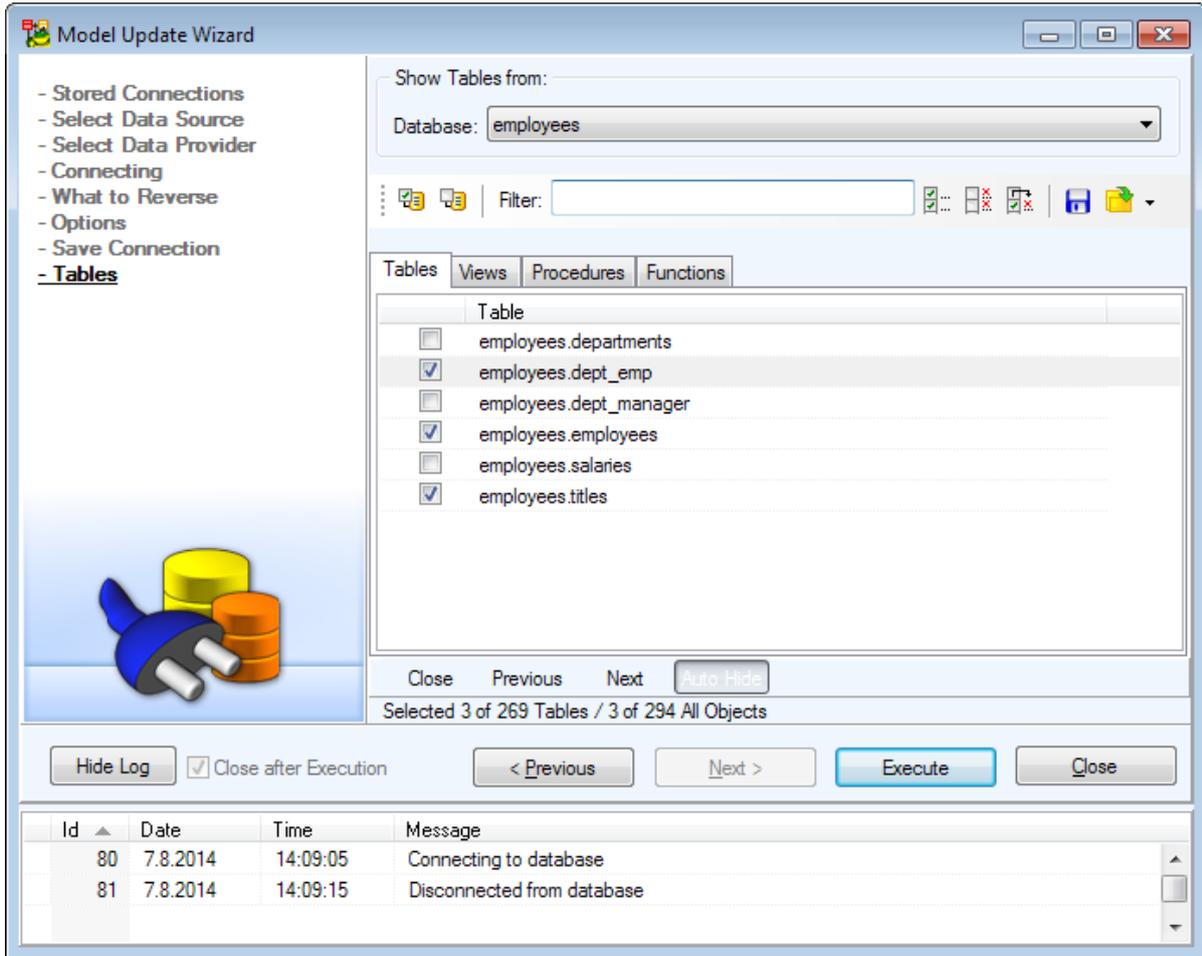


- Click the button on the **Menu Toolbar** or Select **Model | Update Model from Database** to open the **Model Update Wizard**

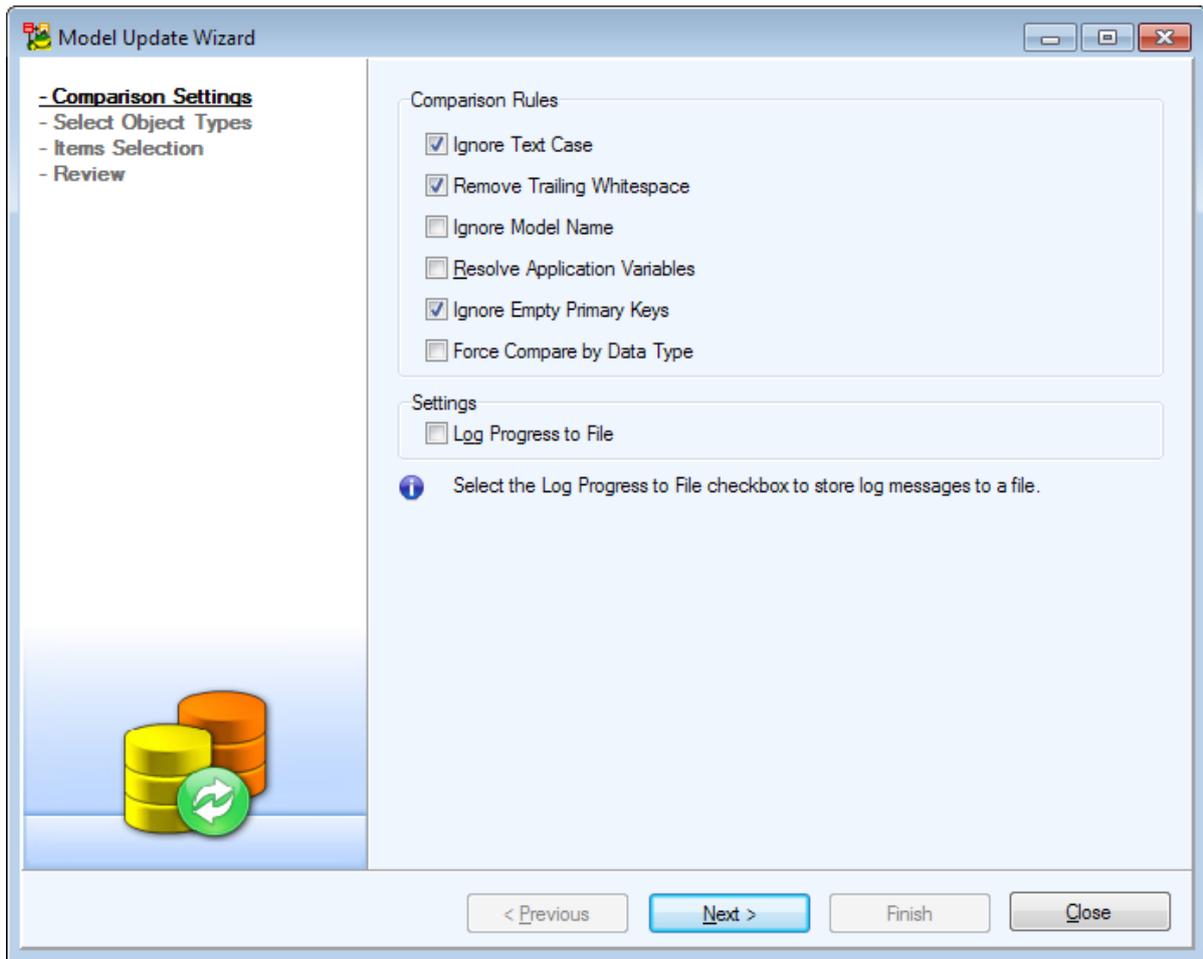


- Select Connection from the list of stored Connections. If you don't already have one, see [How to Set Up A New Connection](#).

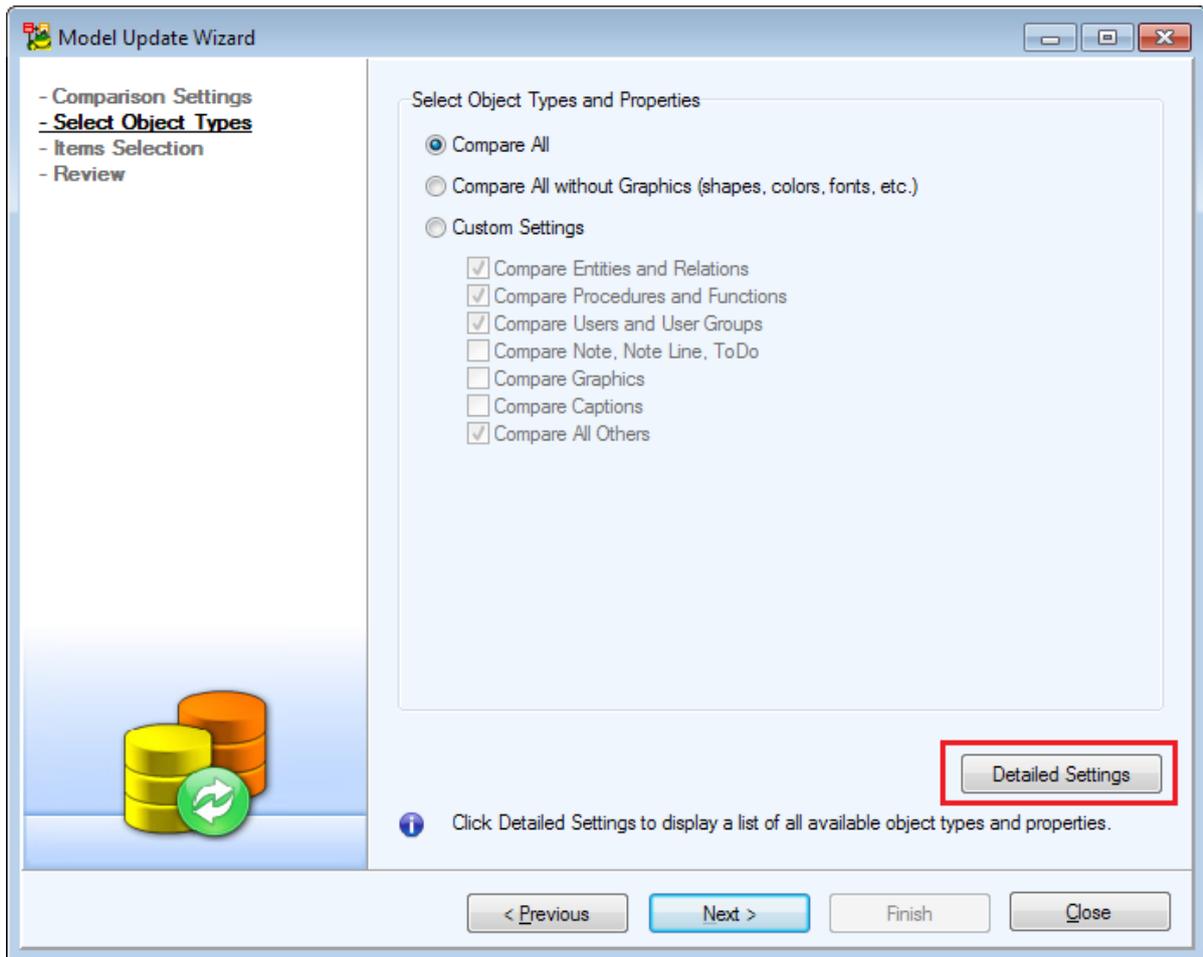
**i** Note: You might be prompted to enter a password to connect to the database. This can be changed by checking **Save Password** checkbox in specific Connection settings.



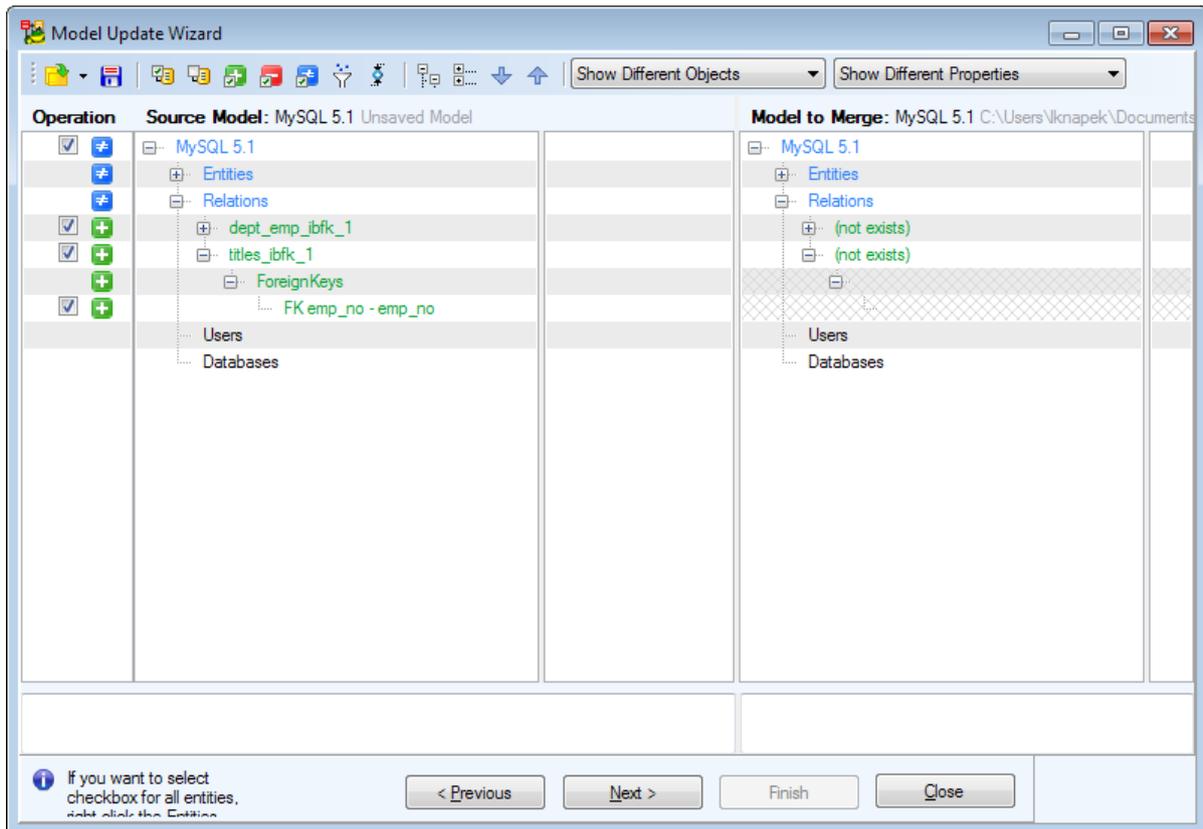
- Select the objects you want to update in your model. You can use the **Filter** and **Select All/Deselect All/Invert Selection** buttons for easier object management. Also note that you can **Export** or **Import** your selection to or from a file. After you're done selecting, click on **Execute** button.



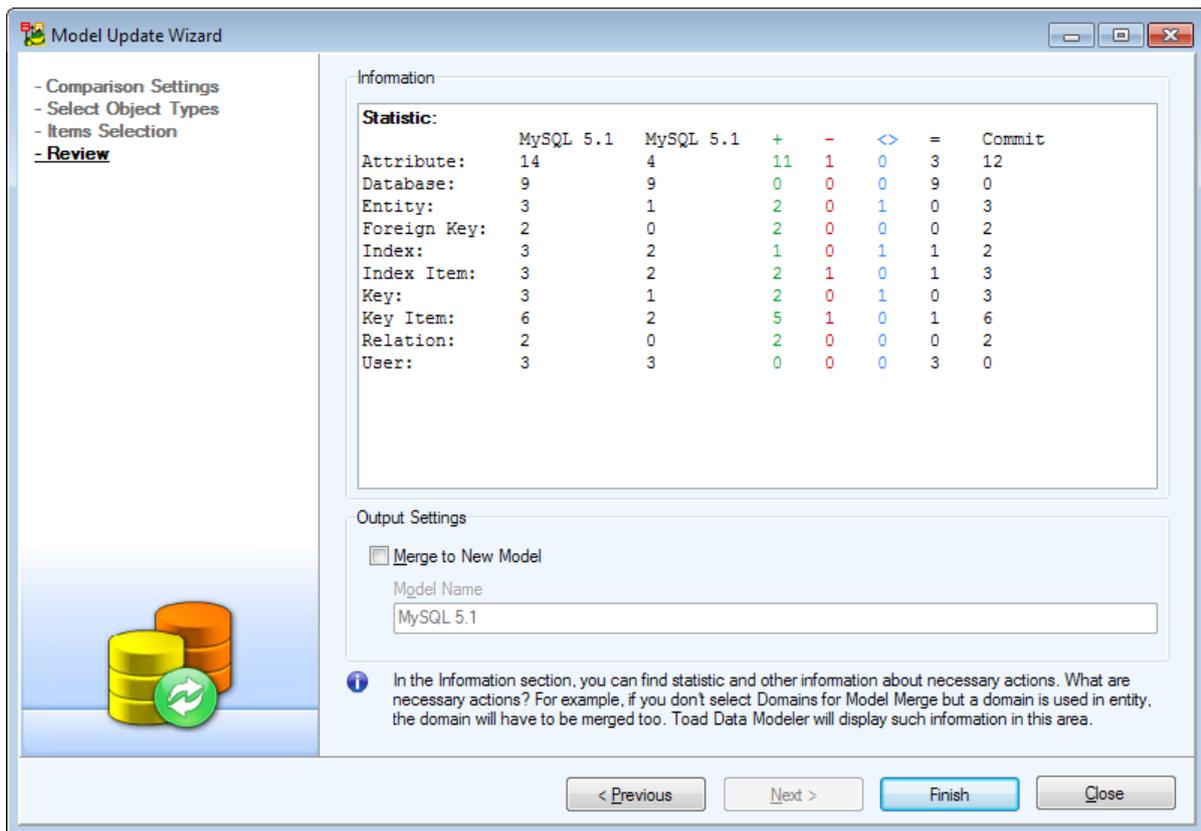
- Before Toad Data Modeler compares your model with the database, you need to go through some settings related to comparison.



- Here you can select what types of objects should Toad Data Modeler compare between given model and the database. If shown settings aren't detailed enough for you, click on the **Detailed Settings** button. This will allow you to choose not only from all groups of objects, but from the objects themselves. To update everything in your model, check every checkbox or select **Compare All**.
- After you're done with selecting the objects, proceed to the next dialog.



- You are now able to see the Compare Tree. It displays all differences between your current model and the database. The Model Update changes the model on the right (your model) so it matches the model on the left (the database). Check the changes you would like to execute and proceed to the next dialog.

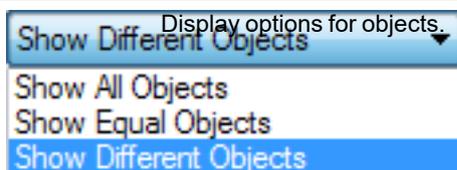


- The last dialog shows you the review of the **Model Comparison**. An option to **Merge Model** is available in case you don't want to only update your model. This option creates a new model which will appear in your **Application View** after finishing the update. Click on **Finish** and after a short while your Model will be updated.

### Compare Tree Overview

Option	Description
	Import selection from a saved file.
	Save selection to a file.
	Runs the verification process. The verification might return a warning  . Details are available in Verification Log.
	Checks all items.

Option	Description
	Unchecks all items.
	Checks all items that exist in the Source model, but not in the Target model (CREATE).
	Checks all items that does not exist in the Source model, but exist in the Target model (DROP).
	Checks all items that exist in both models, but are different (ALTER).
	Opens the <b>Wildcard Dialog</b> where you can define settings for bulk selection/deselection of the <b>Action</b> box of the items listed on page <b>Select Items</b> .
	<p>Refresh Necessitated Items</p> <p>Explanation: Some objects are related together (e.g. entity and domain, entity and relationship). Let's say you uncheck a Domain in <b>Select Object Types</b> dialog. However you keep an Attribute of the Domain type checked for conversion. In the next screen the Domain will be selected for conversion (and highlighted in gray), even if you don't want it to. This is because of its relationship with the Attribute, which cannot exist without the Domain.</p> <p>Now, if you uncheck the Attribute, the Domain will still be checked for conversion. This is where you use this button. It runs through all checked objects and removes the Domain highlighted in gray since the Attribute is no longer checked. That means the Domain is no longer necessary, since it has no relationships with currently checked objects and you unchecked it in <b>Select Object Types</b> dialog.</p>
	Expand All.
	Collapse All.
	Go to next object in the tree.
	Go to previous object in the tree.



Option	Description
Source Model	The source model (often the database).
Model to Merge	The model which is going to be updated.
Operation	Check this checkbox to update the item. Uncheck this checkbox to not update the item.

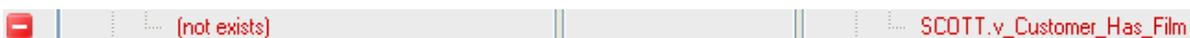
### Compare Tree Icons



Difference between Source model and the updated model.

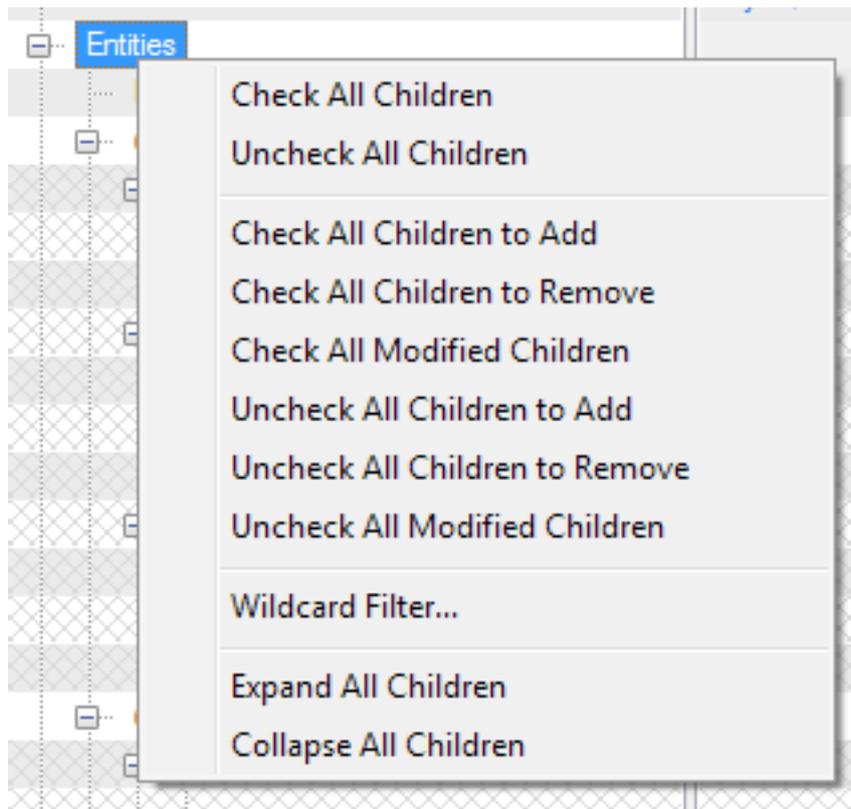


This object exists in Source model but does not exist in the updated model.



This object is missing in Source model but exists in the updated model.

Right-click an item to see the following options:



Option	Description
Check All Children	Checks the <b>Operation</b> checkbox of all children items.
Uncheck All Children	Unchecks the <b>Operation</b> checkbox of all children items.
Check All Children to Add	Checks the <b>Operation</b> checkbox of children items that exist in Source model but not in the Target model (  items) to generate the CREATE statement.
Check All Children to Remove	Checks the <b>Operation</b> checkbox of children items that are missing Source model and exist in Target model (  items) to generate the DROP statement.
Check All Modified Children	Checks the <b>Operation</b> checkbox of children items where properties differ (  items) to generate the ALTER statement.
Uncheck All Children to Add	Unchecks the <b>Operation</b> checkbox of children items that exist in Source model but not in the Target model (  items) to not generate the CREATE statement.
Uncheck All Children to Remove	Unchecks the <b>Operation</b> checkbox of children items that are missing Source model and exist in Target model (  items) to not generate the

Option	Description
	DROP statement.
Uncheck All Modified Children	Unchecks the <b>Operation</b> checkbox of children items where properties differ (  items) to not generate Change Script for this change.
Wildcard Filter	Opens the <b>Wildcard Dialog</b> where you can define settings for bulk selection/deselection of the <b>Operation</b> box of the items listed on page <b>Select Items</b> .
Expand All Children	Expands all sub-items of the selected item.
Collapse All Children	Collapses all sub-items of the selected item.

## Synchronization

Toad Data Modeler helps you with synchronization tasks and allows you to:

- Generate SQL scripts representing changes made to your model
- Transfer changes made in your database to your existing physical models
- Keep your model and your database synchronized and up-to-date during the development process

## Database and Model Synchronization

### Situation A - Generation of Change Scripts

You have made changes in your physical ER diagram in Toad Data Modeler and now you need to apply the changes to your database.

Solution: Compare the models and generate **Alter Script** in Toad Data Modeler. Then use a third party software to connect to your database and execute the generated script. All your changes are now reflected in database. (Note: Change Script generation is not supported for all databases, for further information see [Supported Databases](#).)

### Situation B - Model Update

You or someone else made changes to your existing database (development version of database) and you need to transfer the changes to your model.

Solution: Use the **Model Update** or **Model Merge** features to update your physical model.

## Physical Model and Logical Model Synchronization

You have created a Logical model in Toad Data Modeler and need to convert the logical model to physical model for the selected database system.

Solution: Use the **Model Convert** or **Simple Model Conversion** feature for logical (LER) to physical (PER) model conversion.

**i** Note: If you make changes in your LER model and need to update the existing PER model, perform the LER to PER conversion again and then compare the two PER models.

## Limitations

There are many factors that affect synchronization features in Toad Data Modeler. For example:

- Data types can be defined through logical types that do not exist in physically existing databases, such as Domains, Dictionary Types etc. Toad Data Modeler has to correctly resolve the data types when comparing attribute (table column) properties.
- Objects must be paired by specific rules, for example in your Oracle Database model, schema assignment is not required, but in your database, schema is always present.
- There can be application variables defined in your models, for example `<%ParentTableName%>` and the application variables must be resolved during the comparison as well.

In result, the synchronization of your model and your database is more complicated than comparison of two physically existing databases. From technical point of view, the comparison is based on more complicated algorithm that requires more resources.

## Recommendations

- Use the synchronization features on models of small to average size. If you work with large models, try to split the model to smaller parts (separate models), if possible.
- Use specialized software for comparing and synchronizing two large physically existing databases. You could theoretically reverse engineer both databases, create two models, compare them and try to generate change scripts, but due to limitations mentioned above the process would require more resources and take more time. Other products, that do not work with models, but rather with databases (e.g. Toad for Oracle, Toad for SQL Server, etc.), can offer better performance and additional benefits - like comparison of physical properties and data. Toad Data Modeler is a tool primarily used to create a visual representation of database structure and it should be used during database development. It is not an alternative or competitive tool for specialized schema comparison.

## Model Size Limitations

- Toad Data Modeler doesn't have fixed limitations based on number of objects. You can work with models that contain hundreds, or thousands of tables. Please note that large number of objects in a model may hinder performance significantly, depending on your computer specifications. Number of other items in your model (stored procedures, functions, sequences etc.) affects the performance as well. Other factors include:
  - Type of task - you can model large database structure, generate complex SQL scripts or detailed reports, but synchronization tasks might reach the application limits sooner than generation of SQL scripts, for example.
  - Number of running processes on your machine.
  - Limited access to Microsoft Scripting Engine (standard part of operating system, however in some situations access can be monitored by antiviral software etc.)

# Print

## Tips before You Print

- To set size of the pages, select **File | Page Setup**. To customize the page size, see the **Page Setup** dialog | **Scale** area | and from the **Page** box, select **Custom**. Define Height and Width on the right.
- To print your ER diagram on one page, select **File | Page Setup** | select the **Fit to Page** checkbox.
- Turn off the display of page boundaries - select **Settings | Options | Graphics** | clear the **Visible Page Boundaries** checkbox.
- In **Settings | Options | General**, clear the **Print Gradients** checkbox for much faster print performance. (It is disabled by default.)
- Turn off page numbering and a frame around your ER diagram - select **File | Print | Settings** tab | clear the **Print Frame** and **Print Page Number** checkboxes.

### To print your model

1. Click  on the toolbar (or select **File | Print**).
2. Select a printer and click **Properties** for more configuration options.
3. On tab **Settings**, select from the available options.

#### **i** Note:

- You can also print to PDF and plotter, see [Printing to PDF and plotter](#) for more information.

## Page Setup

1. Select **File | Page Setup**.
2. Select page size, margins, orientation etc.

To define the page size on your own, select **Custom** from the **Page** box. Define Height and Width on the right.

#### **i** Note:

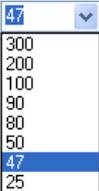
- The measure unit can be set in **Settings | Options | General | Select Unit of Length** - inches or millimeters.
- Page Format settings are saved with a model.

Option	Description
Fit to Page	Select this option to print the entire model on one page. This option is model-dependent.

Option	Description
Enlarge	Select a scale of your model for print.
Portrait/Landscape	The option selected in the <b>Page Format</b> dialog will be automatically set in the <b>Printer Properties</b> dialog. You can set a different orientation for every model, the option is model-dependent.
Printer	Select your printer. Then in the <b>Page</b> box you can see formats that the selected printer supports (enabled items) and other formats (disabled items). (This feature should solve problem with printing on plotters.)

## Preview

Click  on the toolbar (or select **File | Preview**).

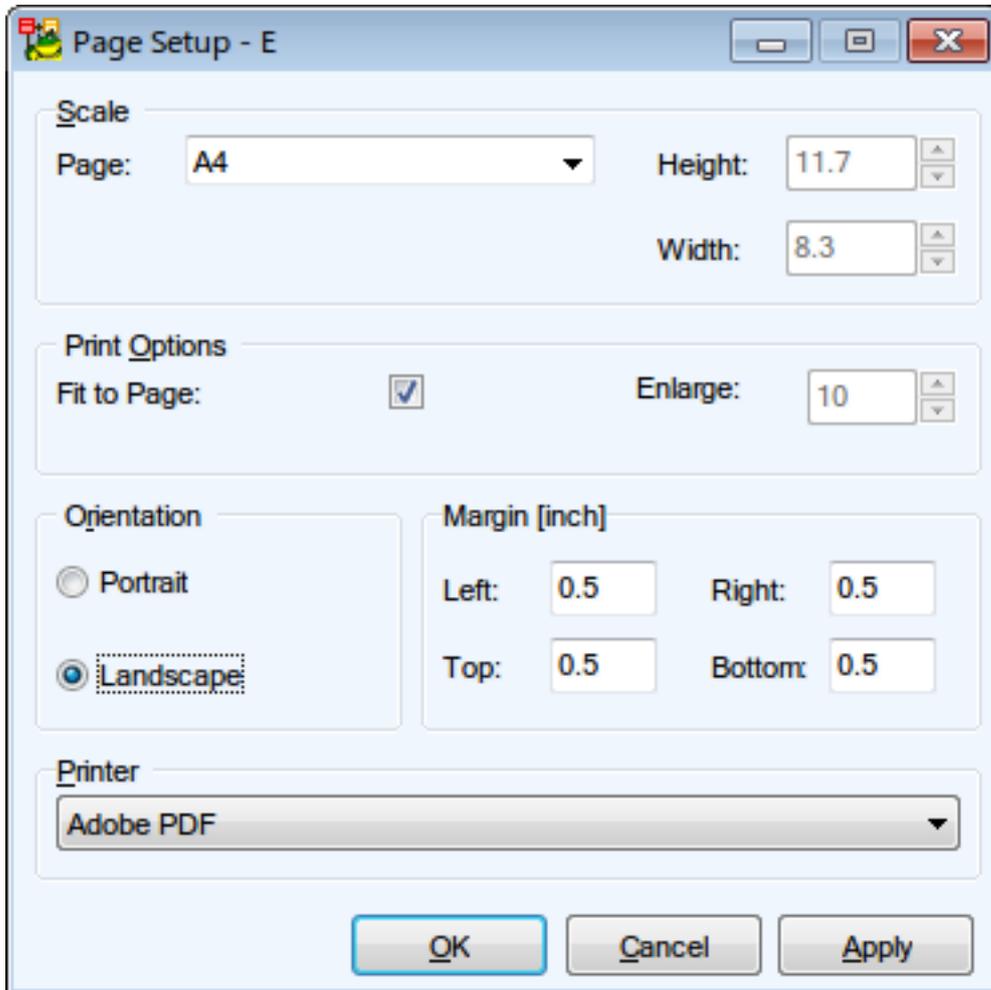
Option	Description
	Switch between pages.
	Displays current page number/total number of pages.
	Sets Scale percentage.
	Opens the <b>Print Setup</b> dialog.
	Displays/hides all pages on the left side of the dialog.
	Exits the dialog.

# PDF Printing

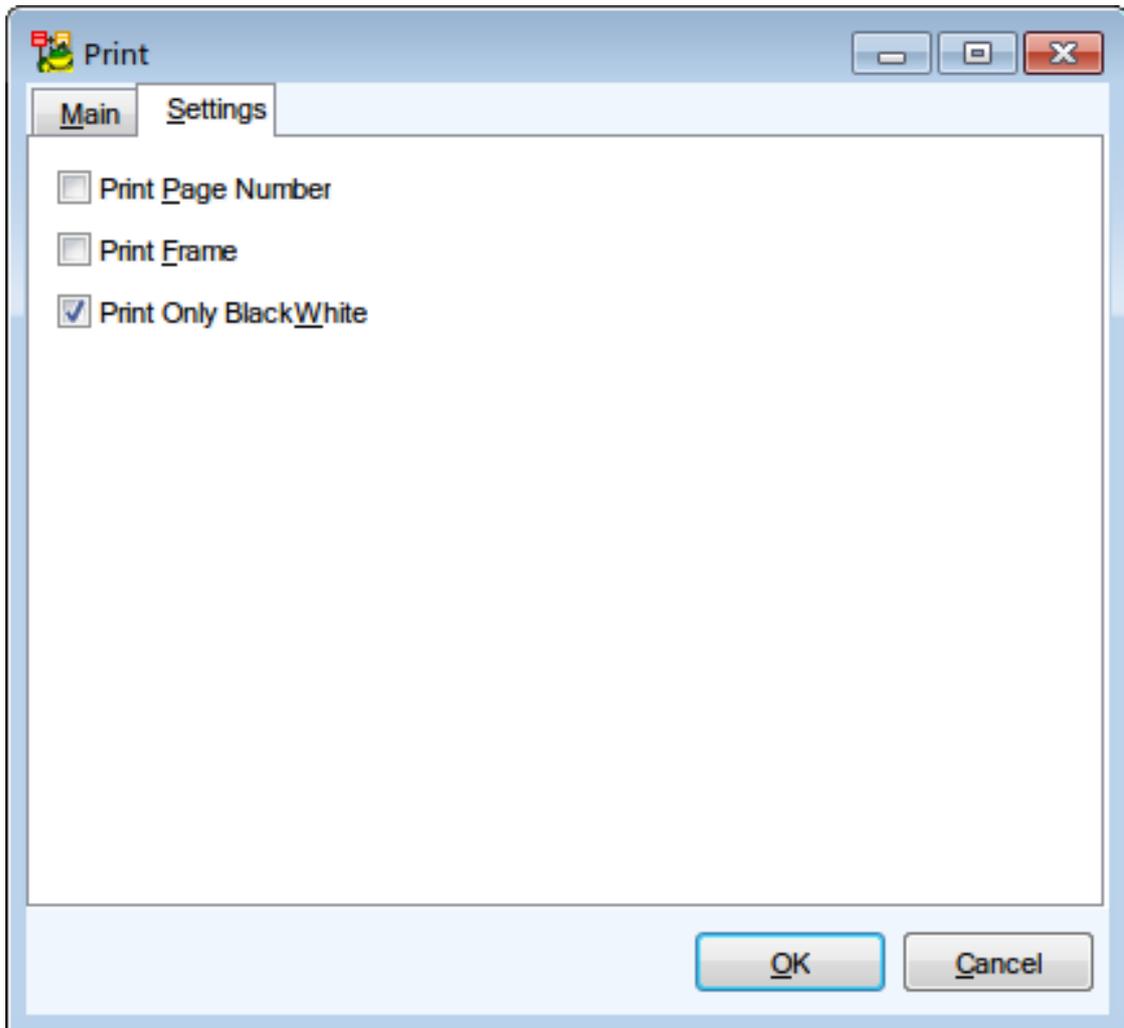
Toad Data Modeler allows you to print ER diagram of your model to a PDF file, which is especially useful step in order to print the diagram to a plotter.

Printing to PDF is similar to printing to printer:

1. Go to **File Menu | Page Setup** and configure printing properties such as your document size, orientation, margins...



2. Once you're done configuring, go to **File Menu | Print**. There are also several options available on **Settings** tab.



3. Click **OK** to print the PDF document. Some PDF printers display an additional configuration dialog in order to print the PDF.

**i** Note: Most PDF printers work with parameters set in the **Page Setup** dialog. However, some PDF printers allow you to customize page size and other settings right before actually printing the document (e.g. Adobe PDF Printer or PDF Factory).

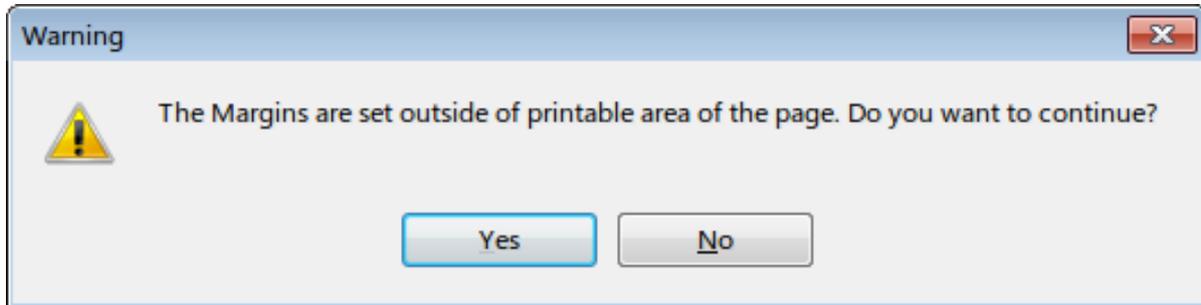
Generally, you should not configure a property in the PDF printer itself if the property can be configured via Page Setup. The exception to this rule are settings that are not available in Page Setup, such as **Image Compression and Downsampling** in Adobe PDF Printer. These settings should not impact the configuration set in Page Setup.

## PDF Printing Limitations

There are several situations, where you may not be able to print the PDF file at all, or the result may not look as intended. Usually, this is not limitation of TDM itself rather than a limitation of some component TDM uses.

## Windows PDF size limitation

Toad Data Modeler uses Windows GDI (graphics device interface) to print PDF files. The GDI has **129 inches** (about 3200 millimeters) limitation, meaning neither height or width may exceed the limit. If any size of the PDF exceeds 129 inches, you will receive a warning message when you try to print the PDF.



If you print the document anyway, its page size will be reverted to A4 (which is default).

## PDF printers size limitation

PDF printers which allow you to customize print properties right before printing the actual document (**Adobe PDF Printer, PDF Factory**) seem to have one limitation in common. If you configure the page size in the PDF printer properties (which is not recommended, see the note above), the height and width should not add up to more than **129 inches** (about 3200 millimeters). If the width+height is more than 129 inches, one of the two things will happen:

- You confirm the **Print** dialog, but nothing happens afterwards (Adobe PDF Printer).
- You confirm the **Print** dialog and get "*The Margins are set outside...*" error (PDF Factory).

Printers using properties set in **Page Setup** seem to handle larger sizes well (e.g. PDF Creator, Bullzip PDF Printer, Nitro PDF).

## Printing to Plotter

When printing to plotter, it is recommended to print PDF of your model ER diagram first. Toad Data Modeler supports variety of common plotter paper size formats, but you may find yourself in a situation where you need to print diagram larger than 129 inches, but you keep running into various limitations (mentioned in chapter above).

To print such a diagram, follow these steps:

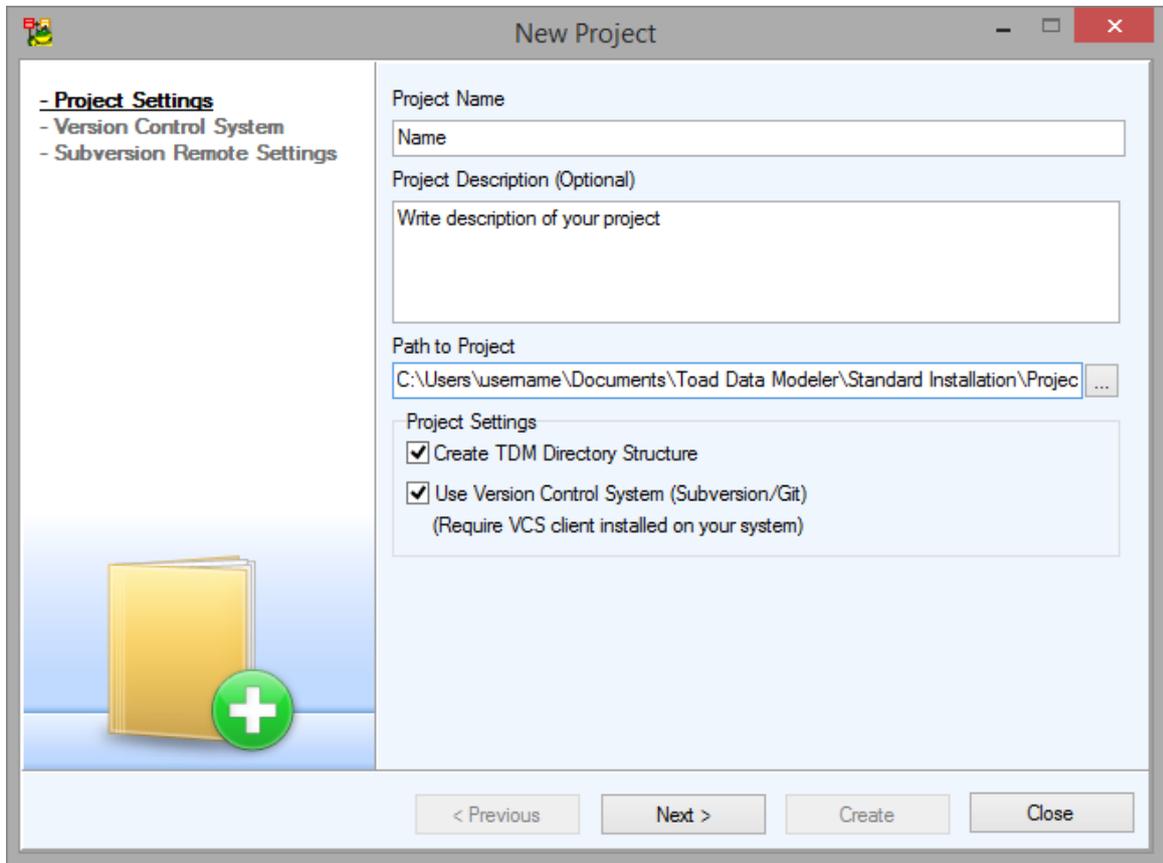
1. Know your plotter paper size. In our case, let's say the desired size is 150 to 75 inches.
2. Calculate the paper size height to width ratio (e.g.  $150/75 = 2$ ).
3. Create a PDF file with the same ratio as your desired paper size. Make sure it does not exceed any limitation (e.g. 100x50, 50x25). Also make sure that the model looks good enough when printed (Does the model fit on the 50x25 paper size? Or should you rather print 100x50 PDF?).
4. Print the PDF file to your plotter. The PDF file should be upscaled (e.g. 100x50 should be multiplied by 1.5 -> 150x50). It may be necessary to configure the upscaling in your plotter settings. Since the diagram image has been vectorized when it was printed to PDF, it can be upscaled at no quality cost.
5. The result should come out as a sharp, clear, well readable diagram of your model.

# Create New Project

Projects are used to group multiple models, files and other items.

## To create a new project

- Select **File | New | Project**



- Check **Create TDM Directory Structure** to create a project with a predefined directory structure



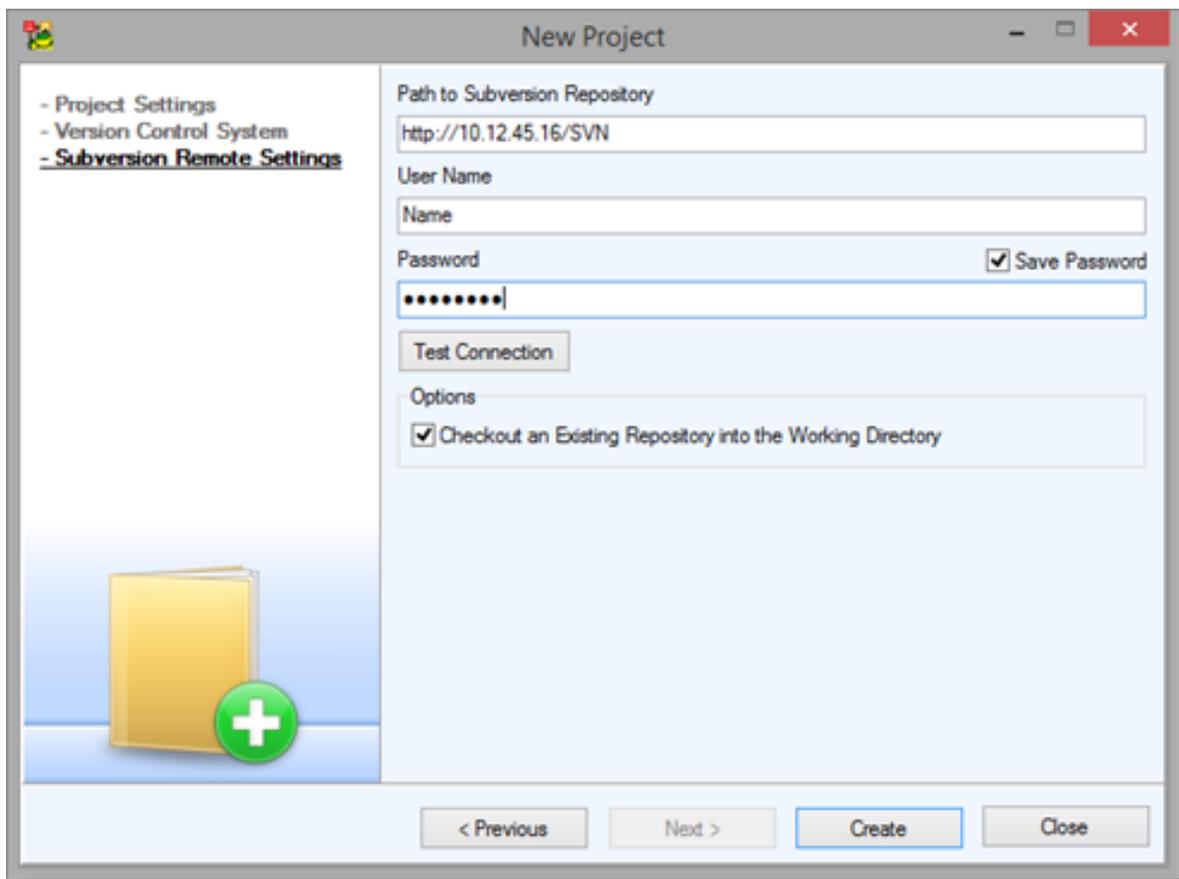
- Check **Use Version Control** to create a version controlled project. In the following steps you will need to configure it in order to create a new project. You can set up Version Control system for your projects anytime later

### To select a version control system

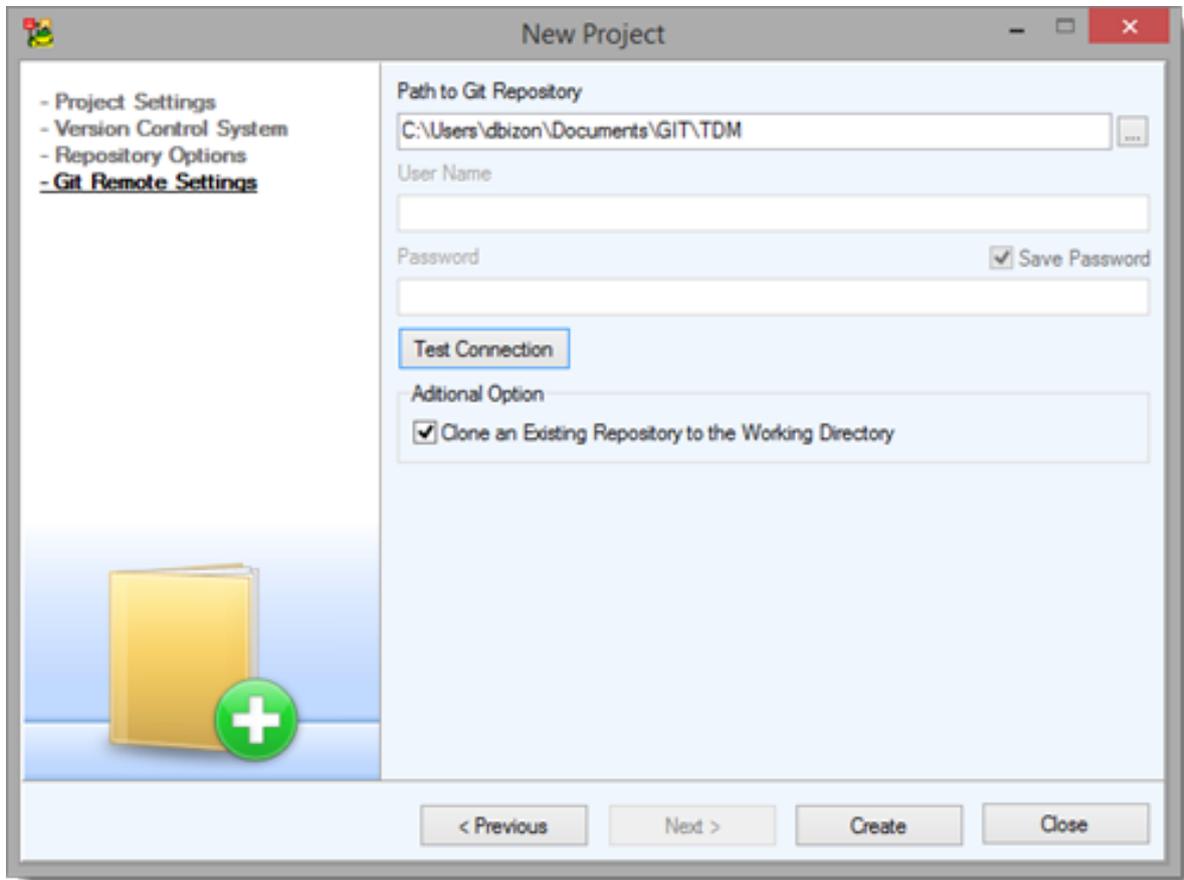
- Select a type of version control system:
- **Git** - Enter the path to the Git client executable (e.g git.exe)
- **Subversion** - Enter the path to the svn client executable (e.g svn.exe)
  - Check **Use Read Only Locking Mechanism** to enable locking files (set read-only attributes to them)
  - Check **Non Interactive** to disable all interactive prompting
  - Check **No Authentication Cache** to not save authentication tokens into cache
- Click **Next** to enter details of your repository

### To setup your repository

- **Subversion** - Enter the path (server address) to your subversion repository
  - Enter your **User Name** and **Password**
  - Check **Checkout an existing repository into the working directory** to checkout the files into your project folder



- **Git** - Check **Initialize Local Repository** to initialize your local Git repository
  - Check **Configure Remote Repository** to enter details of your remote Git repository, click **Next**
  - Enter the path to your local Git repository and check **Clone an Existing Repository to the Working Directory** if you want to pull the latest files from your remote repository

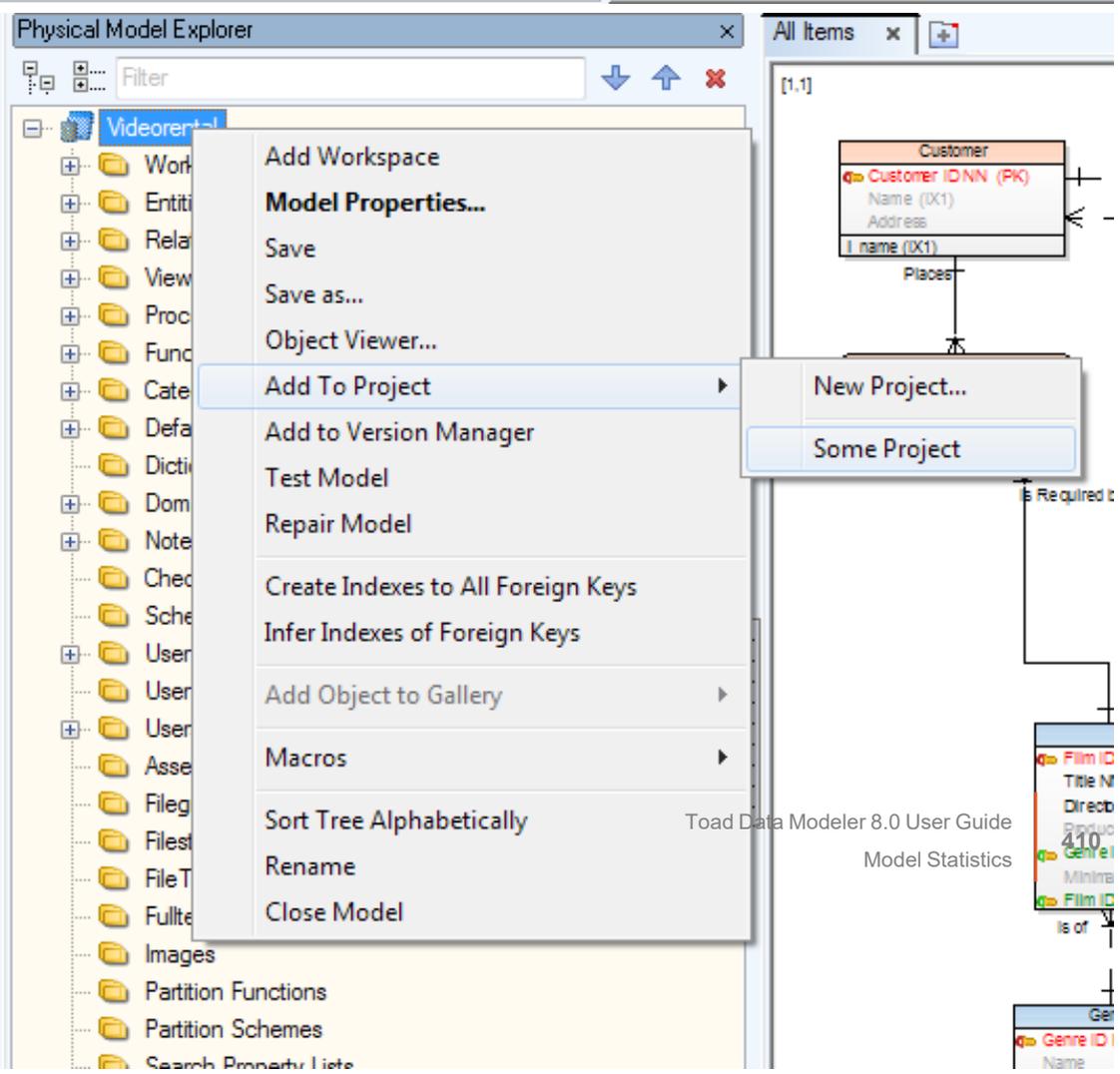
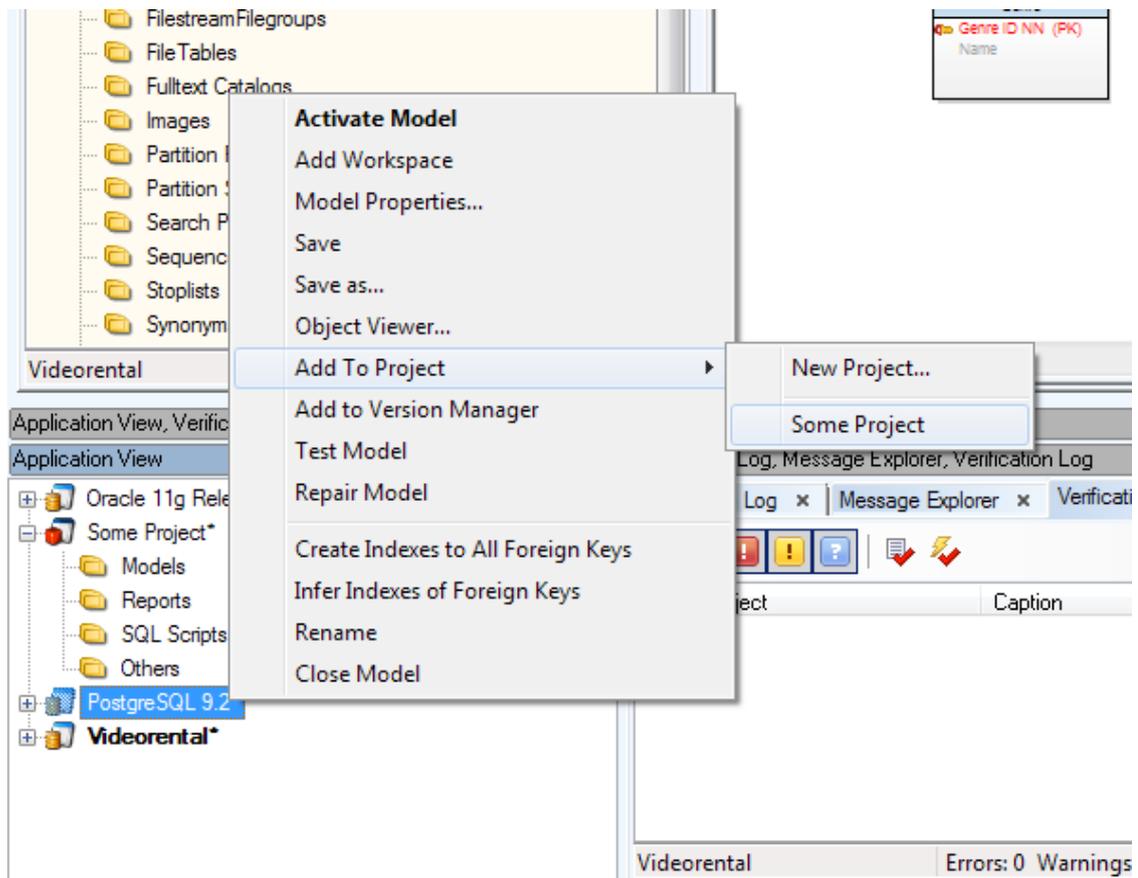


**NOTE:** The working directory for your SVN/git repository is the root folder of your new project.

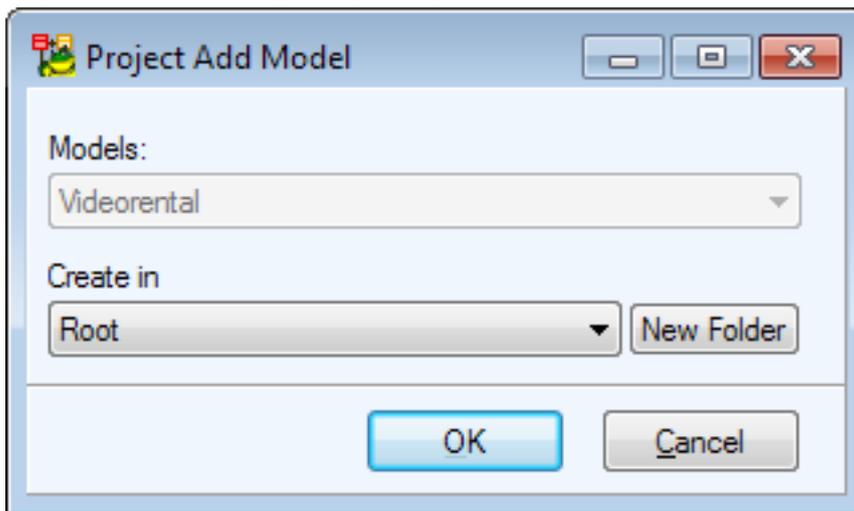
## Add Existing Models to Project

When you create a new project, you might want to add already existing models to it. This can be achieved in two ways:

- In **Application View** or **Model Explorer** right-click a model and select **Add to Project | \*Project Name\***.

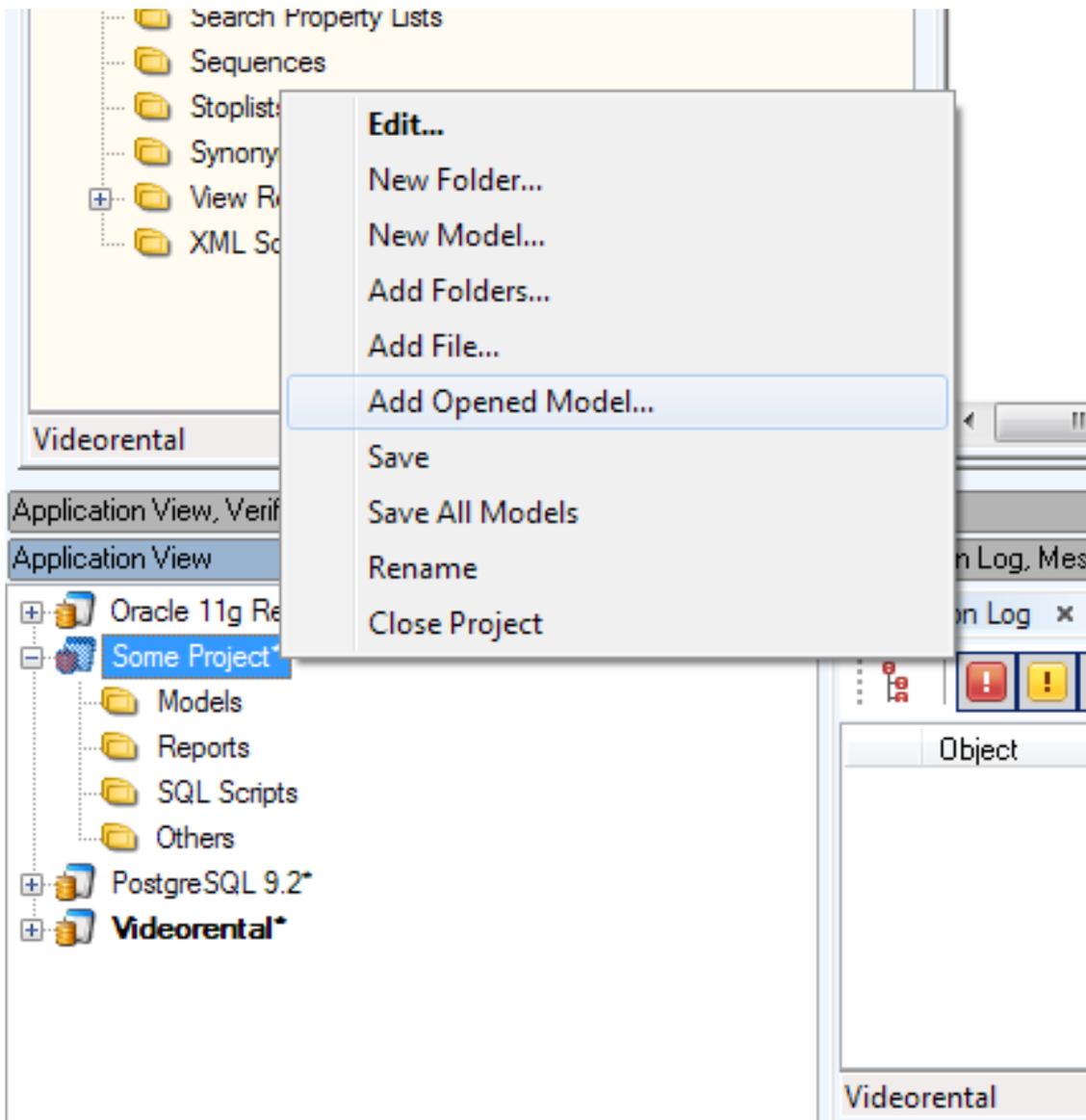


**Project Add Model** dialog displays and you can decide where to save the model.

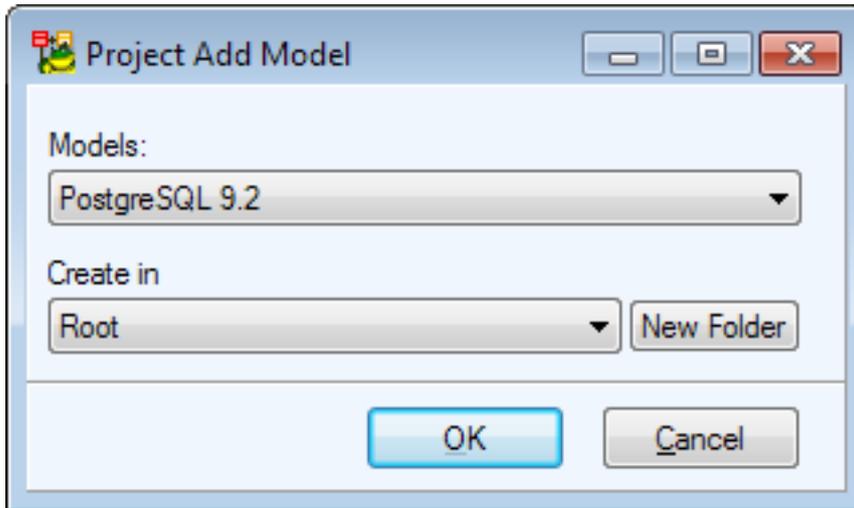


or

- In **Application View**, right-click your project and select **Add Opened Model**



- **Project Add Model** dialog displays and you can choose from currently opened models and specify their location in the Project structure.

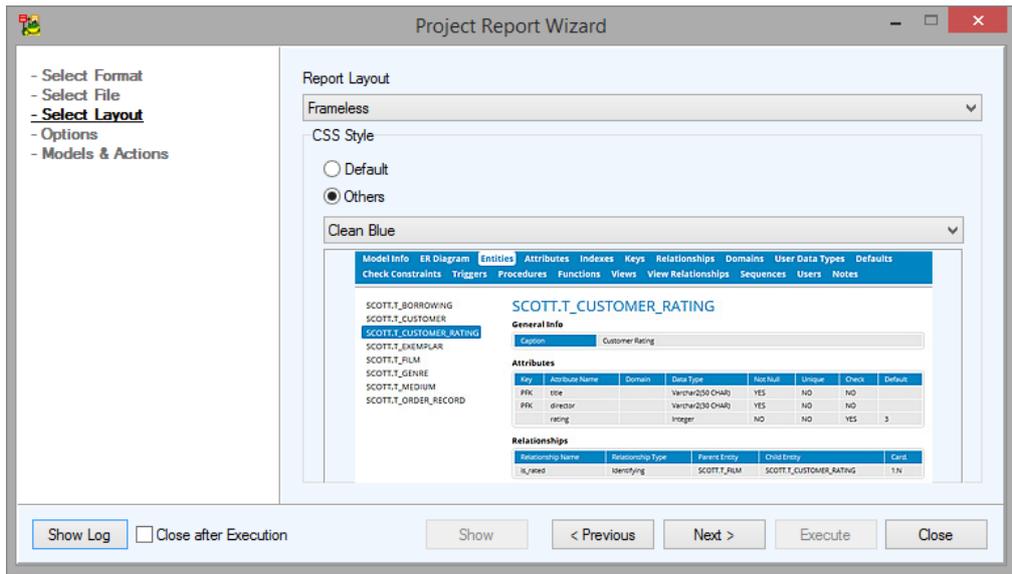


## Project Reports

Now you can easily create a report for your Toad Data Modeler projects.

### *To create a project report*

- Right-click a project in **Application View** and select **Report**
- Select the desired format for your report
- Select the destination:
  - Intelligence Central
  - A designated folder
- Choose your layout and style for the report and preview the look below
- In **Options** select what you want to include in your report
- In **Models & Actions** select which models you want to include and which **Model Action** you want to perform
- After clicking **Execute** the report will be created in Intelligence Central or your local destination folder



## XSL Transformation

This feature allows you to create user outputs for your physical models very fast. The output can be in any format that supports XSL language - e.g. HTML, PDF, CSV, text or XML.

You select among various XSL templates. Each template generates a different output (HTML, CSV etc.) Advanced users can customize the templates and modify their content to generate an output that will fit their needs.

You can generate the following reports or outputs:

- Comments (HTML)
- Complete XML (XML)
- Entities (CSV)
- Notes (HTML)
- Tablespaces (HTML)
- To Do Report (HTML)

### To generate any of the outputs

Select **Model | Generate Report | Reports / XSL Transformation**.

Option	Description
XSL Template Name	Shows types of various XSL templates that you can select.
Output File	Path where the output file will be saved. It corresponds with the path defined in <b>Settings   Options   Paths   Reports</b> .
Template Description	Gives information on the selected template.
More>>	Shows/hides options for advanced users interested in

Option	Description
	customization of the templates.
XSL File Location	Path where appropriate system template is stored. System templates shouldn't be modified. Via the button on the right, you can add other templates you created and that are missing in the box <b>XSL Template Name</b> .
Input Values	In this area, templates with parameters are displayed. You can edit them directly here.
Save XSD File As	Allows you to save the XSD file without the necessity to open it in associated application.
Show XSD File	Shows the XSD file. The XSD file describes structure of source XML that is an input for XSL transformation.
Transform	Starts the process of transformation.
View	Shows the result.

## Customize XSLT Templates

XSLT is an XML based language used for transformation of XML source documents into other documents. Output files can be XML, HTML, TXT, CSV, SVG, XSD and so on.

In general, for transformation of XML files it is necessary to specify:

1. Source (XML)
2. Template with instructions on how to convert a source to output (XSLT file)
3. Output file (where to store the output)

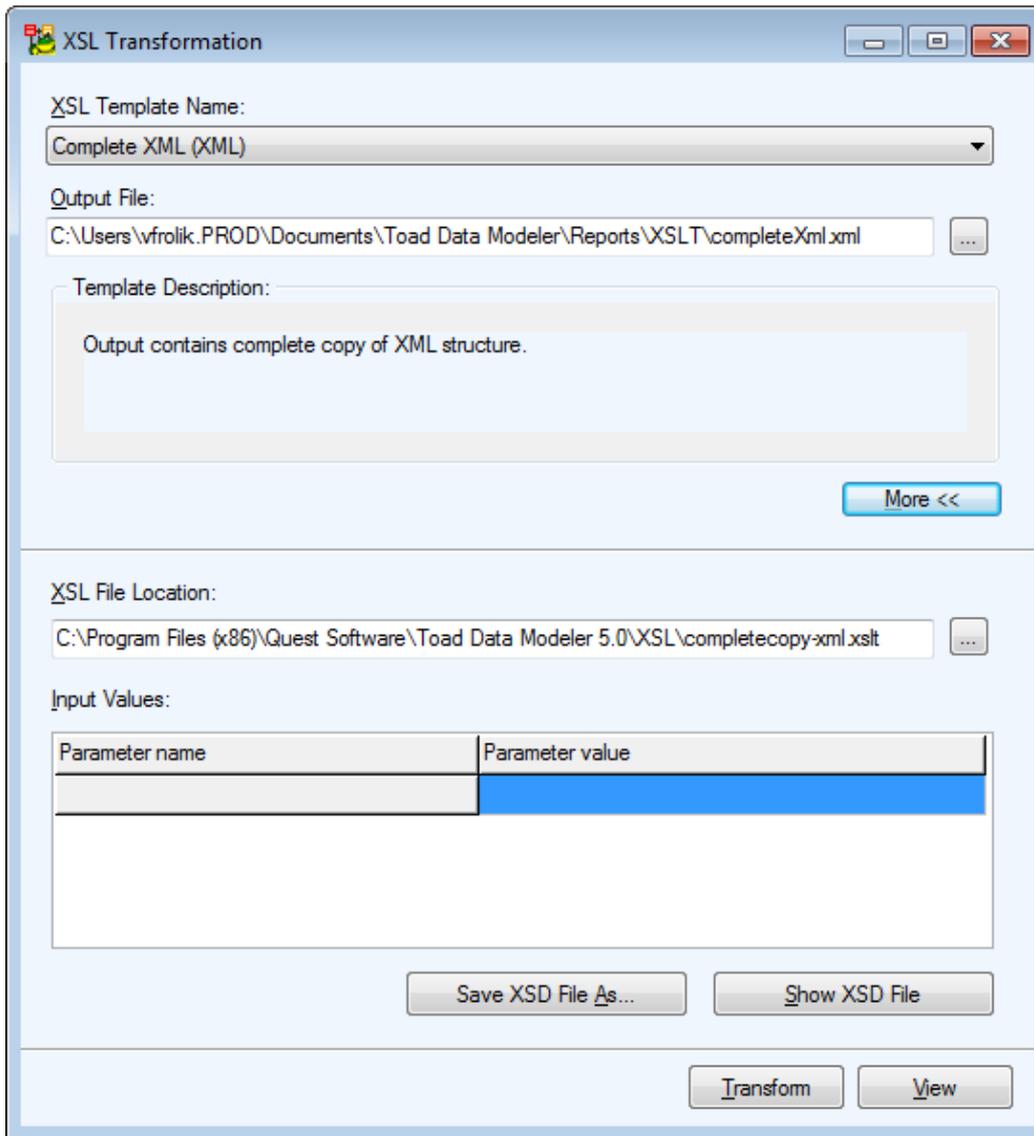
Toad Data Modeler allows you to transform simplified XML structure to the selected output. The simplified XML contains less XML data than standard TXP files (TXP is a standard file suffix for Toad Data Modeler models, however, its structure is also XML.)

### Where to Find the Simplified XML File?

You will not find it. You have to generate it. Every database system contains different items (sequences...), that's why also simplified XML structures for MySQL and Oracle may differ.

#### **To generate XML file**

1. Click **Model | Generate Report | XSL Transformation** and from the **XSL Template Name** box select **Complete XML (XML)**.



2. Click **Transform** and **View** to open the generated simplified XML file for your database model. You will see a structure of XML file and a content of your model, including entity names, attribute names, information about relationships, comments etc.

```

<SimpleTexts/>
<LabelEllipses/>
<LabelQuadrangles/>
+ <WorkingSets></WorkingSets>
<BeforeScript/>
+ <AfterScript></AfterScript>
- <Entities>
+ <Entity></Entity>
+ <Entity></Entity>
+ <Entity></Entity>
- <Entity>
  <FullName>SCOTT.T_MEDIUM</FullName>
  <Id>{2D7C297B-56D5-4972-8DFC-E2CEB83FDEA0}</Id>
  <Name>T_MEDIUM</Name>
  <IgnoreListWorkingSet/>
  <GalleryItemID>{00000000-0000-0000-0000-000000000000}</GalleryItemID>
  <GalleryID>{00000000-0000-0000-0000-000000000000}</GalleryID>
  <GalleryObjectID>{00000000-0000-0000-0000-000000000000}</GalleryObjectID>
  <RNOffset>3</RNOffset>
  <RNLength>6</RNLength>
  <IgnoreNC>0</IgnoreNC>
  <VerifyState>0</VerifyState>
  <GenerateCode>1</GenerateCode>
  <BeforeScript/>
  <AfterScript/>
  <Notes/>
  <Comments/>
  <Caption>Medium</Caption>
  <Size>0</Size>
  <DataWarehouseType>0</DataWarehouseType>
- <Relationship>
- <ReferencedObject>
  <Type>Relation</Type>
  <Id>{35BFD099-EBA9-4422-AE0C-14919A4D2BEC}</Id>
  <Name>is_available_on</Name>
  <FullName>is_available_on</FullName>
  <ReferencedObject/>
</Relationship>
- <Attributes>

```

For creation of new XSL templates, it is useful to work with XSD files. XSD files describe structure of XML source file and helps you to understand the XML structure, what nodes may appear in XML and so on. You can click **Show XSD File** or **Save XSD File As** to display or save XSD file for your simplified XML structure.

## Predefined XSL Templates

By default, the following reports or outputs can be generated:

- Comments (HTML)
- Complete XML (XML)
- Entities (CSV)
- Notes (HTML)
- Tablespace (HTML)
- ToDo Report (HTML)

Using these predefined items you can generate report of all ToDo items, create CSV files with information about entity names, captions and descriptions etc.

## Path to XSLT files

There are two folders where XSLT files are stored.

1. System folder: C:\Program Files\Quest Software\Toad Data Modeler 3\XSL
2. User-defined files: C:\Documents and Settings\\My Documents\Toad Data Modeler\\XSL

## Sample XSLT File

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:tdm="http://www.quest.com/toad-data-modeler" xmlns:msxsl="urn:schemas-microsoft-com:xslt">
  <tdm:description>Output contains Entity Name, Caption and Comments. You can change List Separator. Click
  MORE and change value in the Parameter Value column.</tdm:description>
  <tdm:caption>Entities (CSV)</tdm:caption>
  <tdm:default-filename>entities</tdm:default-filename>
  <tdm:default-suffix>csv</tdm:default-suffix>
  <tdm:db-platforms>
    <tdm:db-platform>all</tdm:db-platform>
  </tdm:db-platforms>
  <msxsl:script language="JScript" implements-prefix="tdm">
    function closeInApostrophes( nodelist ) {
      var text = nodelist.nextNode().text;
      return( '' + text.replace( /\'/g, '""' ) + '' );
    }
  </msxsl:script>
  <xsl:output method="text" />
  <xsl:param name="ListSeparator" select="," />
  <xsl:template match="/">
    <xsl:for-each select="Model/Entities/Entity">
      <xsl:if test="position() = 1">
        <xsl:text>Entity Name</xsl:text>
        <xsl:value-of select="$ListSeparator" />
        <xsl:text>Entity Caption</xsl:text>
        <xsl:value-of select="$ListSeparator" />
        <xsl:text>Entity Description</xsl:text>
        <xsl:text>&#10;</xsl:text>
      </xsl:if>
      <xsl:value-of select="tdm:closeInApostrophes(Name)" />
      <xsl:value-of select="$ListSeparator" />
      <xsl:value-of select="tdm:closeInApostrophes(Caption)" />
      <xsl:value-of select="$ListSeparator" />
      <xsl:if test="(Comments)!=''">
        <xsl:value-of select="tdm:closeInApostrophes(Comments)" />
      </xsl:if>
      <xsl:text>&#10;</xsl:text>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

## Special instructions, parameters

In all XSLT files there must be the following special instructions:

```
<tdm:description>Description</tdm:description>
<tdm:caption>Caption (CSV) - will appear in combo box</tdm:caption>
<tdm:default-filename>output-file-name</tdm:default-filename>
<tdm:default-suffix>html</tdm:default-suffix>
<tdm:db-platforms>
  <tdm:db-platform>all</tdm:db-platform>
</tdm:db-platforms>
```

If you want to create a template only for selected target databases, use the following:

```
<tdm:db-platforms>
  <tdm:db-platform>OR9</tdm:db-platform>
  <tdm:db-platform>OR10</tdm:db-platform>
  <tdm:db-platform>OR11</tdm:db-platform>
</tdm:db-platforms>
```

Parameters specified this way:

```
<xsl:param name="GenerateModelInfo" select="True" />
```

will display in the **Input Values** table of the **XSL Transformation** dialog.

## Edit Existing XSLT Templates

1. Copy the system XSLT template to your user section. Default locations are:

System: C:\Program Files\Quest Software\Toad Data Modeler 5.0\XSL

User: C:\Documents and Settings\\My Documents\Toad Data Modeler\\XSL

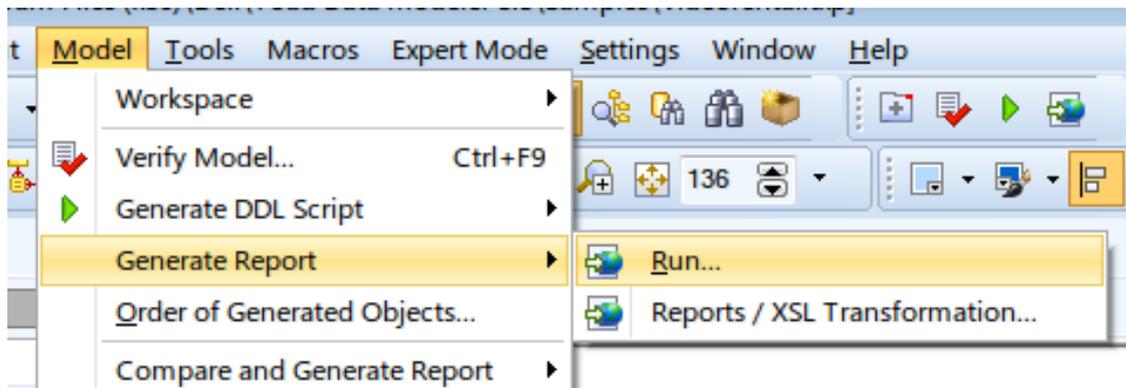
2. Edit the code then.

## HTML Reports

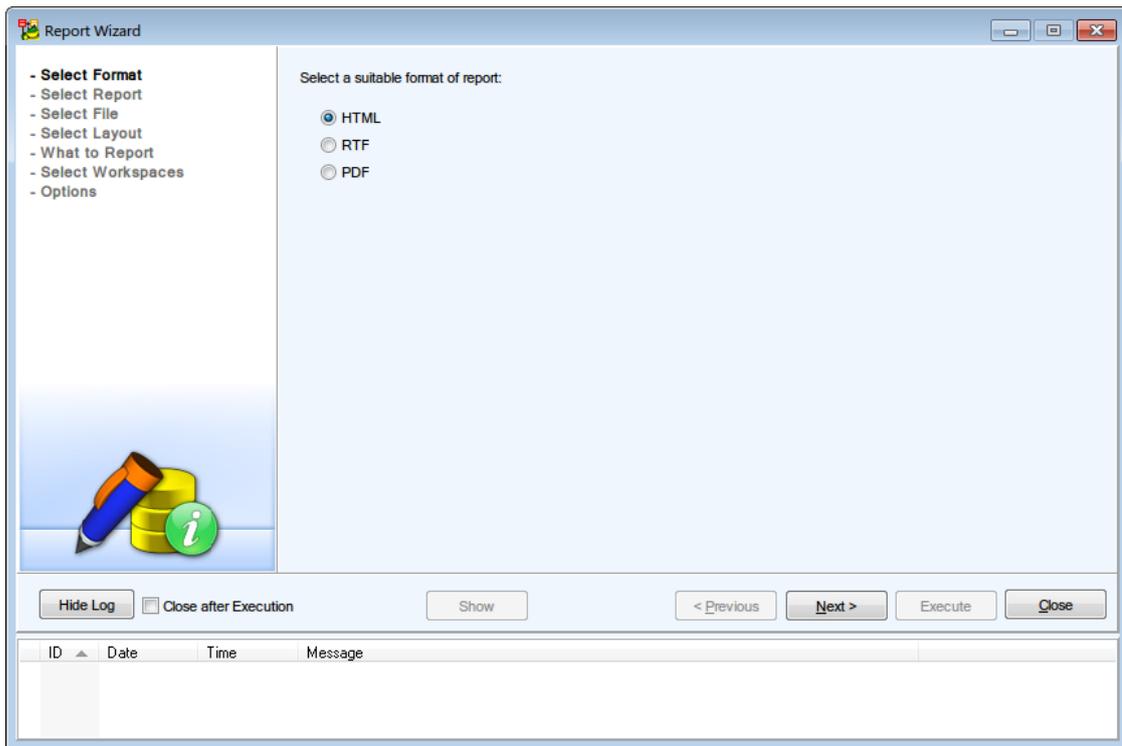
HTML report is great for viewing information about your model objects in an interactive matter. HTML report can also contain images of workspaces, as opposed to RTF/PDF reports. It is also the only type of report you can generate for **Metamodels**.

To generate a HTML report:

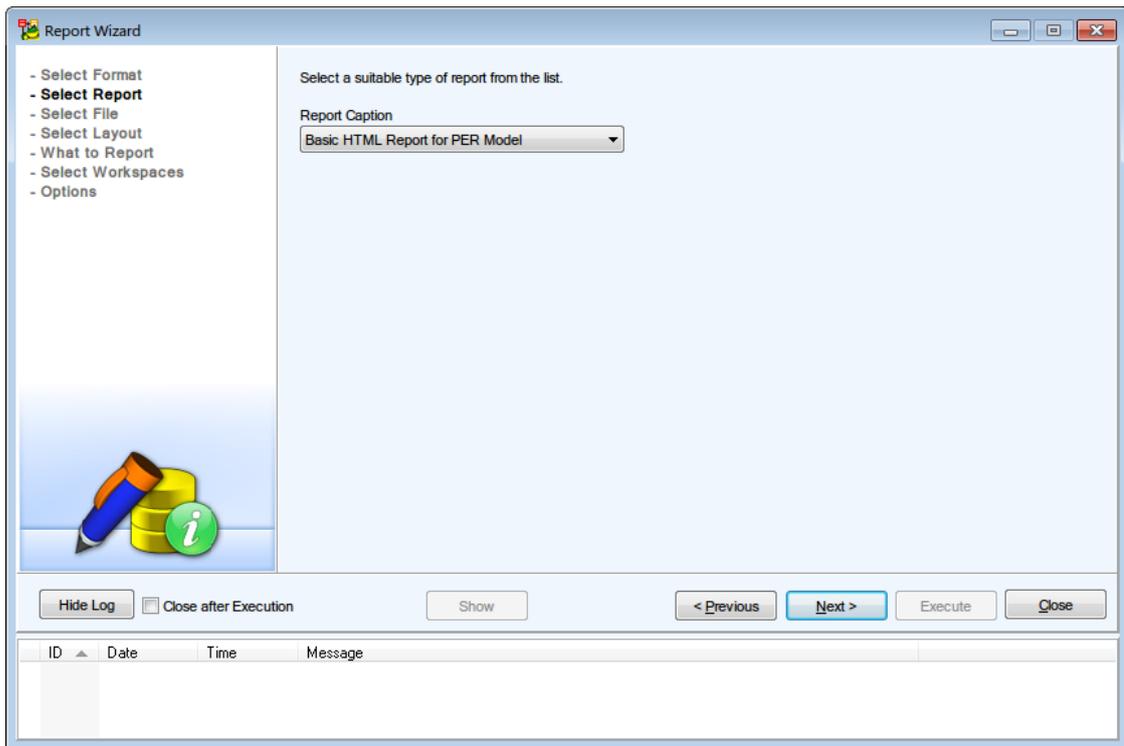
1. Click  on **Model Toolbar** (or go to **Model Menu | Generate Report | Run**).



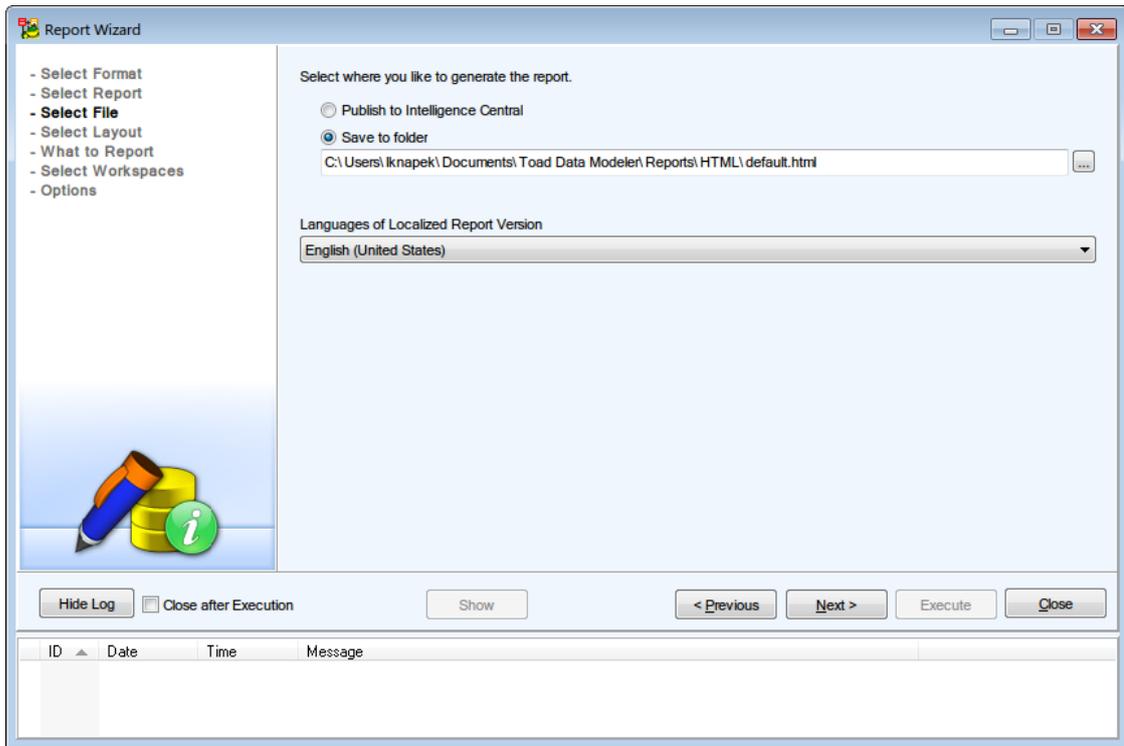
2. In **Select Format** section, choose **HTML** format.



3. In the next section, select the type of the **HTML** report.

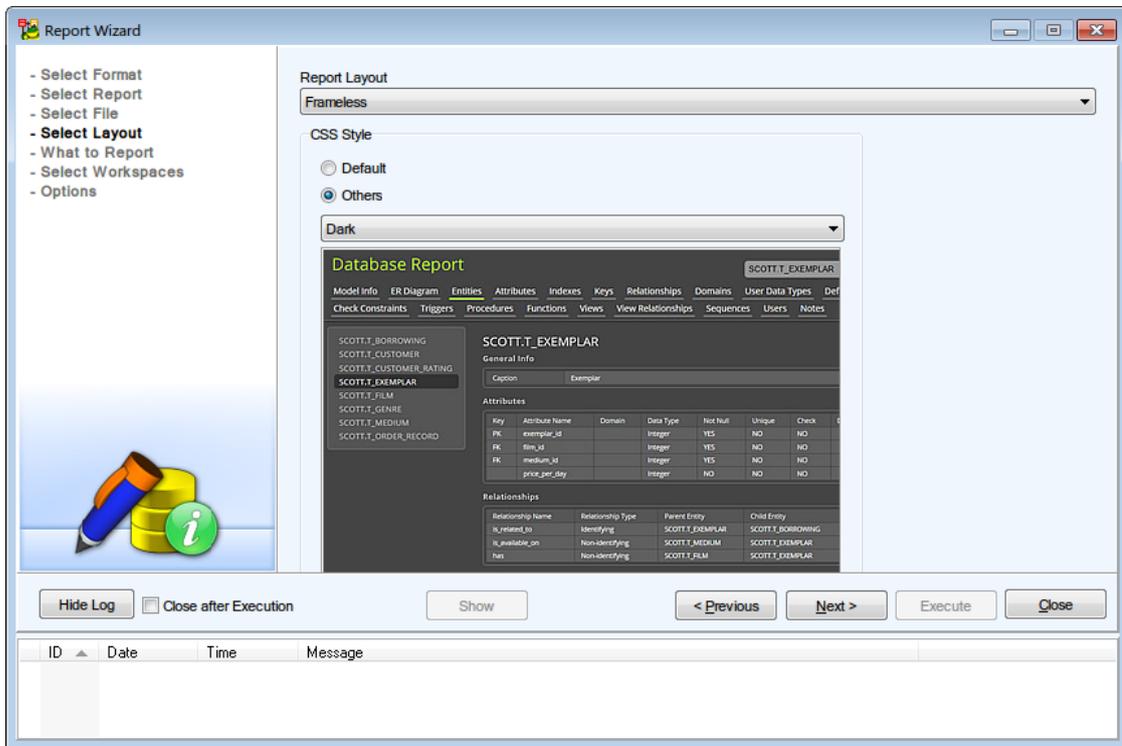


4. In **Select File** section, select the report location. You can either save the report to a folder, or you can publish it to a **Toad Intelligence Central (TIC)** server (see [Basic TIC Actions](#) for more information).

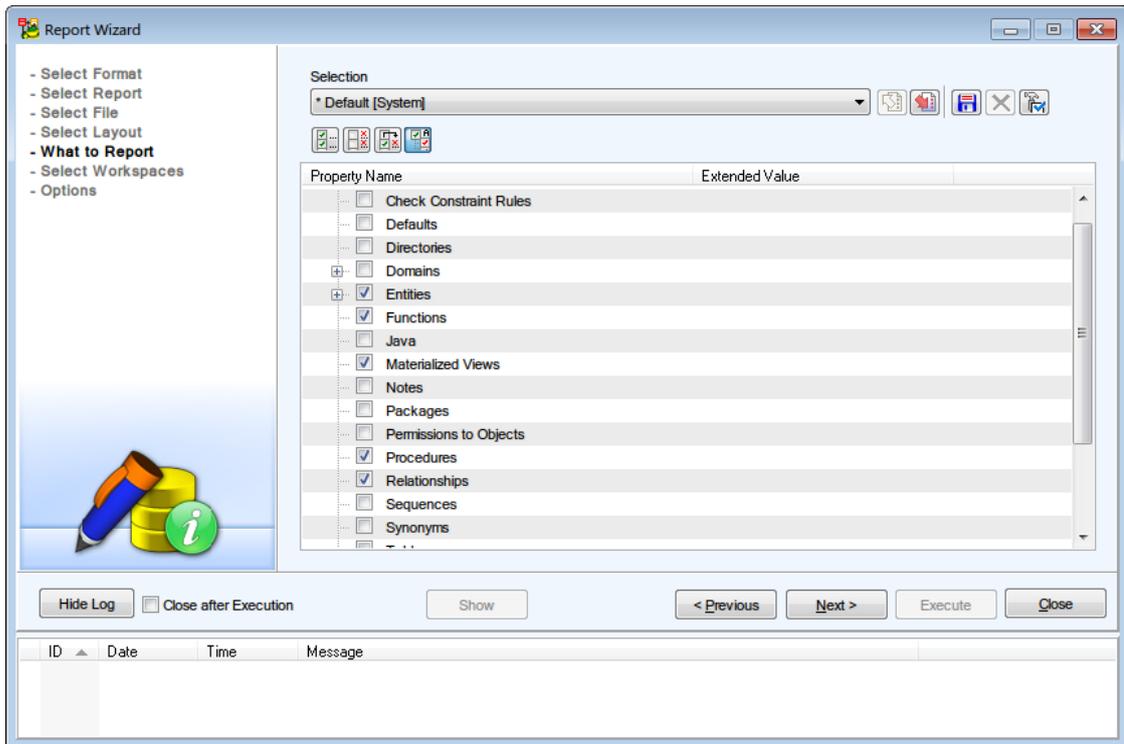


**i** Tip: You can generate localized reports. Download the appropriate language package from the [community website](#) and import it to Toad Data Modeler. See [Dictionaries](#) for more information.

- In the next section, you can choose the **Report Layout**. If you generate report for a large model, you should choose **Frames - Top/Left** menu. Generating **Frameless** reports consumes more RAM and viewing the report might feel sluggish on lower-spec PCs. Other than that, the layout choice is a purely cosmetic one.



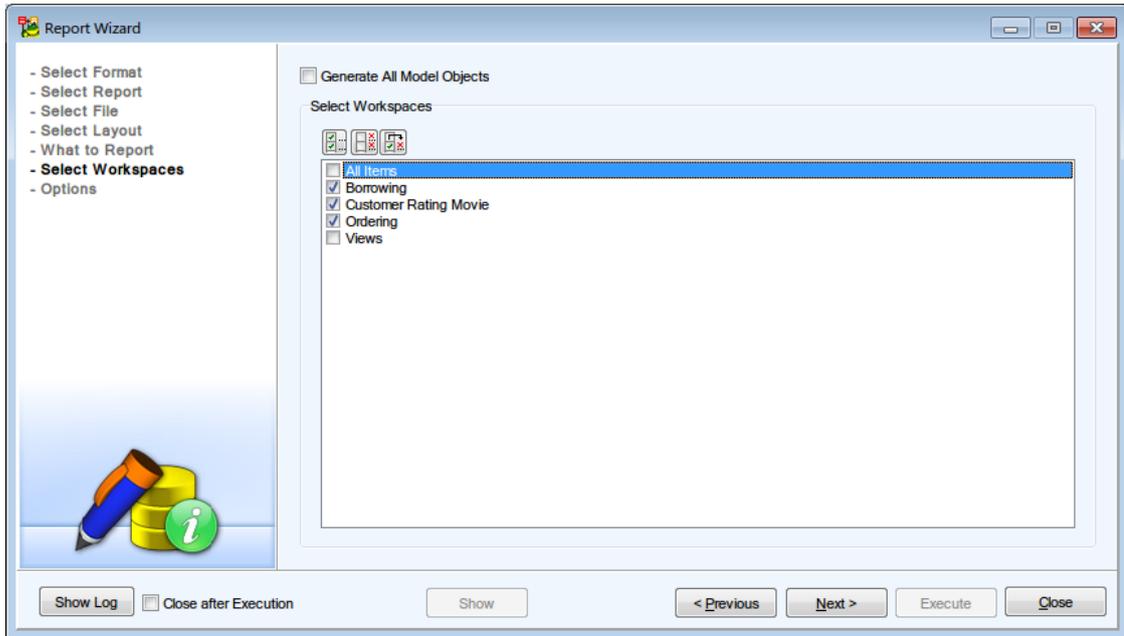
6. The **What to Report** section allows you to select specific object types that should be included in the report.



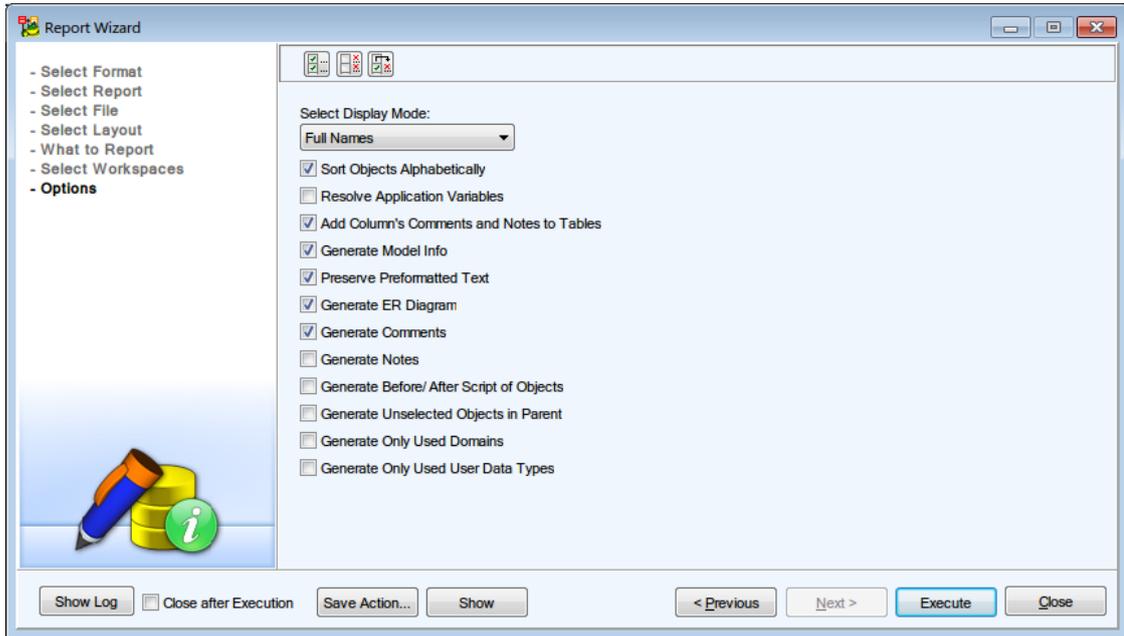
7. **Select Workspaces** section allows to you specify for which workspaces the report should be generated.

**i** Note: The report will be generated for model objects based on the following rules:

- When you check a specific workspace, all of its objects and an image of the workspace will be included in the report.
- When you check **Generate All Model Objects**, all model objects will be included in the report.
- Both rules respect the selection made in **What to Report** section.



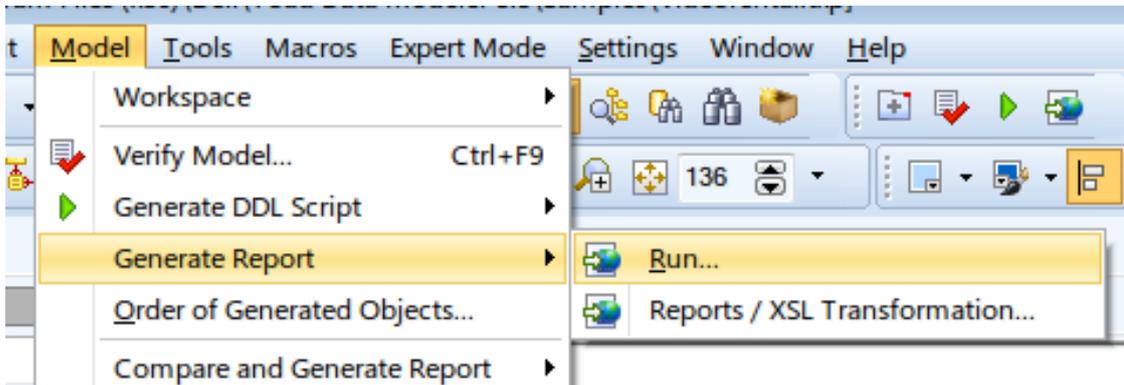
- The final section contains several options allowing you to further customize the generated report. Click on **Execute** to generate the report. Once the report is generated, you can view it by clicking on **Show**.



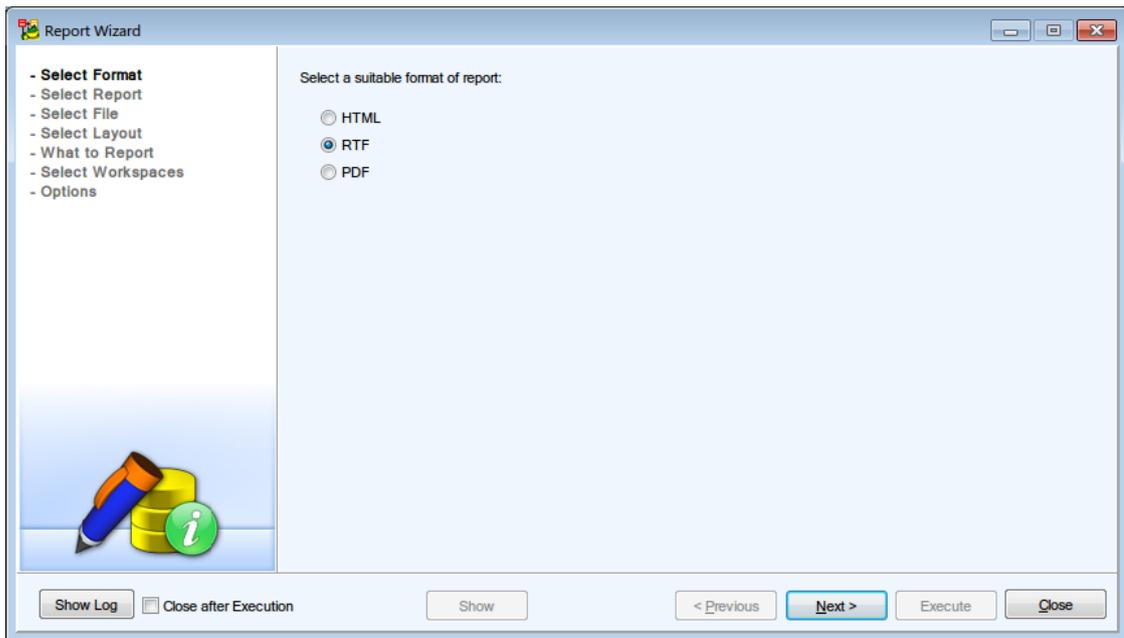
## RTF Reports

To generate a RTF report:

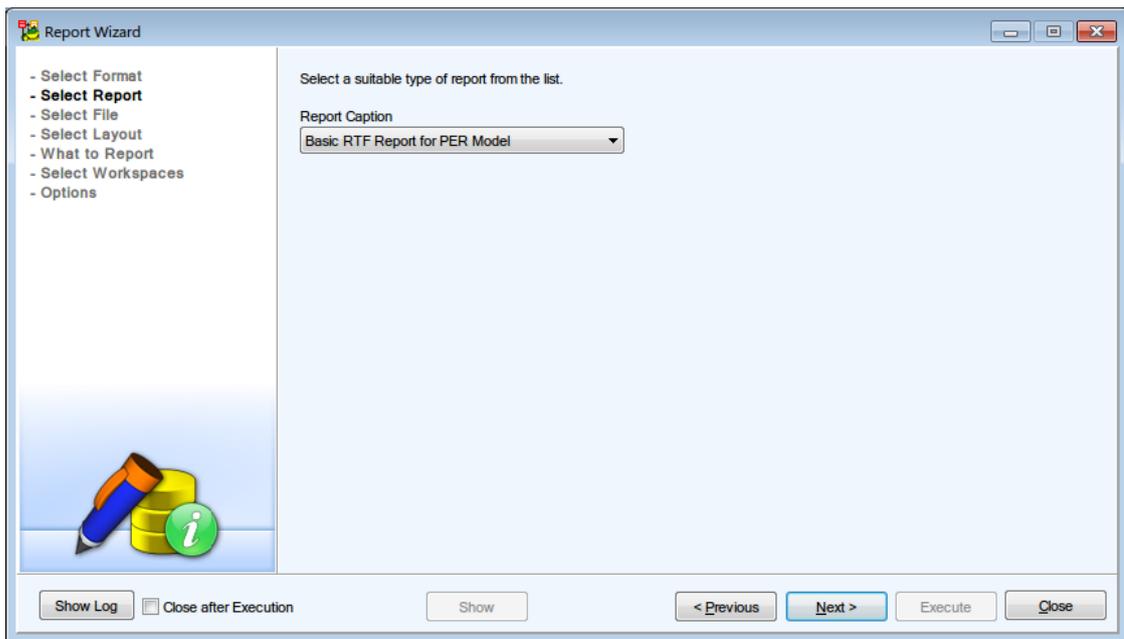
- Click  on **Model Toolbar** (or go to **Model Menu | Generate Report | Run**).



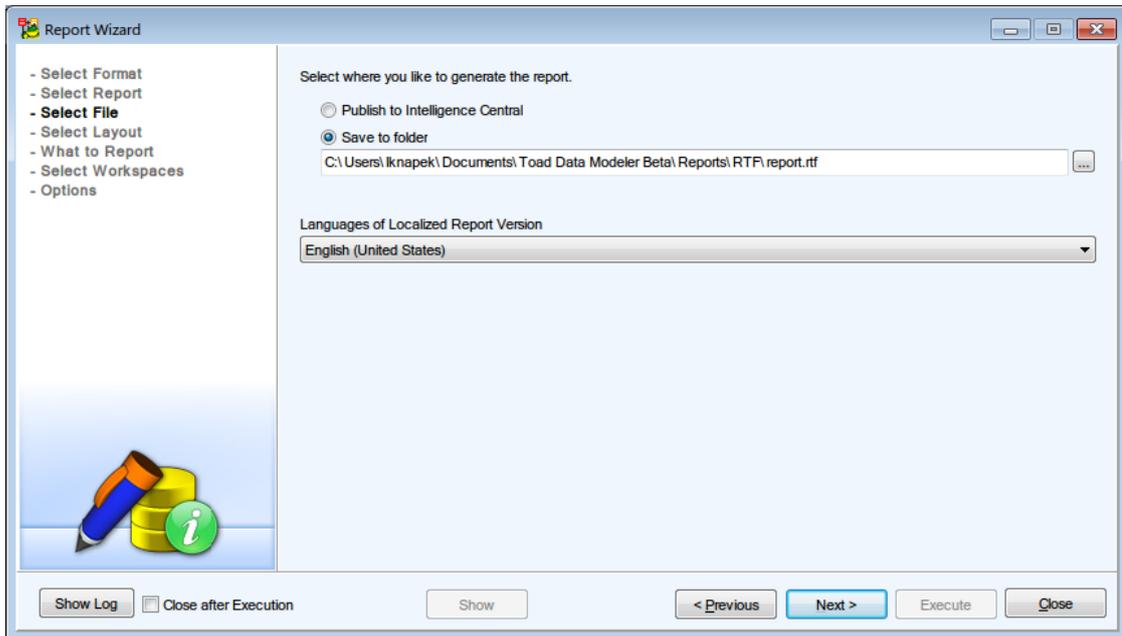
2. In **Select Format** section, choose **RTF** format.



3. In the next section, select the type of the **RTF** report.

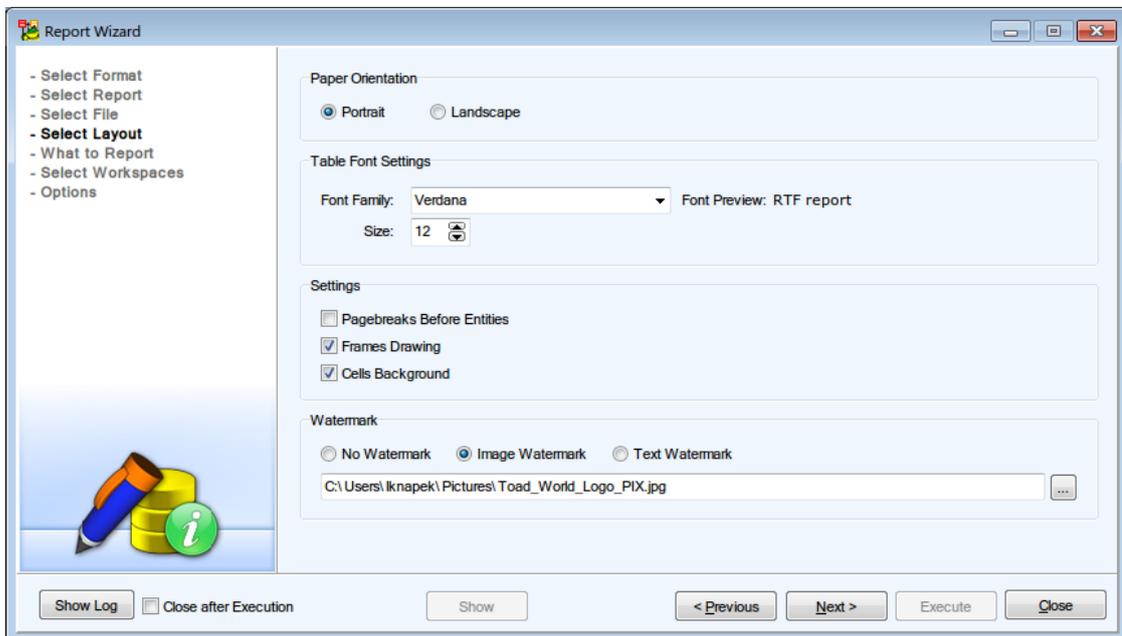


- In **Select File** section, select the report location. You can either save the report to a folder, or you can publish it to a **Toad Intelligence Central (TIC)** server (see [Basic TIC Actions](#) for more information).

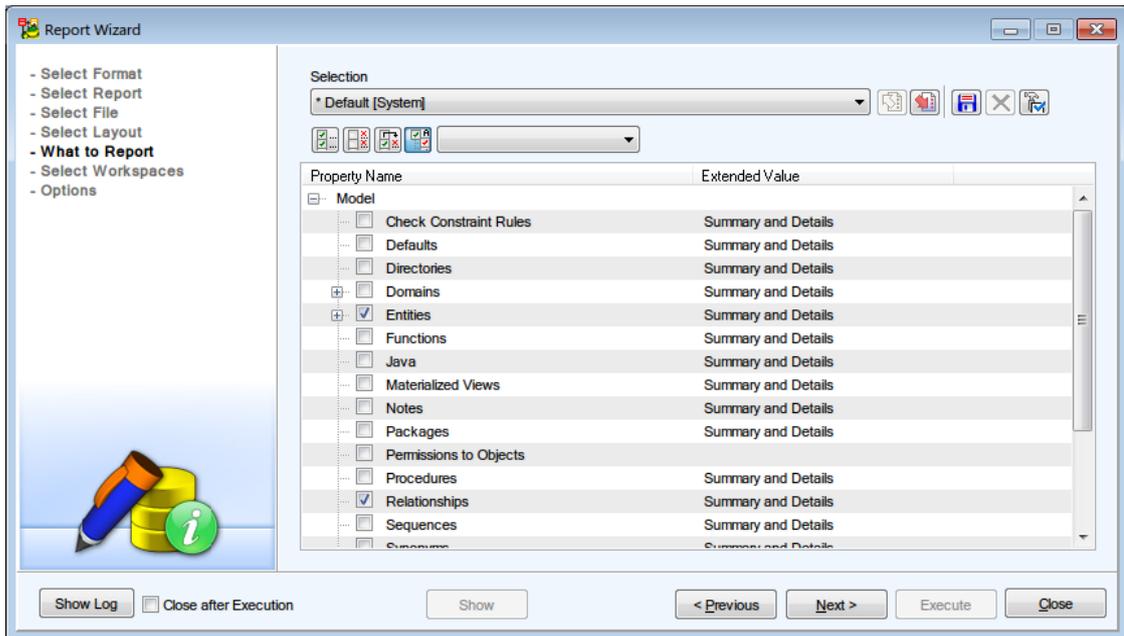


**i** Tip: You can generate localized reports. Download the appropriate language package from the [community website](#) and import it to Toad Data Modeler. See [Dictionaries](#) for more information.

- The next section allows you to customize the look of your report. You can change the report **Orientation**, **Font Settings** and add **Watermark**.



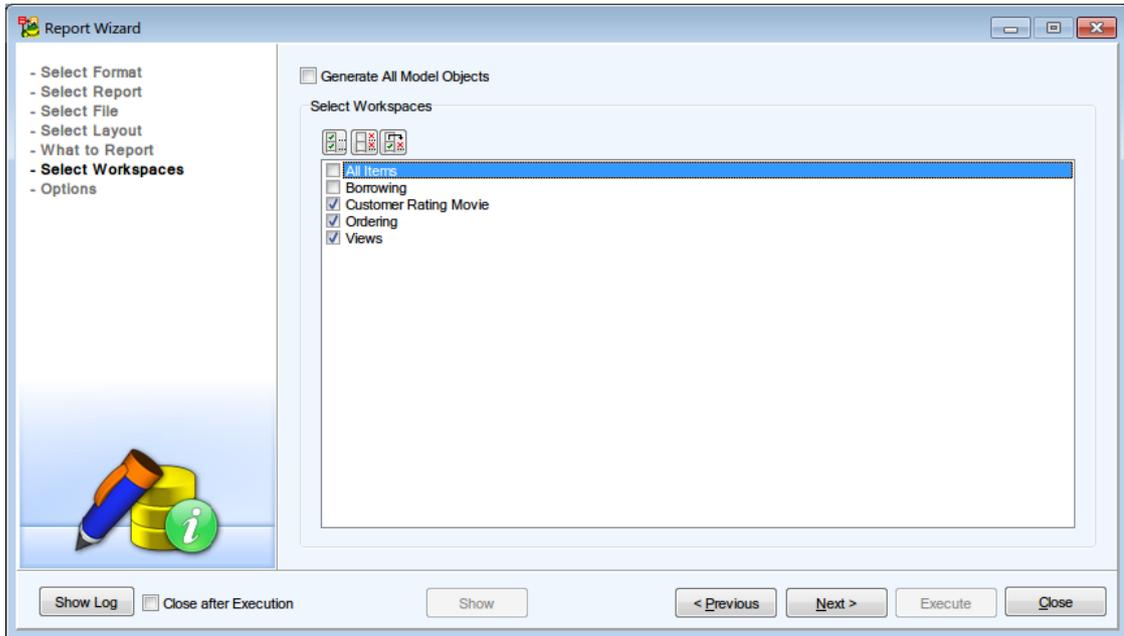
6. The **What to Report** section allows you to select specific object types that should be included in the report.



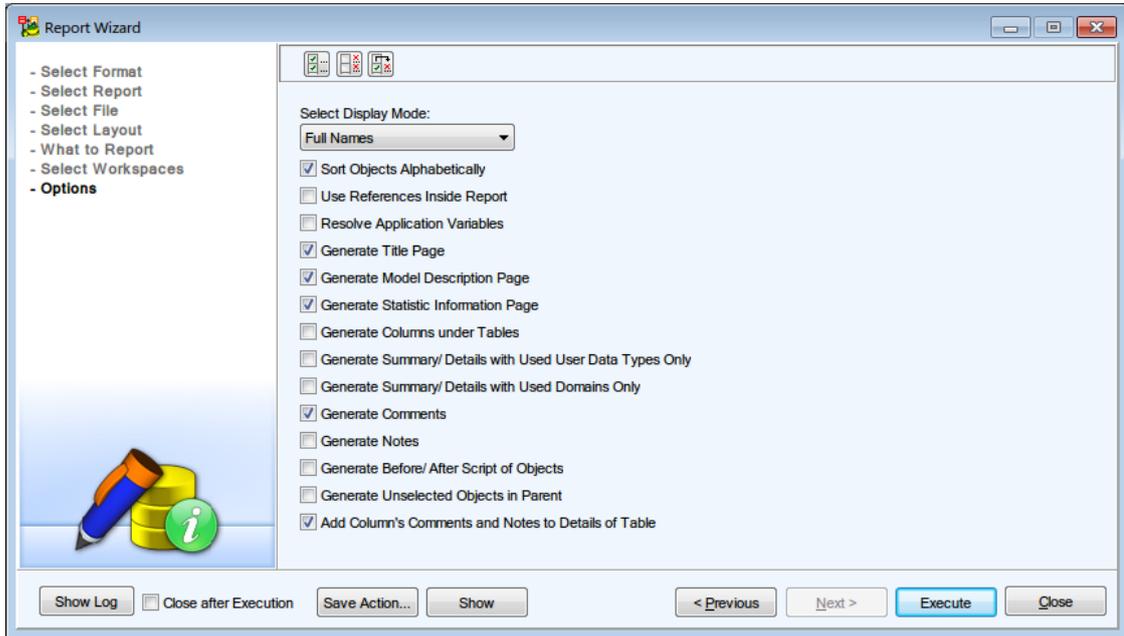
7. **Select Workspaces** section allows to you specify for which workspaces the report should be generated.

**i** Note: The report will be generated for model objects based on the following rules:

- When you check a specific workspace, all of its objects and an image of the workspace will be included in the report.
- When you check **Generate All Model Objects**, all model objects will be included in the report.
- Both rules respect the selection made in **What to Report** section.



- The final section contains several options allowing you to further customize the generated report. Click on **Execute** to generate the report. Once the report is generated, you can view it by clicking on **Show**.

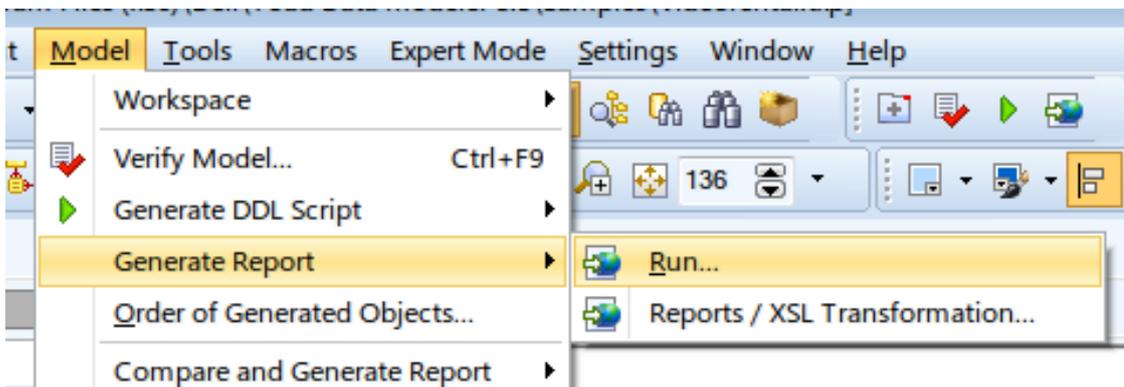


## PDF Reports

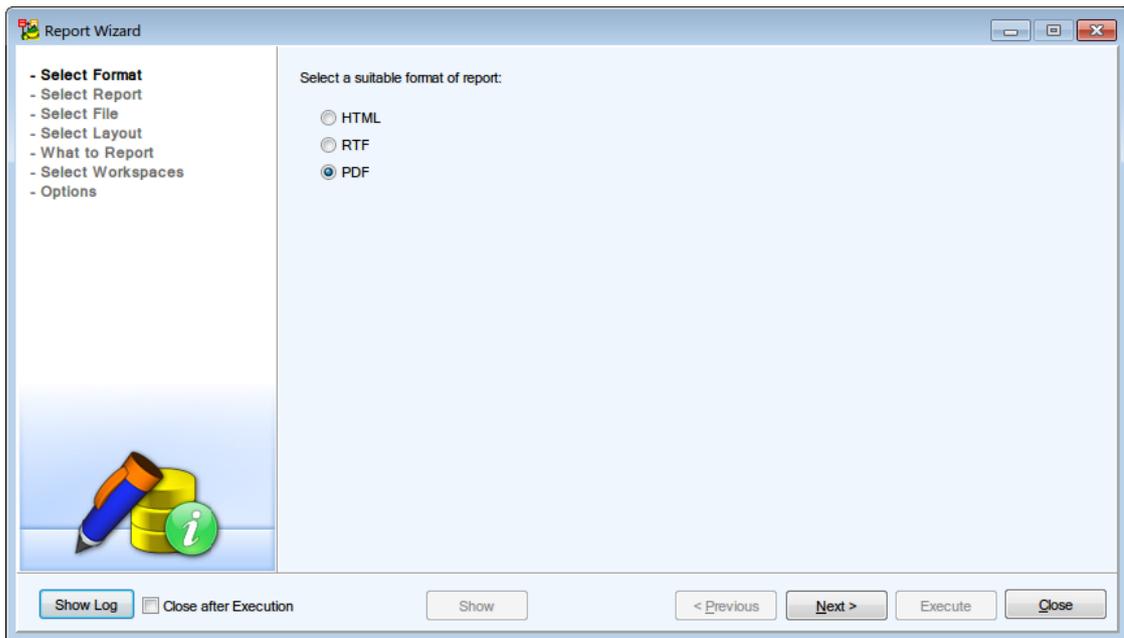
PDF report generation is similar to RTF report generation.

To generate a RTF report:

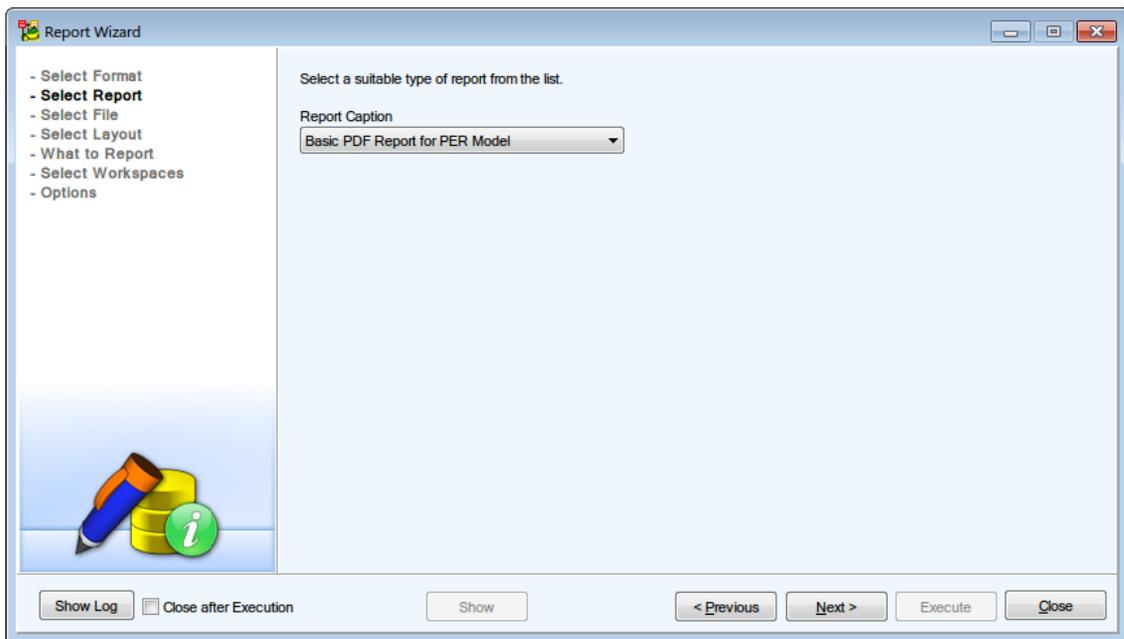
- Click  on **Model Toolbar** (or go to **Model Menu | Generate Report | Run**).



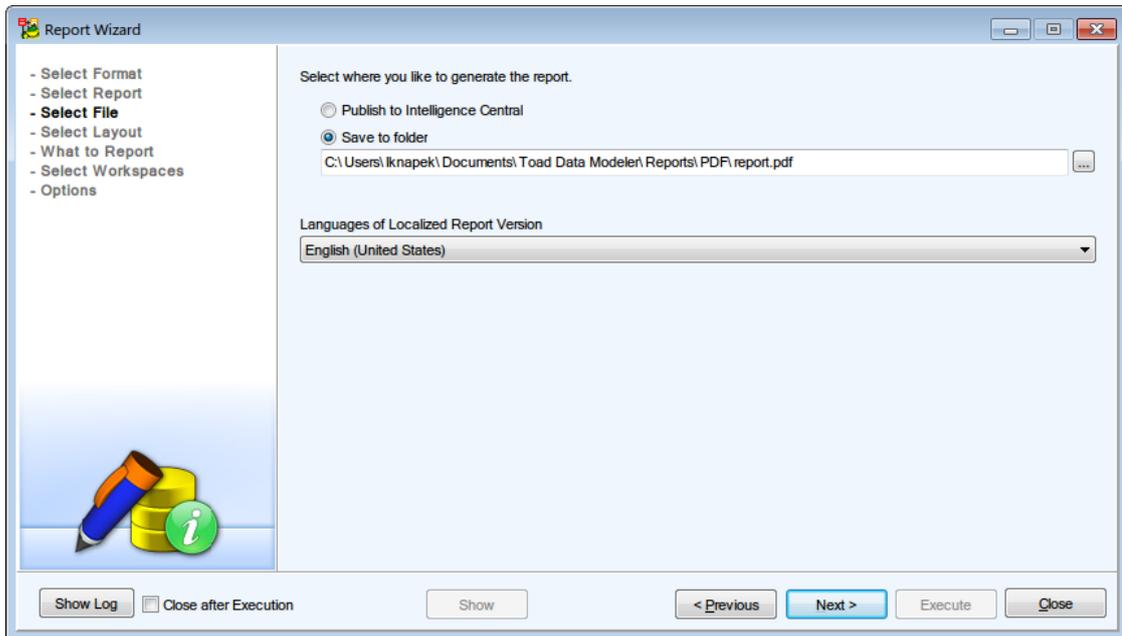
2. In **Select Format** section, choose **PDF** format.



3. In the next section, select the type of the **PDF** report.

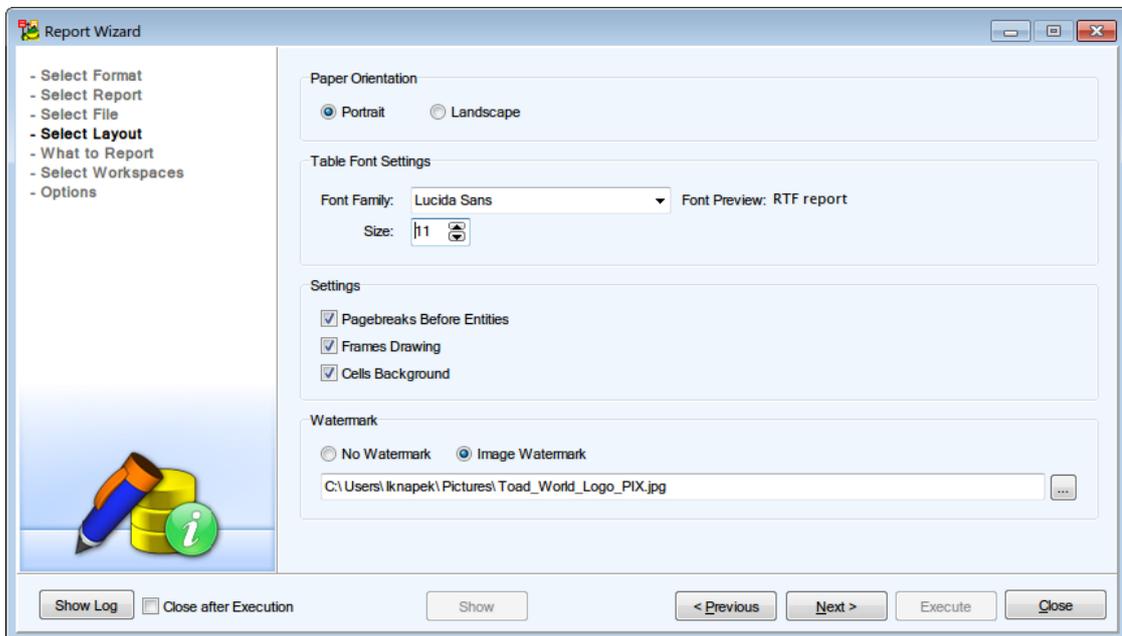


4. In **Select File** section, select the report location. You can either save the report to a folder, or you can publish it to a **Toad Intelligence Central (TIC)** server (see [Basic TIC Actions](#) for more information).

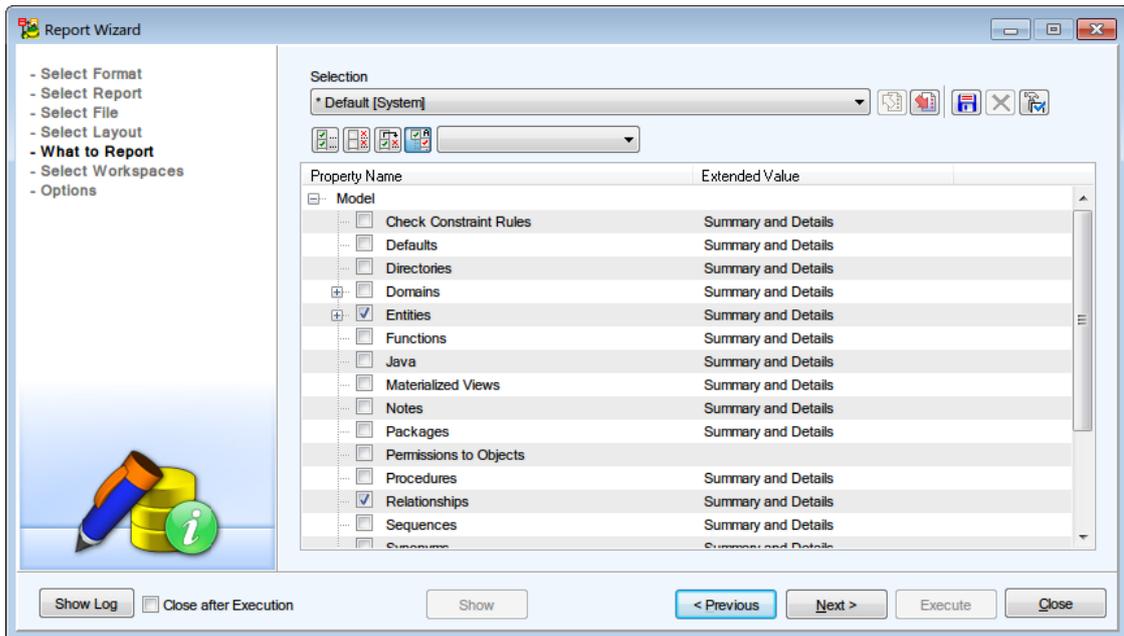


**i** Tip: You can generate localized reports. Download the appropriate language package from the [community website](#) and import it to Toad Data Modeler. See [Dictionaries](#) for more information.

5. The next section allows you to customize the look of your report. You can change the report **Orientation**, **Font Settings** and add **Watermark**.



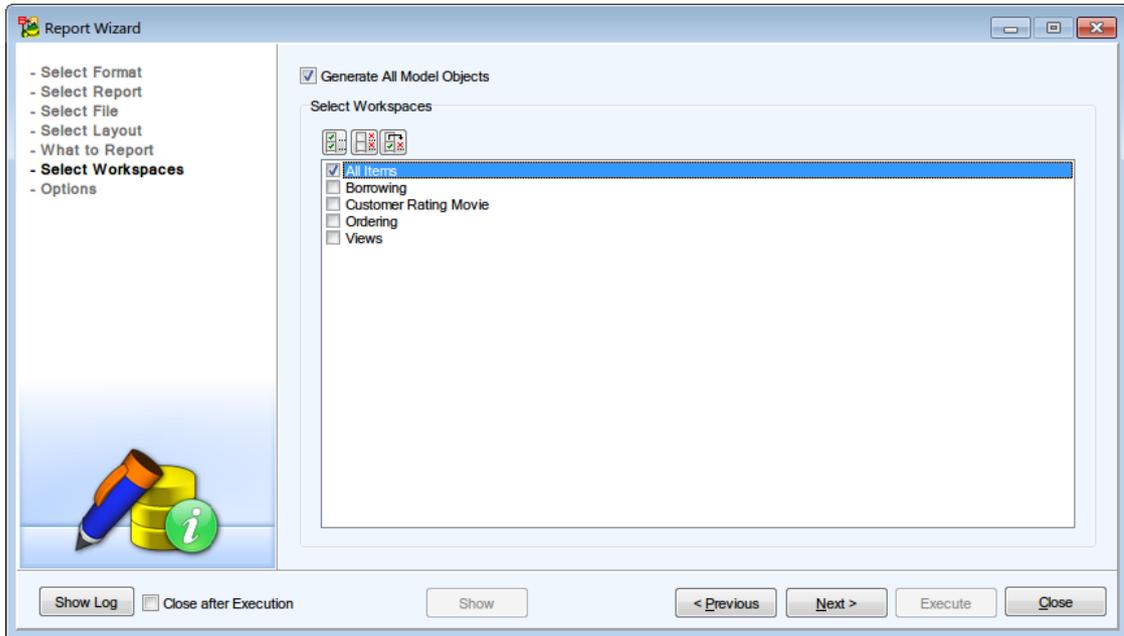
6. The **What to Report** section allows you to select specific object types that should be included in the report.



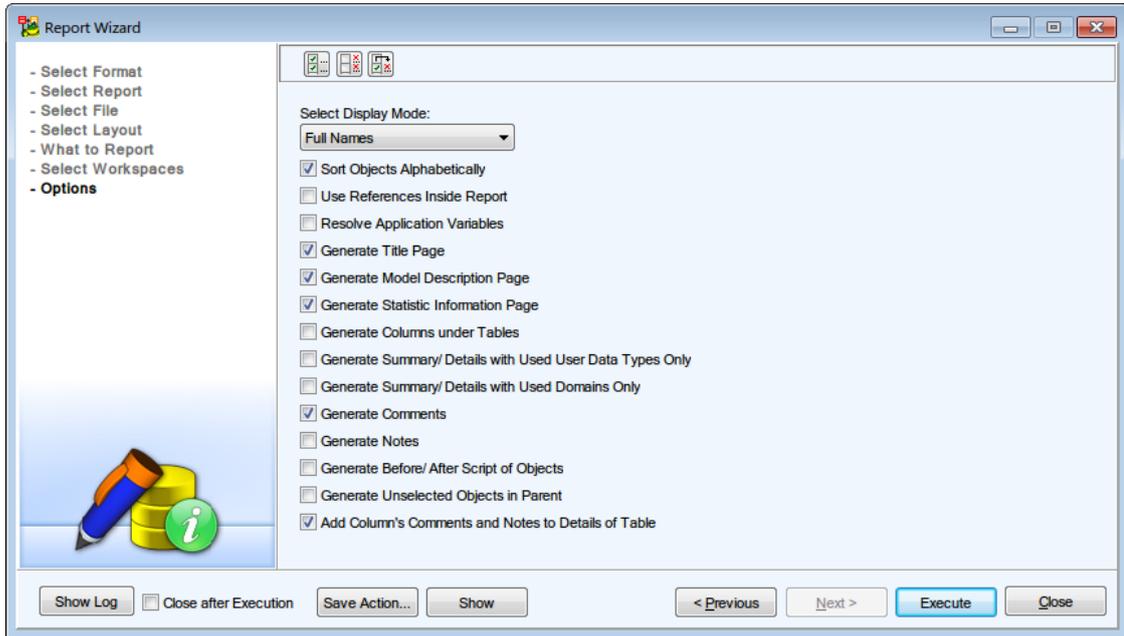
7. **Select Workspaces** section allows to you specify for which workspaces the report should be generated.

**i** Note: The report will be generated for model objects based on the following rules:

- When you check a specific workspace, all of its objects and an image of the workspace will be included in the report.
- When you check **Generate All Model Objects**, all model objects will be included in the report.
- Both rules respect the selection made in **What to Report** section.



- The final section contains several options allowing you to further customize the generated report. Click on **Execute** to generate the report. Once the report is generated, you can view it by clicking on **Show**.



## Connections

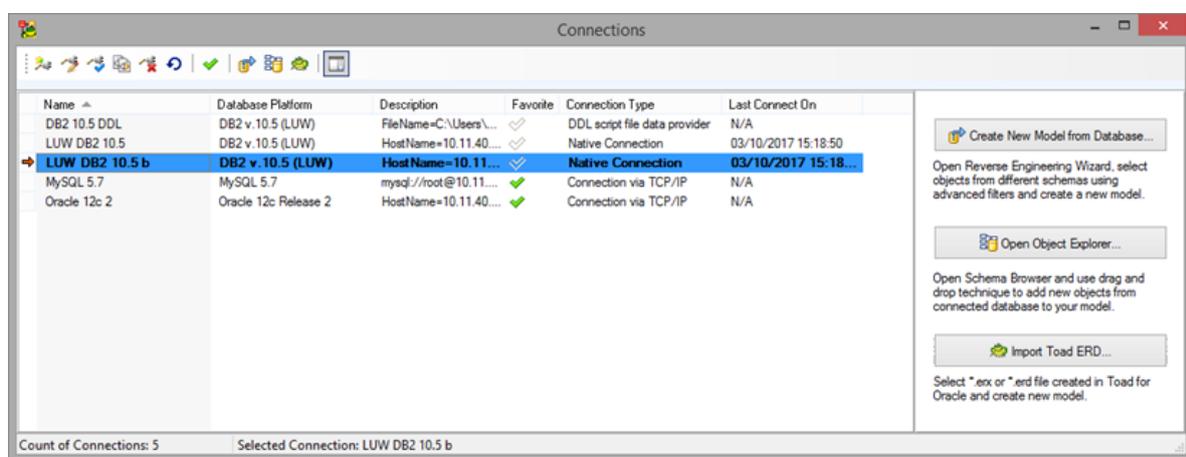
**Connections** displays all connections to databases and to DDL files. You can easily create and edit connections and also launch **Reverse Engineering** wizard or perform operations with models.

### To manage connections

- Select **File | Reverse Engineering | Connections**.

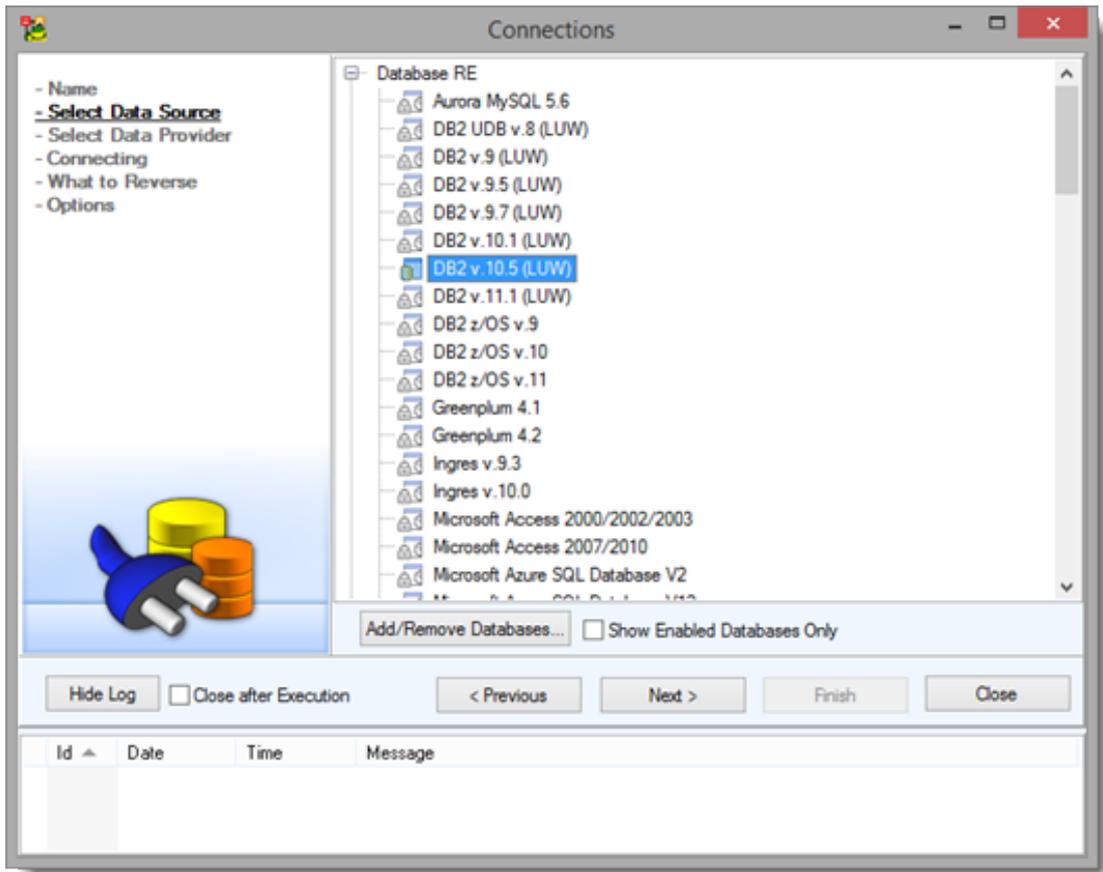
Button	Description
<b>New Connection</b>	Create a new connection to a database or to a DDL script file.
<b>Edit Connection</b>	Edit existing database or DDL file connection.
<b>Test Connection</b>	Tests a connection.
<b>Copy Connection</b>	Clones a connection. Creates a connection based on an existing one.
<b>Delete Connection</b>	Deletes a connection.
<b>Reload Connections</b>	Refreshes the list of connections. If you manually add a connection to Connections file this list needs to be refreshed. The path to Connections file is defined in <b>Settings   Options   Reverse</b>

Button	Description
<b>Engineering   Path to Connections.</b>	
<b>Show Favorites Only</b>	Only displays your favorite connections. Double-click into Favorite column to mark a connection as favorite.
<b>Create New Model from Database</b>	Launches <b>Reverse Engineering</b> wizard.
<b>Open Object Explorer</b>	Launches <b>Object Explorer</b> to drag and drop objects into your model from the connection.
<b>Import Toad ERD</b>	Creates a new model based on *.erx or *.erd file from Toad.
<b>Show Right Bar</b>	Displays a right-hand bar with extra buttons.



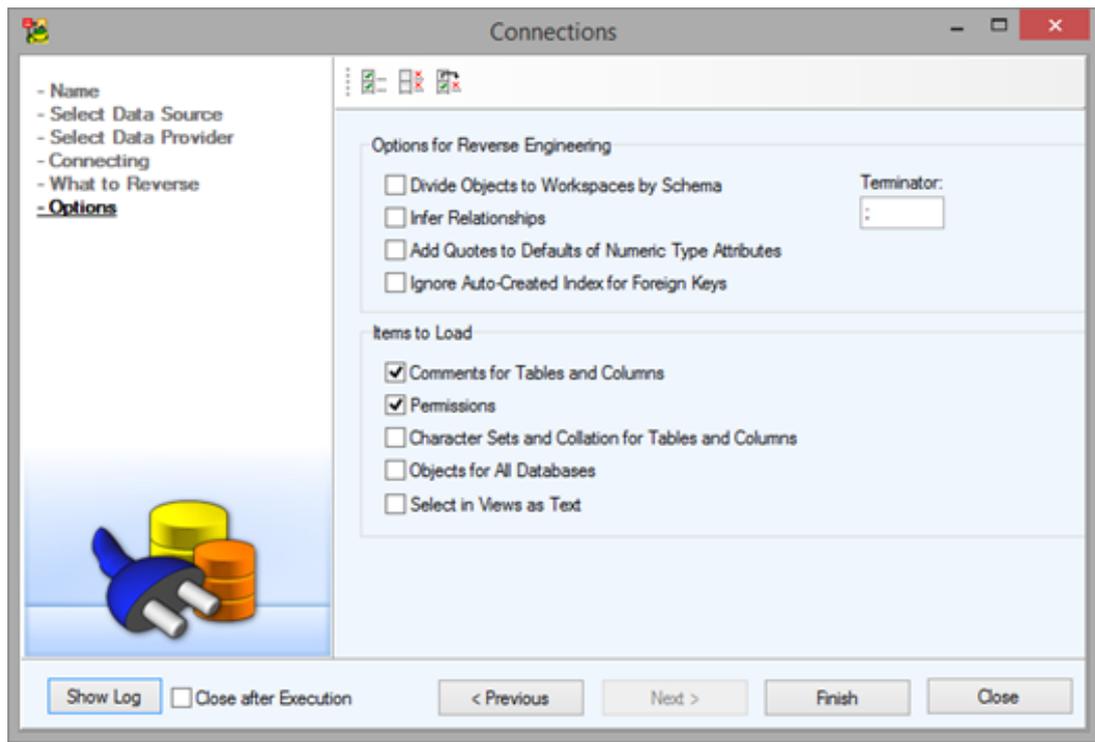
### To create a new database connection

1. Click  **New Connection** and define a new connection name.
2. Select data source. Select the desired type of database RE. See the [Supported Databases](#) for more info on Toad Data Modeler support for **Reverse Engineering**. Click **Add/Remove Databases** to enable or disable databases. Check **Show Enabled Databases** to only show your enabled database types.



3. Select the data provider for your connection to a database. See more information on the connection methods available for specific databases in [Types of Connections by Databases](#).
4. Enter your connection information or edit your connection string directly.
5. Select what you want to perform **Reverse Engineering** with. Use **Auto Check** button to automatically select parent objects when any of their dependent objects are selected.
6. In **Options** configure settings used for **Reverse Engineering**. See **Help | Databases | {specific database} | Reverse Engineering** for more information on available options. The following options for **Reverse Engineering** are common for more database types:

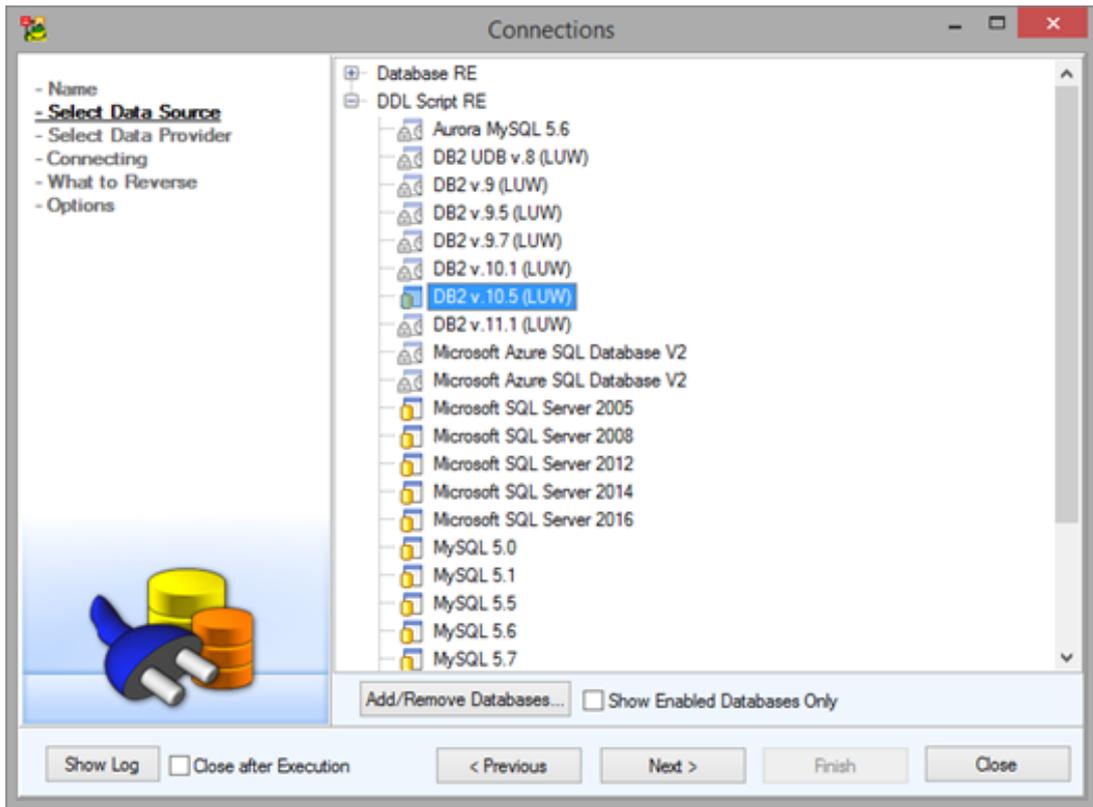
Option	Description
<b>Divide Objects to Workspaces by Schema</b>	The resulting model will have a workspace for each schema.
<b>Infer Relationships</b>	Check to automatically generate relationships between entities in the resulting model. If unchecked you will be prompted during <b>Reverse Engineering</b> only when no relationships are found in the resulting model. <i>Default: Unchecked.</i>
<b>Terminator</b>	Select the desired terminator for SQL statements.



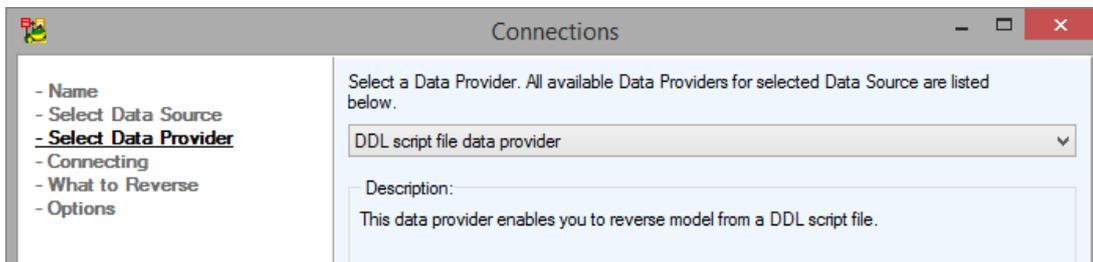
7. Click **Finish** to save the new connection.

### **To create a new DDL file connection**

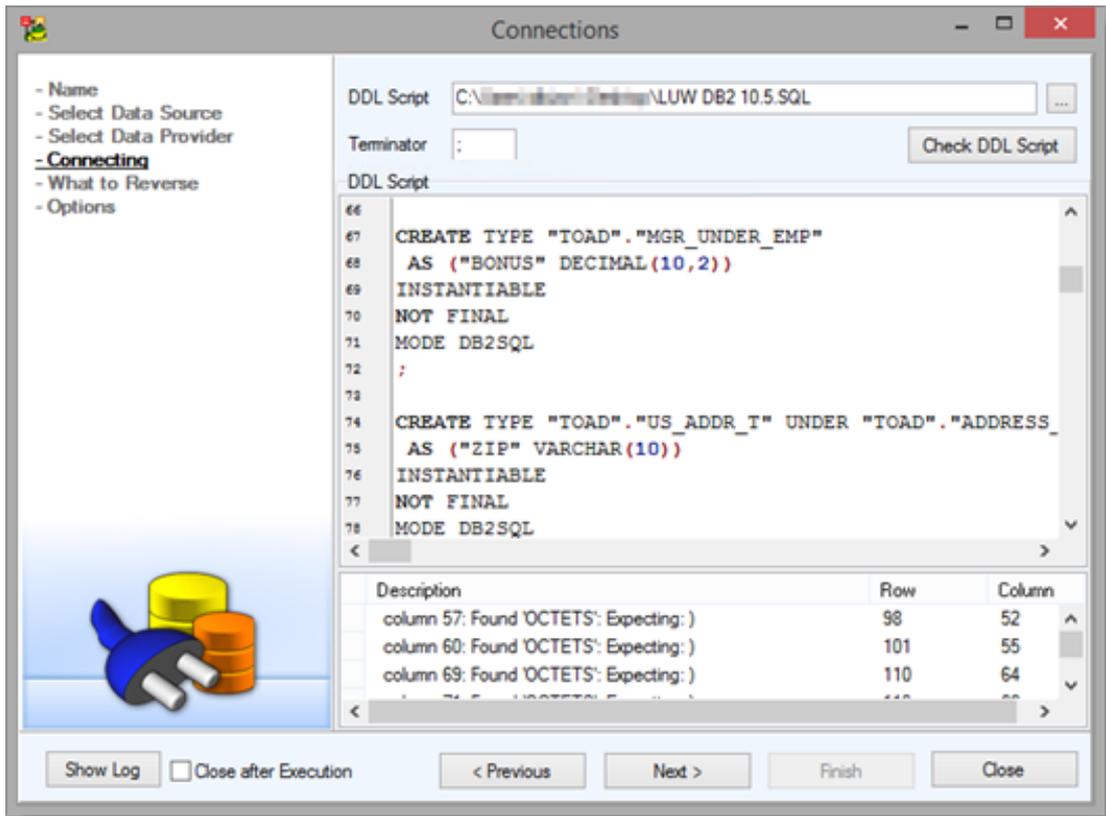
1. Click  **New Connection** and define a new connection name.
2. Select data source. Select the desired type of DDL Script RE. See the [Supported Databases](#) for more info on Toad Data Modeler support for **Import from SQL files**. Click **Add/Remove Databases** to enable or disable databases. Check **Show Enabled Databases** to only show your enabled database types.



3. Select the data provider for your connection to a DDL script file.

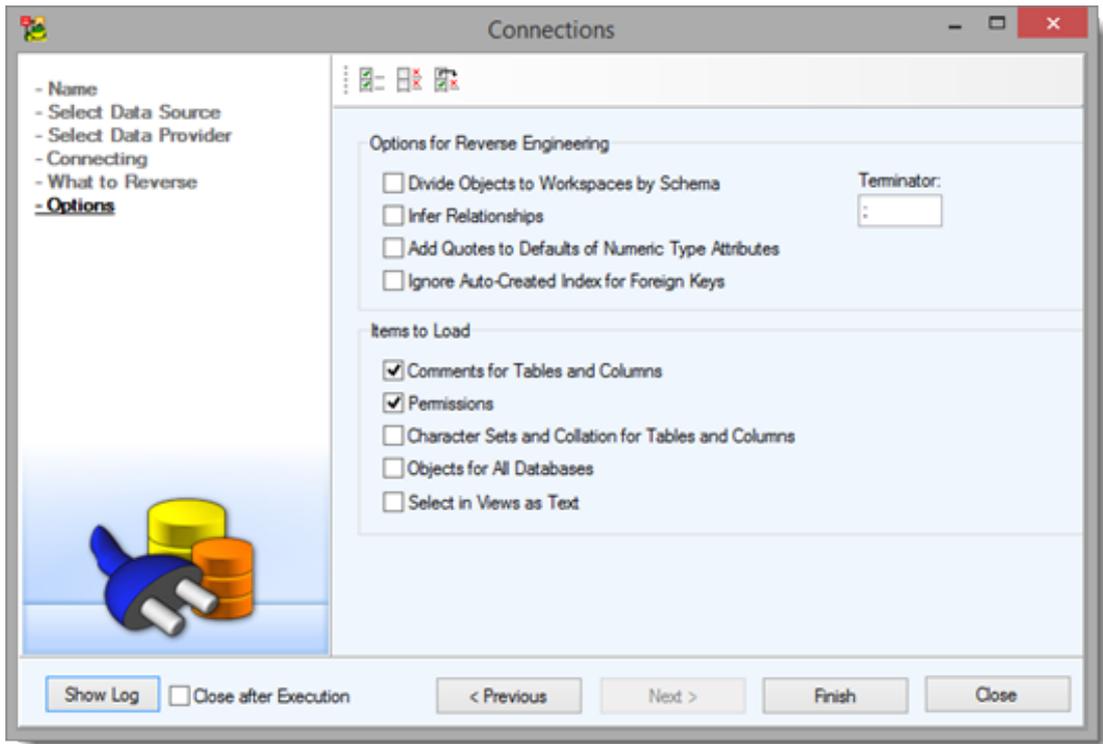


4. Click **Open DDL Script File** and select the file that you want to use for connecting. Enter the desired terminator for SQL statements. The DDL script is displayed after opening the file.



5. Select what you want to perform **Reverse Engineering** with.
6. In **Options** configure settings used for **Reverse Engineering**. See **Help | Databases | {specific database} | Reverse Engineering** for more information on available options. The following options for **Reverse Engineering** are common for more database types:

Option	Description
<b>Divide Objects to Workspaces by Schema</b>	The resulting model will have a workspace for each schema.
<b>Infer Relationships</b>	Check to automatically generate relationships between entities in the resulting model. If unchecked you will be prompted during <b>Reverse Engineering</b> only when no relationships are found in the resulting model. <i>Default: Unchecked.</i>



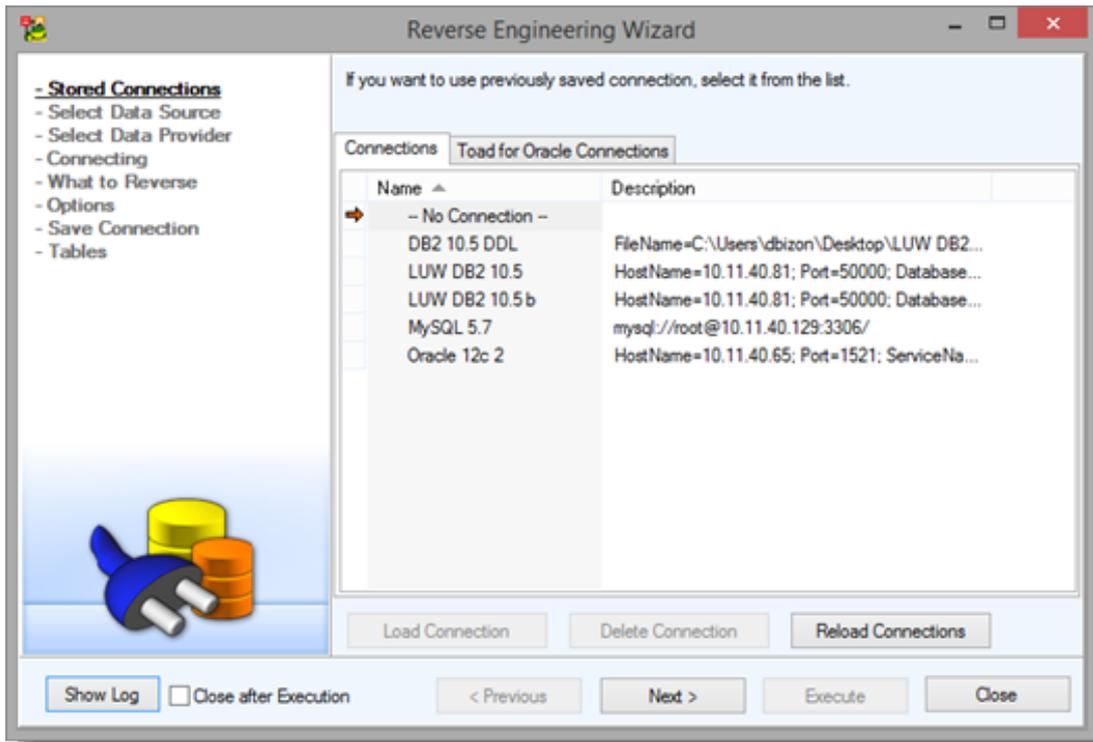
7. Click **Finish** to save the new connection.

## Reverse Engineering Wizard

Reverse Engineering Wizard guides you through the process of creating a model from an existing database or a DDL file.

### *To create a model by Reverse Engineering*

1. Click **File | Reverse Engineering | Reverse Engineering Wizard**. Select one of your stored connections or select "No connection" to define a new source in the following steps of the wizard. You can also select one of the connections shared with Toad for Oracle or created during import of an ERD file from Toad for Oracle. Select a connection and click **Load** to proceed to the last step of the wizard or define a new connection. See [Connections](#) for more information on how to create a connection to a database or to a DDL file.

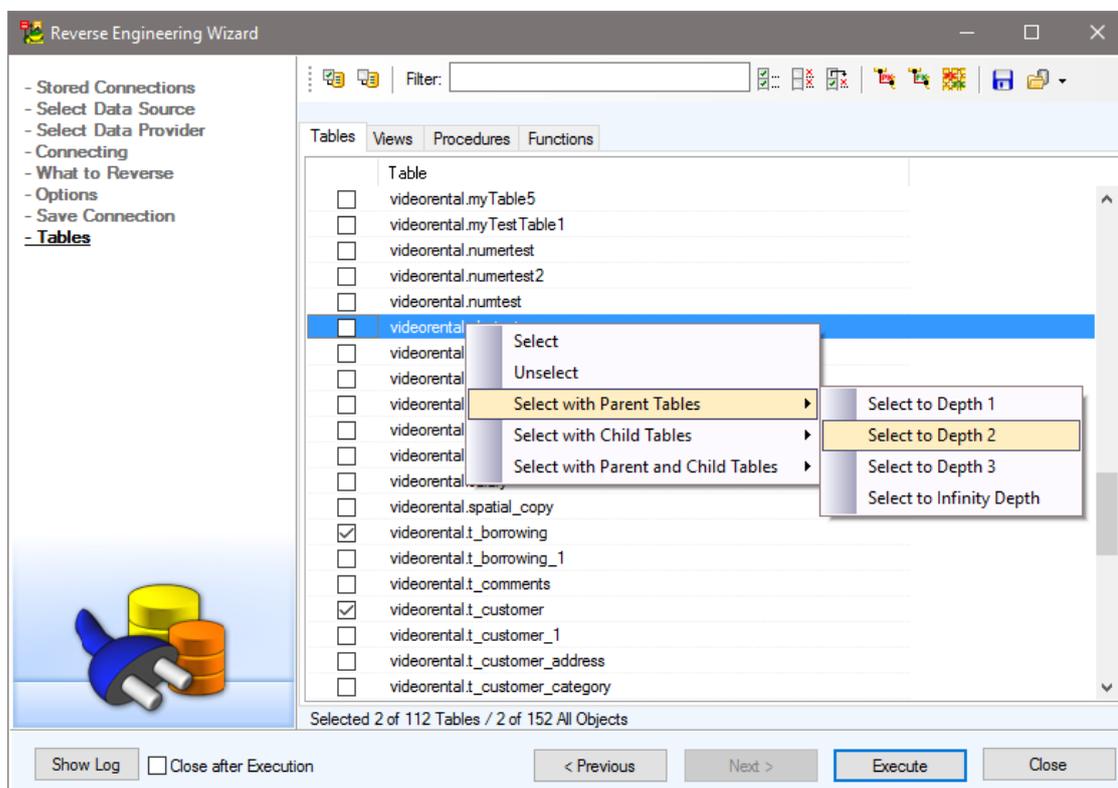


2. Select the objects you want to include in your target model and click **Execute** to create a new model.

Feature	Description
Schema	Select a specific schema you want to include in the target model, all tables or all selected tables.
Select All on All Tabs	Selects all objects on all tabs.
Deselect All on All Tabs	Deselects all objects on all tabs.
Filter	Type to filter objects. <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p><b>i</b> TIP: Use wildcards - example: "*user" or "?ser".</p> <p>* - replaces unlimited number of characters</p> <p>? - replaces any single character</p> </div>
Select All	Selects all object on the current tab.
Deselect All	Deselects all object on the current tab.
Invert Selection	Inverts selection on the current tab.
Select Parent Tables	Selects all parent tables of the currently selected tables.
Select Child Tables	Selects all child tables of the currently selected tables.
Select Parent and Child Tables	Selects all child and all parent tables of the currently selected tables.

Feature	Description
Export Selection	Export selection as a *.wsxr file.
Import Selection	Imports a selection from *.wsxr file.

**TIP:** Select parent and child tables easily using the right-click menu. You can select child and/or parent tables up to 3 levels of depth or all levels (Infinity Depth).



## Object Explorer

Use **Object Explorer** to easily drag and drop objects from your connections.

### To use Drag and Drop method

1. Click **File | Reverse Engineering | Connections**, select a connection and click **Open Object Explorer** to launch it for the selected connection.

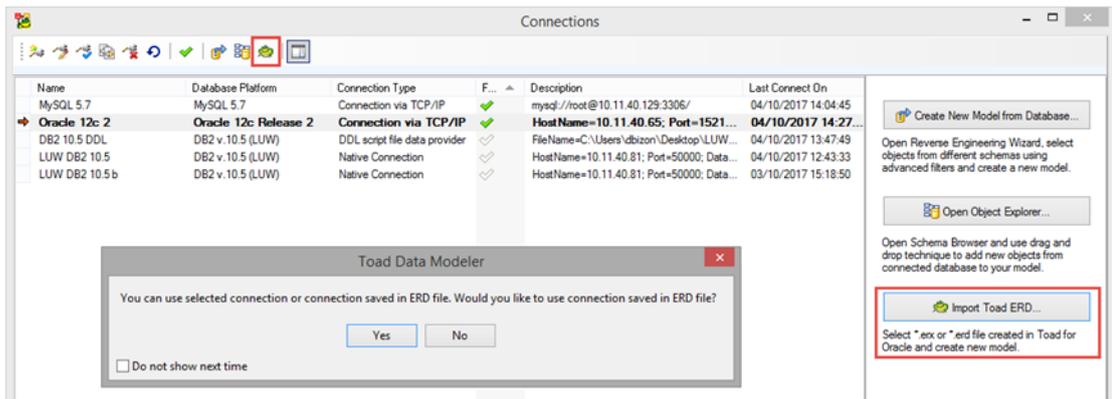


# Import Toad ERD

**i** **NOTE:** For more info on Toad for Oracle integration see [Toad for Oracle® Connections](#) and other related help topics.

## To create a model by importing a diagram created in Toad for Oracle

1. Click **File | Reverse Engineering | Connections**, click **Import Toad ERD** and select a file to import. You can use a selected connection or the connection saved in the imported file. Click “No” to use your own saved connection. If you click “Yes” an alias for a connection will be temporarily saved. The alias is removed after having refreshed **Connections** or restarting Toad Data Modeler.



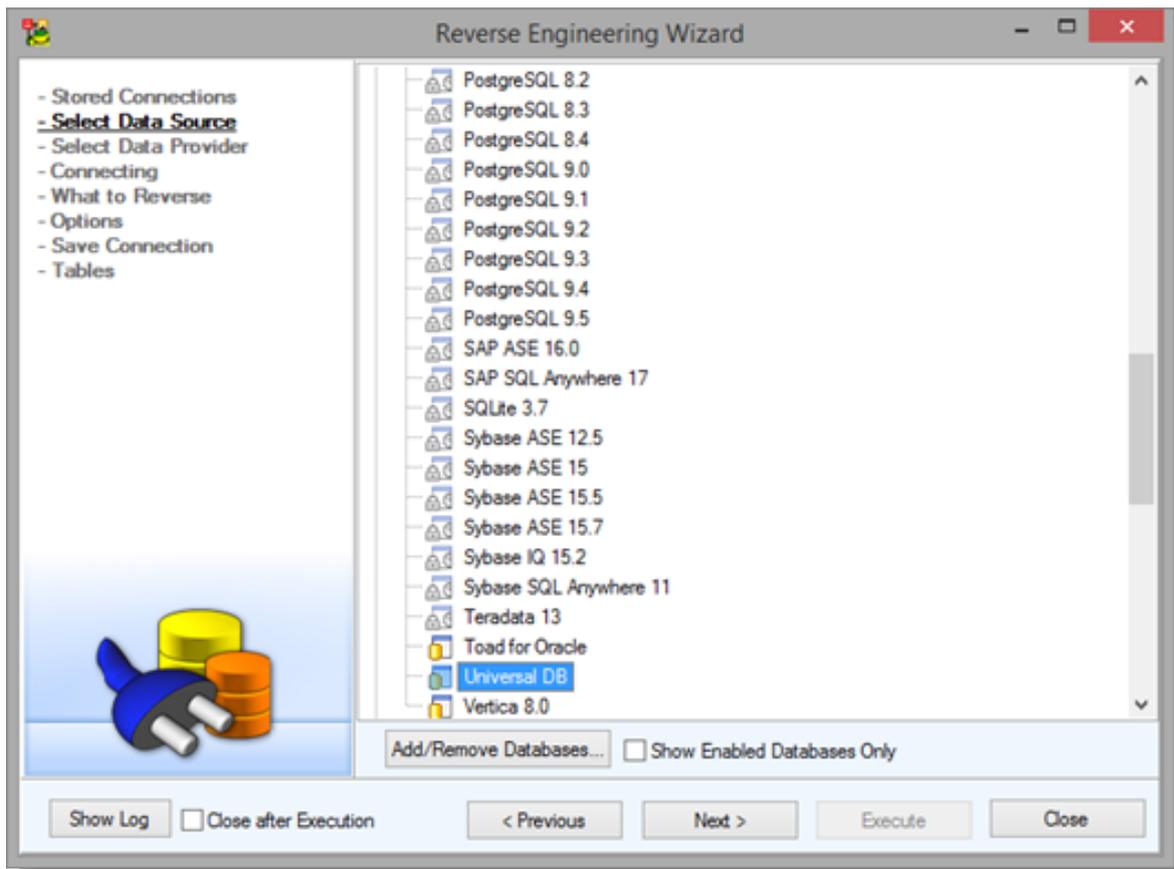
2. A new model is created from the imported Toad for Oracle file.

# Universal DB/ANSI Model

Toad Data Modeler allows you to reverse engineer a database structure of unsupported database platforms. Tables, columns, primary keys, indexes, relationships, procedures and views are loaded (depends on the database). It is necessary to convert the reversed model to a model of any supported databases.

## To reverse an ANSI model

- Follow the steps defined in [Connections](#) and select Universal DB in **Select Data Source** step.



## About Scripting and Customization

Toad Data Modeler and its behavior can be easily customized in several ways to better fit your needs and requirements. You can also leverage the power of scripting and automation to improve your workflow.

To start scripting/customizing, you need to enable **Expert Mode** first.

### **To enable Expert Mode**

Check **Expert Mode** in **Settings | Options | General**

**Expert Mode** menu and **Options** section **Expert Mode** will appear.

Right-click a model name in **Model Explorer** or **Application View** and select **Test Model** to run an internal test of model consistency.

Right-click a model name in **Model Explorer** or **Application View** and select **Repair Model** to repair broken models.

The scripting and customization capabilities include:

- **Scripts and macros**
- **Packages**

- **Metamodel**
- **Form customization**

**Note:** Useful information about available classes, methods, functions and properties accessible in Toad Data Modeler can be found in the **Reference Guide**.  
This document is available in the **Expert Mode | Reference Guide** (with **Expert Mode** enabled).

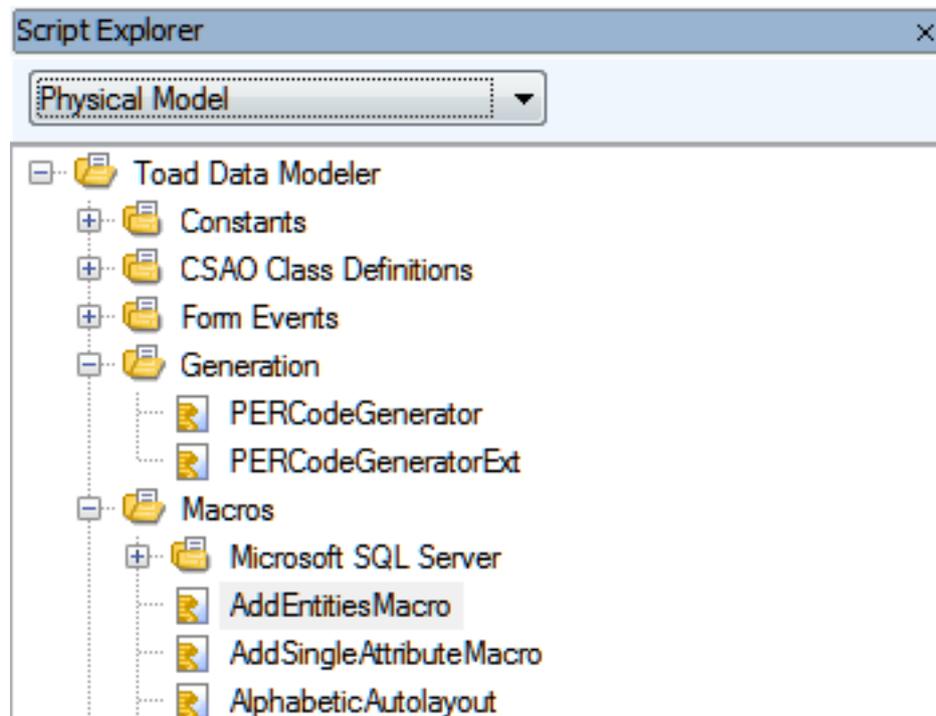
## Scripts/Macros

Code written in JScript, VBScript or Internal script may be used to access and modify various model objects or even perform model actions such as model renaming, merging, DDL script customization, etc.

**Note:** Macros are similar to scripts except they can be configured to appear in custom Macro menu or context menu of selected objects.

Scripts/Macros can be accessed via **Expert Mode Menu | Customization | Script Explorer**.

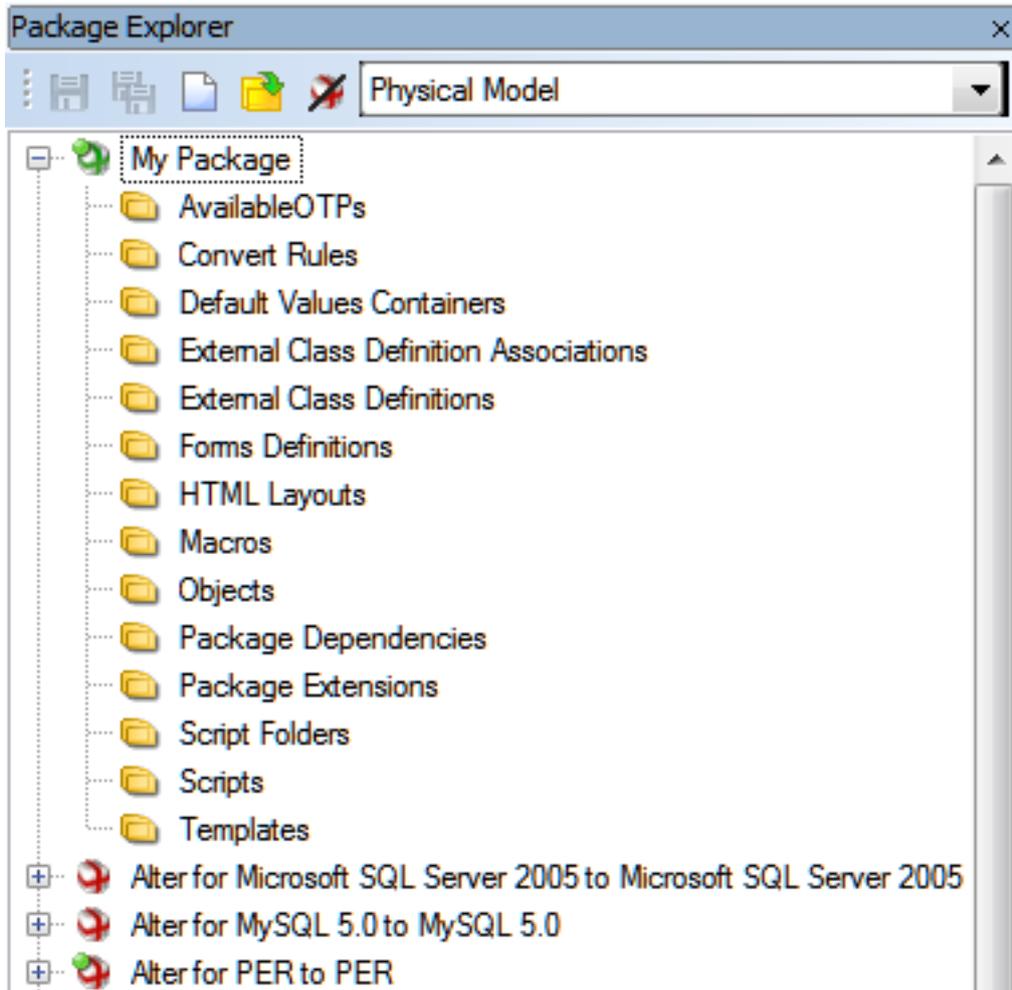
To create and immediately execute scripts, take advantage of **Scripting Window** located in **Expert Mode Menu**.



## Packages

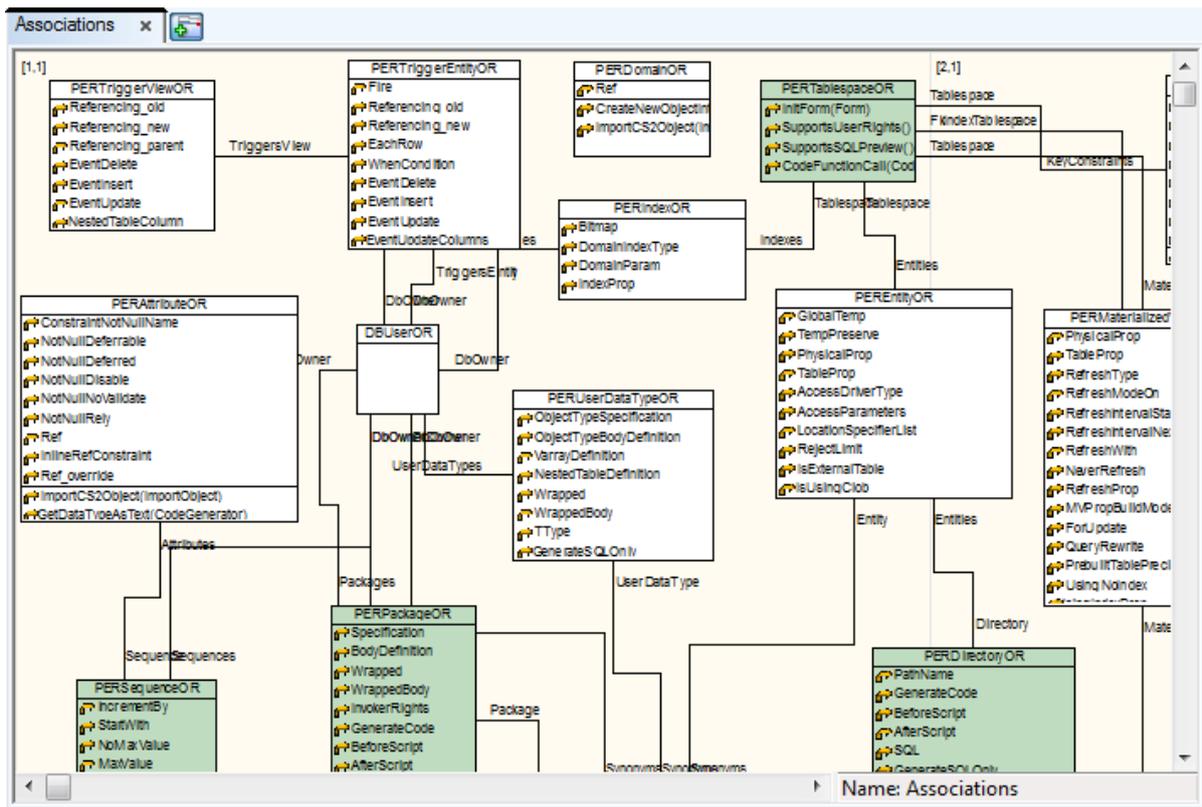
A package contains various means of customization such as scripts, macros, customized form definitions, data type conversion rules, etc. Packages affect models of various database platforms and versions depending on their Visibility and relations with other packages (Extension, Dependency).

Packages can be accessed via **Expert Mode Menu | Customization | Package Explorer**.



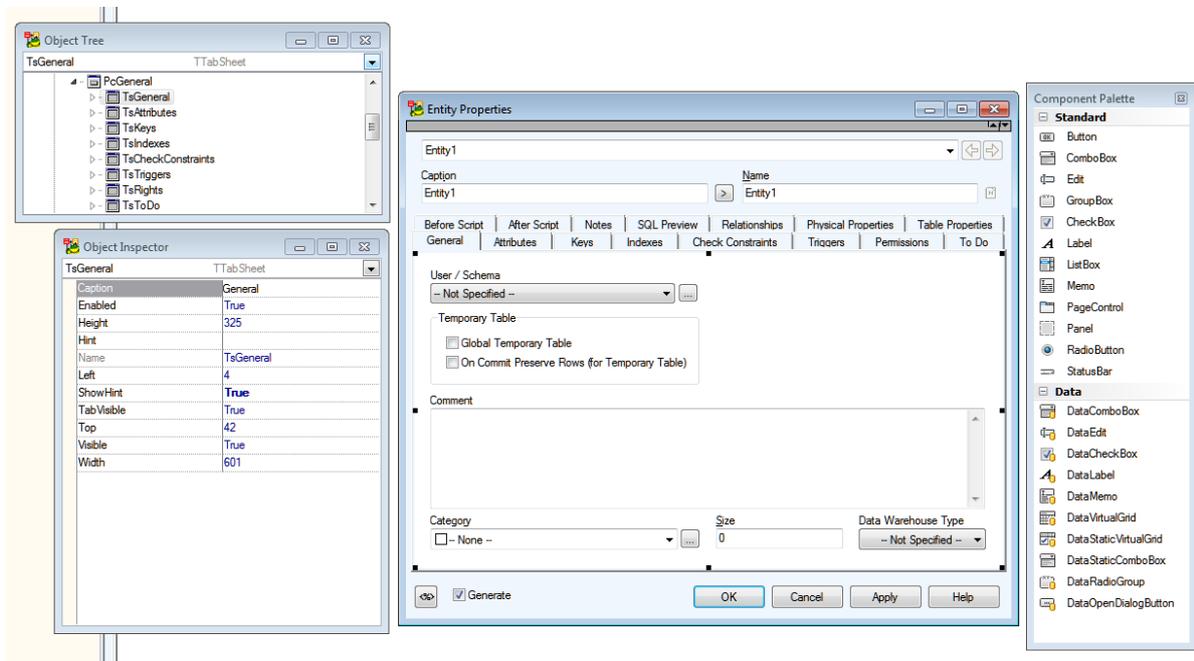
## Metamodel

Metamodel is a graphical representation of objects (classes, properties, methods, etc.) that belong to a selected package. In metamodels, you can create new classes, properties, methods, associations, generalizations, and others in a way similar to UML Class Diagram.



## Form Customization

It is also possible to modify certain application forms (mostly **Properties** dialogs) and either change existing elements or add custom ones.



## Other Notes

### Toad Data Modeler Order of Priority

1. Scripts and definitions stored in **My Package.txg** file have the highest priority.
2. Scripts and definitions stored in other user defined packages (distributed as add-ons, add-on packages) have higher priority than scripts and definitions stored in system packages.
3. Scripts and definitions stored in system packages have the lowest priority.

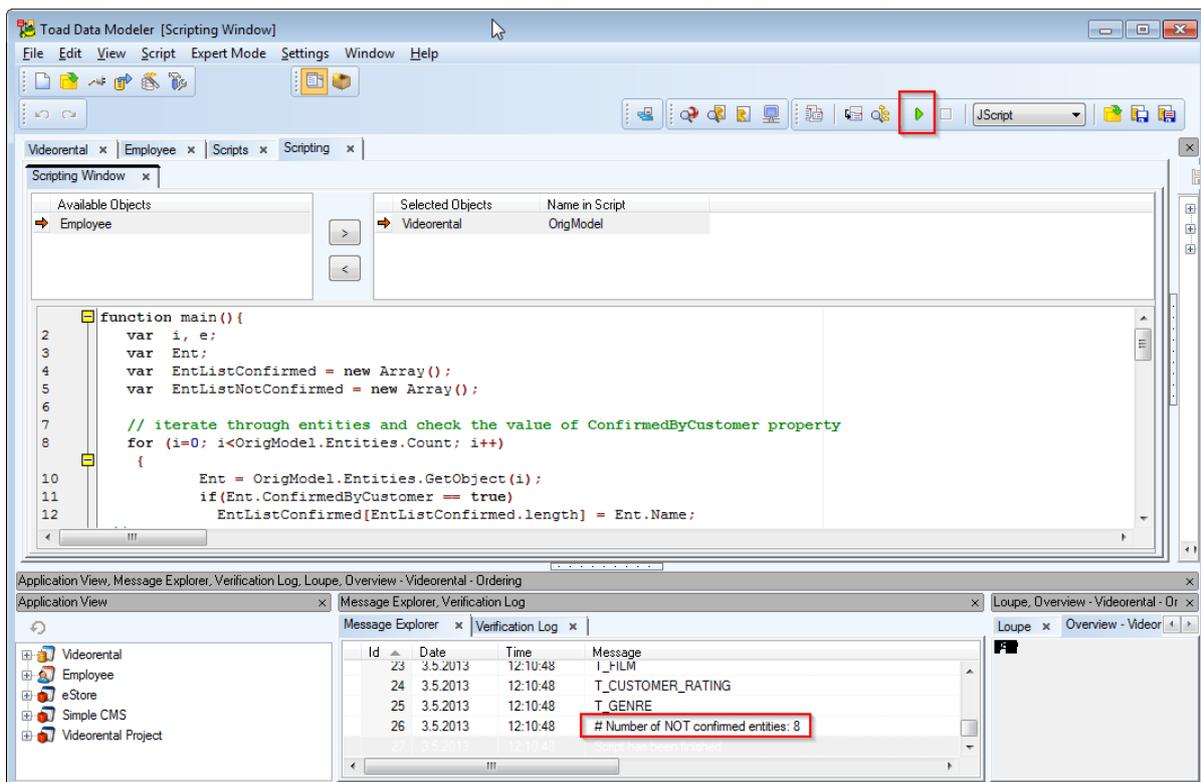
## Scripting Window

Scripting Window allows you to run simple scripts only. In **Scripting Window**, you cannot save scripts as they do not relate to any package.

### To open Scripting Window

1. Enable Expert mode: select **Settings | Options | General** | select the **Expert Mode** checkbox.
2. Select **Expert Mode | Scripting Window**.

In the Application Window, see the new tab **Scripting** with the **Scripting Window** open.



## Scripting Window toolbar



Option	Description
Show Windows Automatically	Displays a corresponding side tab. When you are writing a script, Code Explorer is displayed. When a script is being executed, Log is displayed.
Show Log	Displays a log window that shows log messages and errors related to <b>Scripting Window</b> .
Show Code Explorer	Displays a side tab that lists code segments.
Execute Script	Executes a script in <b>Scripting Window</b> .
Stop Script	Stops a running script.
Type	Switch between: <ul style="list-style-type: none"><li>• JScript</li><li>• VBScript</li><li>• Internal Script</li></ul>
Load Script from File	Load a script from a file.
Save Script	Saves a script.
Save Script as	Saves a script under a new name.

**To display the Available Objects and the Selected Objects windows in Scripting Window**  
Select **View** | **Show Registered Objects**.

## Script Explorer

In **Script Explorer**, you can edit existing scripts, write your own scripts/macros and more.

### To open Script Explorer

1. Enable Expert mode: select **Settings Menu** | **Options** | **General** | check the **Expert Mode** checkbox.
2. Click  on the toolbar or select **Expert Mode Menu** | **Customization** | **Script Explorer**.

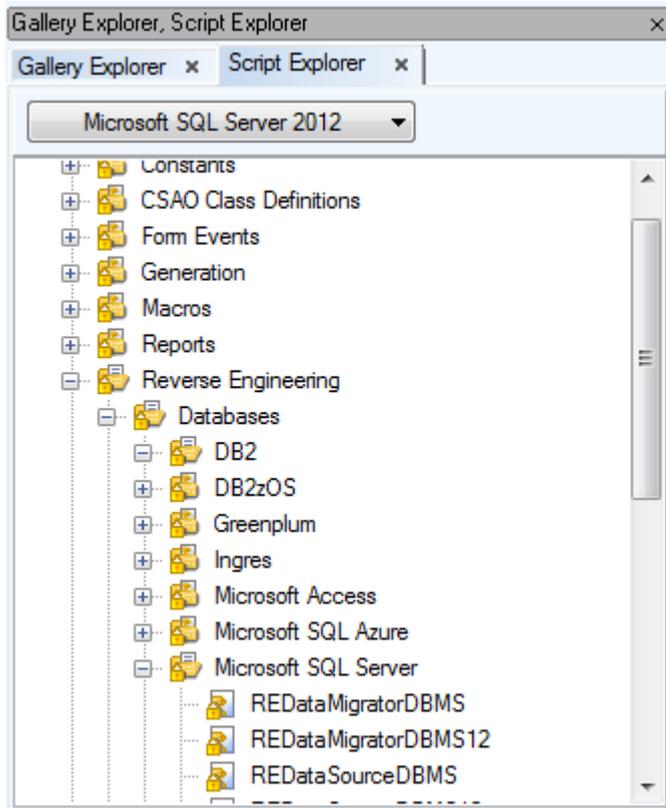
**Script Explorer** does not contain all data that you can find in **Package Explorer**, but only scripts stored in folders. Via the **Script Explorer**, you can:

- Make fine modifications in user scripts - see option **Edit Source Code (in New Window)**
- View source code of read-only scripts (via the **Edit Source Code** option as well), and copy their parts to use them in new scripts

- Extend functionality of existing scripts and modify them significantly - see option **Add New Script**
- Delete user scripts

See [Scripting in Script Editor](#) for more information.

## The Script Explorer Tree



From the combo-box at the top, you can select a script for:

- All Models
- Physical Model
- Specific database
- Metamodel
- Logical Model

**Example:** See the screenshot above - **Microsoft SQL Server 2012** has been selected. The scripts with visibility enabled in Microsoft SQL Server 2012 are displayed.

# Options for Folders

## Folders (Script Categories):

- **Constants** - scripts with constants
- **CSAO Class Definitions** - scripts with class definitions, mostly organized to folders with specific database names
- **Dictionaries** - contains dictionary scripts
- **Form Events** - contains event scripts
- **Reports** - contains scripts for HTML and RTF report generation
- **Reverse engineering** - contains scripts for creating a model via reverse engineering
- **Verification** - contains scripts for verification

**Right-click a folder to see the following options:**

Option	Description
Add New Script	Adds a new script under the selected folder.
Delete Folder	Deletes the selected folder, including its contents. Folders which contain system scripts cannot be deleted - they are locked (see the padlock icon).
Add New Folder	Creates a new folder in selected folder.
Properties	Opens the <b>Properties</b> dialog of the selected folder. From here, you can also manage items of the folder - add, delete and edit them, provided the folder is not read-only.
<b>General Tab</b>	This tab contains some basic information on the folder.
<b>Items Tab</b>	All subfolders and scripts of a particular folder are listed on this tab. Here, you can also manage them - add, edit, delete. (Provided that they are not read-only.)

# Options for Scripts

**Right-click a script to see the following options:**

Option	Description
Create New Script	Creates a new script under the selected script.
Delete Script	Deletes the selected script. If this script contains other script items, they will be deleted as well. System scripts cannot be deleted. They are locked (see the padlock

Option	Description
	icon). It's not possible to delete other read-only scripts either.
Edit Source Code	<p>Opens <b>Script Editor</b> in the Application Window.</p> <p>This option is available for all scripts - user, system and read-only scripts. Note that there is a significant difference though:</p> <ul style="list-style-type: none"> <li>• <i>User scripts that are not locked</i>: You can edit the source code entirely.</li> <li>• <i>System scripts and read-only scripts</i>: You can only view the script and copy its parts. You cannot edit source code of such scripts directly.</li> </ul>
Edit Source Code in New Window	<p>Same as above, however the source code of the selected script will open in a new instance of <b>Script Editor</b>.</p> <p><b>i</b> Note: You can open source code of the same script multiple times, for example when viewing different parts of a script in two windows.</p> <p>If any of the <b>Edit</b> options is selected, the script will be locked automatically, and other modifications of this script won't be allowed. (In Script Explorer, such scripts are disabled.)</p>
Properties	Opens the <b>Script Properties</b> dialog. Properties of system and read-only scripts cannot be edited.

## Script Properties

General Tab	Description
Name	Physical Script name
Caption	Logical Script name
Category	Selects the script category (folder) in which it can be found.
Script Type	Select Script Type: JScript, VBScript and Internal script are available.
Package	Name of the package where the script is stored. User scripts are stored in " <i>My Package</i> " by default.
Script Folder	Name of the folder in Script Explorer in which the script was created. Cannot be changed.
<b>Visibility Tab</b>	<p>On this tab, you can find information about the script visibility. It defines when the script is visible when you filter scripts in Script Explorer.</p> <p>Let's say your script visibility is set to "<i>Physical Model</i>". When you filter scripts in <b>Script Explorer</b> by category, your script will be shown each time the filtered category falls under <i>Physical Model</i>. But if you filter script by the <i>Logical Model</i> category, your script won't be shown.</p>

General Tab	Description
Others Tab	Contains mostly identifying info such as Author, Company, Version etc.
Notes Tab	A tab for writing notes about the script.

## Script Editor

Script Editor is accessible from **Script Explorer** or **Package Explorer**.

### To open Script Editor

- Right-click a script and choose **Edit Source Code** or **Edit Source Code in New Window**.

or

- Simply double click a script/macro in **Script Explorer**

When you open an instance of Script Editor, the **Scripts** tab is displayed at the top of the Application Window. When this tab is focused, a new tab row appears. All opened scripts are shown as tabs there.

**i** Note: Read-only scripts (system scripts and locked scripts) can be viewed only. You can still copy their code and use it in your own scripts.

<b>Left section of Script Editor</b>	List of functions in currently opened script.
Save	Saves changes you made in the script. (Shortcut: <b>CTRL+S</b> )
Rollback	Discards all changes made since the last save.

**i** Note: Double-click a function in the left section of Script Editor to move to its source code in the editor itself.

**i** TIP:

- You can insert bookmarks in your code:
  1. Press **CTRL+SHIFT+number 0-9** to insert a bookmark on selected row.
  2. To move to a created bookmark, press **CTRL + bookmark number**.
- You can also quickly search through all of existing scripts by going to **Expert Mode Menu | Customization | Find in Scripts** or pressing **CTRL+ALT+F**.

## Script Editor Right-Click Options

Copy	Ctrl+C
Cut	Ctrl+X
Paste	Shift+Ins
<hr/>	
Select All	Ctrl+A
<hr/>	
Undo	Ctrl+Z
Redo	Shift+Ctrl+Z
<hr/>	
Find...	Ctrl+F
Search Next	F3
Search All	
Go To Line...	Ctrl+G
<hr/>	
Convert Internal Script	
Script Properties	
Convert Internal Script 2	

These are standard functions that can be used while working with scripts in **Script Editor**.

Convert Internal Script	Option for internal scripts that allows you to see your internal script in JScript.
-------------------------	---

Convert Internal Script 2	Option for internal scripts that allows you to see your internal script in JScript. Moreover, via this option, you can see numbers of lines that you can map in case of errors in the script.
---------------------------	---

**i** Note: If an error occurs and it is an error in script, the script will open in **Script Editor** and the particular problematic line in the script will be highlighted. (Expert mode must be enabled.)

## How to Call Toad Data Modeler from Other Applications

Toad Data Modeler can be called from other applications, for example by running JavaScript code. This section showcases several possible scripts that may be used to perform various Toad Data Modeler tasks.

**i** Note: By default, scripts from outside Toad Data Modeler will use the latest installed version of the application. If you want to use some other version, you will need to locate its executable file and execute the following command in its folder:

**TDM.exe /regserver**

### Available tasks

By calling Toad Data Modeler from other applications, you can perform the following tasks silently:

- DDL script generation
- Report and change report generation

- Reverse engineering
- Model conversion

### DDL script generation - Generate.js

```
function Generate(Model, Output)
{
    var Generator = Model.DefaultCodeGenerator;
    Generator.GenerateToFile(Output);
}
/*****
var App = new ActiveXObject("TDM.App");

//Wait until all TDM packages are loaded
while(!App.Application.IsPackagesLoaded)
{
    WScript.Sleep(1000);
}

//Model for which DDL script will be generated
var Model = App.OpenModelFromFile("C:\\My\\ Models\\Videorental.txp");
//DDL script destination folder and name
Generate(Model, 'C:\\Scripts\\VideorentalGeneratedScript.sql');
// Model is deleted only from memory
Model.Delete();
```

### HTML report generation - HtmlReport.js

```
function HTMLReport(Model, System, OutputPath)
{
    var ReportRegistrar = System.CreateObject('ReportRegistrar');
    ReportRegistrar.DataSource = Model;
    var Report = ReportRegistrar.CreateReport
('BasicHTMLPERReport'+Model.ModelDef.Abbrev, 1 , Model); //1 - HTML report
    ReportRegistrar.RegisterLayoutClasses(1);
    Report.Path = OutputPath;
    Report.FileName = 'Report'; //Name of the HTML report file
    Report.Language = 'ENU'; //Abbreviation of language of dictionary used to translate
terms in report (default is english - ENU)
    Report.Kind = 'HTML';
    Report.Layout = ReportRegistrar.GetLayoutClass(0); //Report Layout (0 - Frameless,
1 - Top Menu, 2 - Left Menu)
    Report.CSS = Report.Layout.CSSList.GetObject(0); //CSS style (Frameless 0-10, Top
Menu 0-2, Left Menu 0-1)
    Report.GenerateInfo = false; //If true, adds information about model to the report
    Report.Generate();
}
/*****
var App = new ActiveXObject("TDM.App");

//Wait until all TDM packages are loaded
while(!App.Application.IsPackagesLoaded)
```

```

{
    WScript.Sleep(1000);
}

//Model for which the report will be generated
var Model = App.OpenModelFromFile("C:\\Models\\Videorental.txp");
//Report destination folder
HTMLReport(Model, App.System, 'C:\\Reports\\' );

```

### Change script generation - GenerateChangeScript.js

```

function ChangeFiles(App, FileName1, FileName2, Output)
{
    var Model1 = App.OpenModelFromFile(FileName1);
    var Model2 = App.OpenModelFromFile(FileName2);
    var Convertor = Model1.CreateNewObjectInternal(25000);

    Convertor.Model2ToAlter = true;
    Convertor.Converting = true; // Do not modify
    Convertor.Altering = false; // Do not modify
    Convertor.Model1Model = Model1;
    Convertor.Model2Model = Model2;
    Convertor.CreateDefaultAlterScriptSetting();
    Convertor.InitSelectedOTPs();
    Convertor.InitAvailableOTPs();
    Convertor.SelectPhysicalPropertiesOnlyInSelectedOTPs();
    Convertor.AlterFileName = Output;
    var Stream = App.System.CreateObject('TextStream');
    Convertor.SynchronizeModels();
    Convertor.LoadAllDifferences();
    Convertor.SelectAllModel1ToModel2(true, false);
    Convertor.RunAlter();
    Stream.Text = Convertor.AlterScript;
    Stream.FileName = Convertor.AlterFileName;
    Stream.SaveToFile();

    Convertor.Delete();
    Model1.Delete();
    Model2.Delete();
}
//*****
var App = new ActiveXObject("TDM.App");

//Wait until all TDM packages are loaded
while(!App.Application.IsPackagesLoaded)
{
    WScript.Sleep(1000);
}
//First and second model to be compared, change script destination folder and name
ChangeFiles(App, "C:\\Models\\Videorental.txp", "C:\\Models\\Videorental2.txp",
"C:\\Scripts\\VideorentalChangeScript.sql" );

```

## Change report generation - GenerateChangeReport.js

```
function ChangeReport(App, FileName1, FileName2, OutputPath)
{
    var System = App.System;
    var Model1 = App.OpenModelFromFile(FileName1);
    var Model2 = App.OpenModelFromFile(FileName2);
    var Convertor = Model1.CreateNewObjectInternal(25000);

    Convertor.Model2ToAlter = false;
    Convertor.Converting = true; //Do not modify
    Convertor.Altering = false; //Do not modify
    Convertor.Model1Model = Model1;
    Convertor.Model2Model = Model2;
    Convertor.CreateDefaultAlterScriptSetting();
    Convertor.InitSelectedOTPs();
    Convertor.InitAvailableOTPs();

    Convertor.SynchronizeModels();
    Convertor.LoadAllDifferences();
    Convertor.SelectAllModel1ToModel2(true, false);

    var ReportRegistrar = System.CreateObject('ReportRegistrar');
    ReportRegistrar.DataSource = Convertor;
    var Report = ReportRegistrar.CreateReport('UniversalHTMLAlterReport', 6,
    Convertor); // 6 - diff HTML
    ReportRegistrar.RegisterLayoutClasses(6);
    Report.Path = OutputPath;
    Report.FileName = 'Report'; //Name of the HTML report file
    Report.Language = 'ENU'; //Abbreviation of language of dictionary used to translate
    terms in report (default is english - ENU)
    Report.Kind = 'HTML';
    Report.Layout = ReportRegistrar.GetLayoutClass(0); //Report Layout (0 - Frameless,
    1 - Top Menu, 2 - Left Menu)
    Report.CSS = Report.Layout.CSSList.GetObject(0); //CSS style (Frameless 0-10, Top
    Menu 0-2, Left Menu 0-1)
    Report.Generate();

    Convertor.Delete();
    Model1.Delete();
    Model2.Delete();
}
//*****
var App = new ActiveXObject("TDM.App");

//Wait until all TDM packages are loaded
while(!App.Application.IsPackagesLoaded)
{
    WScript.Sleep(1000);
}
```

```
ChangeReport
(App, "C:\\Models\\Videorental.txp", "C:\\Models\\Videorental2.txp", "C:\\Reports\\")
```

### Converting model to PostgreSQL - ConvertToPg.js

```
function Convert(Model, App)
{
    var Convertor = Model.CreateNewObjectInternal(25000);
    Convertor.Model2ToAlter = false;
    Convertor.Model2ToConvert = true;
    Convertor.Model2ModelTemp = true;

    Convertor.Converting = true; //Do not modify
    Convertor.Altering = false; //Do not modify
    Convertor.Model1Model = Model;

    var PM = App.System.GetInterface('PackageManager');
    var DestinationMDef = PM.ModelDefs.GetObjectByName('PostgreSQL 9.0'); //Target
database platform and version
    DestinationMDef.LoadPackages();

    Convertor.Model2ModelDef = DestinationMDef;
    Convertor.Model2Model = App.NewModel(2001, DestinationMDef, true, true); //2001 =
Physical ER Model (PERModel)

    Convertor.CreateDefaultAlterScriptSetting();
    Convertor.InitSelectedOTPs();
    Convertor.InitAvailableOTPs();

    Convertor.SynchronizeModels();
    Convertor.LoadAllDifferences();
    Convertor.SelectAllModel1ToModel2(true, false);
    Convertor.RunConvert();

    var Result = Convertor.Model2Model;
    Convertor.Delete();
    return Result;
}

function Generate(Model, Output)
{
    var Generator = Model.DefaultCodeGenerator;
    // Sample generator settings changes. For more available settings, see class
PERCodeGenerator in Reference Guide
    Generator.UseQuotations = false;
    Generator.GenerateToFile(Output);
}
//*****
var App = new ActiveXObject("TDM.App");

//Wait until all TDM packages are loaded
while(!App.Application.IsPackagesLoaded)
```

```

{
    WScript.Sleep(1000);
}

//Source Model Path
var Model = App.OpenModelFromFile("C:\\Models\\Videorental.txp");
var ModelPG = Convert(Model, App);
// Save model
App.SaveModelToFile(Model, 'C:\\Models\\VideorentalPG.txp');
// Generate SQL
Generate(ModelPG, 'C:\\Models\\script.sql');
// Delete only from memory.
ModelPG.Delete();
Model.Delete();

```

### Reverse engineering using existing Alias - REExistingAlias.js

```

function LoadPackagesByModelDefName (AName, System)
{
    var PM = System.GetInterface('PackageManager');
    var MDef = PM.ModelDefs.GetObjectByName (AName);
    MDef.LoadPackages ();
}
/*****
function REByAlias (Alias)
{
    var REDataMigrator = Alias.REDataMigrator;
    REDataMigrator.InitiateREDataMigrator ();
    REDataMigrator.InitObjects ();
    var List = Alias.REStruct.Tables;
    var i;

    for (i=0;i<List.Count;i++)
    {
        List.GetObject(i).Selected = true;
    }

    Alias.REDataMigrator.REDataProvider.OpenConnection();//NEW
    Alias.REDataMigrator.LoadObjects ();
    Alias.REDataMigrator.REDataProvider.CloseConnection();//NEW
    if (Alias.REStruct.Model!=null)
    {
        return Alias.REStruct.Model;
    }
    else
    {
        return null;
    }
}
/*****
function RE(AliasName, App, AModelDefName)
{

```

```

var REManager = App.REManager;
//Load Packages
LoadPackagesByModelDefName (AModelDefName, App.System);
var Alias;
Alias = REManager.REAliases.GetObjectByName (AliasName);
if (Alias==null)
{
    var AliasPath = App.ApplicationConfig.AliasesPath;
    Alias = REManager.LoadREALiasFromFile (AliasPath+AliasName+'.txa');
}
return REByAlias (Alias);
}
//*****
var App = new ActiveXObject ("TDM.App");

//Wait until the application loads all packages
while (!App.Application.IsPackagesLoaded)
{
    WScript.Sleep (1000);
}

var Model = RE ('PG9 unidac', App, 'PostgreSQL 9.0');
if (App.Application.SaveModelToFile (Model, 'C:\\test.txp'))
{
    WScript.Echo ("OK");
}
else
{
    WScript.Echo ("Error");
}

```

### Reverse engineering and new alias - RENewAlias.js

```

function REByAlias (Alias)
{
    var REDataMigrator = Alias.REDataMigrator;
    REDataMigrator.InitiateREDataMigrator ();
    REDataMigrator.InitObjects ();

    var List = Alias.REStruct.Tables;
    var i;
    //Select some objects to RE, only example
    for (i=0;i<List.Count;i++)
    {
        List.GetObject (i).Selected = true;
    }

    Alias.REDataMigrator.LoadObjects ();
    if (Alias.REStruct.Model!=null)
    {
        return Alias.REStruct.Model;
    }
}

```

```

}
/*****
function LoadPackagesByModelDefName (AName, System)
{
    var PM = System.GetInterface('PackageManager');
    var MDef = PM.ModelDefs.GetObjectByName (AName);
    MDef.LoadPackages ();
}
/*****
function SetAliasParameters (REALias)
{
    var REDataProvider = REALias.REDataProvider;
    REDataProvider.SetConnectionParam('HostName', 'localhost');
    REDataProvider.SetConnectionParam('DatabaseName', 'Videorental');
    REDataProvider.SetConnectionParamAsInt('Port', 0);
    REDataProvider.SetConnectionParam('UserName', 'Scott');
    REDataProvider.Password = 'Lion';
    return REALias;
}

/*****
*****
var App = new ActiveXObject("TDM.App");

//Wait until the application loads all packages
while (!App.Application.IsPackagesLoaded)
{
    WScript.Sleep(1000);
}

LoadPackagesByModelDefName('PostgreSQL 9.0', App.System);
var REManager = App.System.GetInterface('REManager');
var REALias = REManager.CreateAlias('REDataSourceDBPG90',
'REDataProviderUniDACClientPG');
REALias = SetAliasParameters(REALias);

var Model = REByAlias(REALias);
REALias.Delete();

if (App.SaveModelToFile(Model, 'C:\\Models\\Videorental.txp'))
{
    WScript.Echo("OK");
}
else
{
    WScript.Echo("Error");
}

```

## Package Explorer

**Package Explorer** displays package structure in Toad Data Modeler and allows you to:

- Manage objects saved in the packages (rename, move, copy objects, open **Object Properties** dialog etc.)
- Access all package Metamodels

### To open Package Explorer

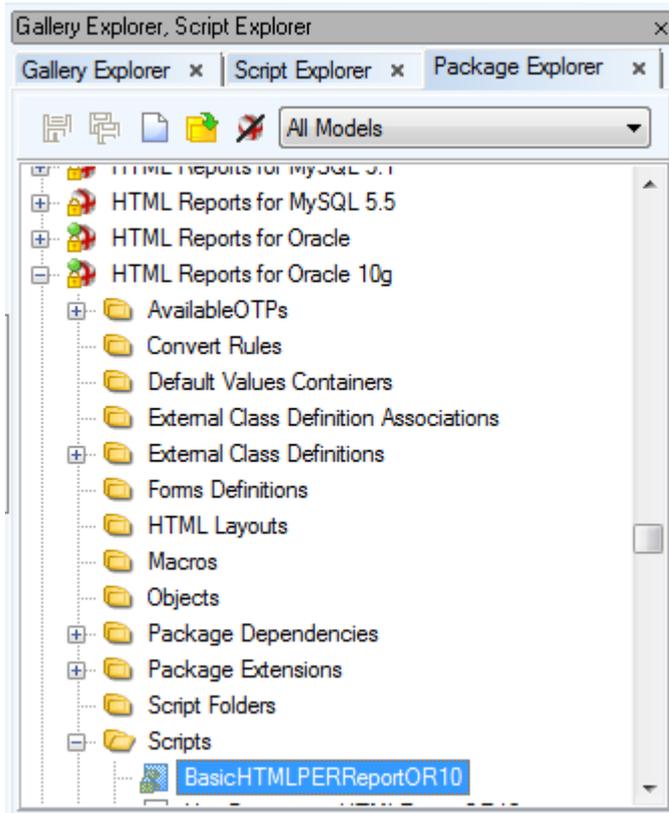
1. Enable Expert mode: select **Settings Menu | Options | General** | select the **Expert Mode** checkbox.
2. Click  on the toolbar or select **Expert Mode Menu | Customization | Package Explorer**.

Toad Data Modeler contains packages for:

- Specific supported databases (e.g. Oracle, Microsoft SQL Server 2019, MySQL 8.0 etc.)
- Physical ER models
- Logical models
- HTML and RTF reports
- and other functions, such as SQL/DDL generation, reverse engineering etc.

## The Package Explorer Tree

Via the box at the top, you can filter the list of packages.



Read-only packages/scripts cannot be edited.

Root folders contain packages that have various lists of objects, e.g.:

- **Forms Definitions** - contain partial definitions with a list of changed visual components of forms modified by user
- **External Class Definitions**- contain classes and their members designed by user (see the "Metamodel" topic)
- **Package Dependencies** - show dependencies between the selected package and other packages

## Right-click Package options

Option	Description
Load Package	Loads the package from disk to Toad Data Modeler. For faster work with packages and for faster loading of application, some information is loaded only on demand.
Delete Package	Deletes the selected Package and all items it contains.
Disallow Package	Selected package won't be loaded during next Toad Data Modeler launch.
Export Package	Exports the selected Package in <i>.tbg</i> format to the chosen folder
Open Metamodel	Opens the selected Package Metamodel. See <a href="#">Metamodels</a> for more information.
Extend with New Package	Creates a new package that will extend the currently selected package.
Create Dependent Package	Creates a new package that will be dependent on the currently selected package (will inherit from it).
Properties	Contains information about the selected package and its items (General information, Visibility, Script folders, Scripts, Description). From here, you can also manage the scripts in the package - add, delete or edit them. (See the following example.)  <b>Package File</b> box in the <b>Package Properties</b> dialog - Path to the location where the package is stored. Click the button on the right to open the location in file explorer.
Save BIN	The selected XML package (*.txg) will be saved in binary format (*.tbg).
Save All as BIN	All XML packages (*.txg) will be saved in binary format (*.tbg).

## Package Explorer Toolbar

Icon	Command
	<b>Save Actual</b> - saves changes to actually active Package

Icon	Command
	<b>Save All</b> - saves all changes made in all modified Packages
	Creates a new user Package.
	Adds an existing Package to the Package Explorer.
	Hides all system Packages, leaving only the user and add-on ones visible.

## Icons of Packages in Package Explorer

Package	Unlocked				Locked			
	Loaded	Loaded-Modified	Unloaded	Disallowed	Loaded	Unloaded	Disallowed	Error during Loading
System								
Add-On								
My Package				-			-	

State	Description
Locked	<p>A package is locked when:</p> <ul style="list-style-type: none"> <li>a) it is marked as read-only on the disk</li> <li>b) it is a system package and user does not have <b>Expert Mode</b> enabled</li> </ul> <p><b>i</b> Note: All system packages are read-only by default. However, in Expert mode it is possible to change the lock/unlock property of the package (right-click the package in Package Explorer   <b>Properties</b>.) Generally, in Expert mode it is possible to lock/unlock system packages and add-on packages.</p>
Loaded	Complete package has been loaded to memory.
Loaded - Modified	Package has been modified by user.
Unloaded	Package has not been loaded to memory.
Disallowed	Package has been disallowed by user (right-click the package in

State	Description
	Package Explorer   <b>Disallow Package</b> ).
Error during Loading	<p>Loading of the package failed. It is an error state. This situation can happen e.g. when a dependent package of this package is missing (for example it was not selected during installation of the application).</p> <p><b>i</b> Note: It is not possible to edit a script when package is locked. Unlock a package to edit its scripts.</p>

## Scripting in Script Editor

In Toad Data Modeler, you can create and edit scripts via Script Explorer and Package Explorer. For scripting purposes, Script Explorer should be used. ([Script Explorer](#))

### **To create a new script**

Select a script/folder in **Script Explorer** | right-click and select **Create New Script**.

### **To edit already existing script**

Select a script/folder in **Script Explorer** | right-click and select **Edit Source Code(in New Window)**.

In both cases **Script Editor** opens.

When you modify a script source code or write a new script code:

- The particular script is locked and no other user can modify it.
- The script is disabled in Script and Package Explorers.
- Use **Commit** to save the changes and **Rollback** to cancel the changes in Script Editor. As soon as you press any of these buttons, the script becomes available for other users (will be unlocked automatically).

**i** Note: After you click **Commit**, the changes you've made for the script will be saved within the TDM application, however they will not be saved in particular package on your disk (My Package). You can either save the package in Package Explorer, or close the application - the changes will be saved in appropriate package and on disk automatically.

Scripting Languages in Toad Data Modeler:

- JScript
- Visual Basic
- Internal Script -[Internal Scripting](#)

### **To make a petty script modification**

(It doesn't work for read-only scripts.)

1. In **Script Explorer**, select a script that you want to modify.
2. Right-click the script and select **Edit Source Code**.
3. In the **Script Editor**, edit the source code directly.

4. Press **Commit** to confirm changes, or **Rollback** to cancel the changes.

5. Save the changes in appropriate package and on your disk - Click  in Package Explorer.

### **To modify functions of system scripts**

System scripts are read-only and it's not possible to edit their source code directly. (See disabled **Commit** and **Rollback** icons in **Script Editor**.)

If you want to edit any functions of a system script, you have to create a new script and edit appropriate functions in it. For detailed example, please see User Guide, "Customization - Sample" chapter.

### **To write a new script from scratch**

In Toad Data Modeler, you can create a new script or further extend functionality of already existing scripts - user and also system scripts.

1. Open **Script Explorer**.
2. Select a folder (category) where you want to add the new script.
3. Right-click the selected folder and select **Add New Script**. -> A new item will display in the **Script Explorer** under the selected folder.
4. Double-click the new script to open it in **Script Editor**.
5. Write or insert the scripting code to the right window. Use **Commit** or **Rollback** for saving and canceling the changes. Remember to save the changes in the package too.

## **Internal Scripting**

In Toad Data Modeler, the following scripts are supported:

- JScript
- Visual Basic
- Internal Script

Use the internal script for writing more extensive texts where only few commands are contained.

The internal script is similar to markup languages such as XML, HTML etc.

Every sign that is not a text must be marked by this sign - "#".

## **Key Words**

- import
- require
- if
- else
- endif
- script - Script function is generated. It allows users to define script type (e.g. JScript or VBScript) where the content between script and endscript commands is written.
- endscript

- `proc` - Creates a function in JScript. It is possible to define parameters here.
- `endproc` -
- `call` - Calls a procedure (function). It is possible to define parameters here.
- `<%` - Beginning of macro.
- `%>` - End of macro.
- `Forall` - Executes iteration over list.
- `@` - Shows that the following text is an expression.

## "script" Command

Use this command to insert to the internal script another part of a scripting language.

```
#script language="{Scripting language}"
{Code of Scripting Language}
#endscript
```

### Example:

```
#script language="Jscript"

function something() {
    Log.Information('My Message');
}

#endscript
```

## "proc" and "endproc" Command

Use this command to create a procedure that is available in internal language.

```
#proc
Name([parameters])
Code of Procedure
#endproc
```

### Example:

```
#proc
Greetings()
    Hello
    Buy
#endproc
```

## "Call" Command

Use this command to call procedures defined by command "proc".

### Example:

```
#call Greetings()
```

## "forall" Command

Use this command if you want to execute iteration over a list. The result of the iteration should be a text.

Command syntax is the following:

```
#forall LIST_NAME (PARAM1, PARAM2, PARAM3, PARAM4, PARAM5 );
```

**Example:**

```
#forall Model.tables('Create Table'+IterateItem.Name+'(',',',GenerateColumns(),',',',')');
```

## "<%", "%>" Command

Angled brackets with percentage insert part of internal script where JScript is used (or another scripting language). The expression is evaluated immediately.

**Example:**

```
<% if (Model.Count>0) Log.Information('\something'); %>
```

## "@" Command

It is similar to "<% %>", however the difference is that after the @ sign, only one expression follows.

# Creating Custom Properties

To further extend the functionality of Toad Data Modeler, you can create your own custom properties for most objects. To create a new custom property:

1. **Right-click** an object in **Model Explorer** and select **Add Custom Property**.

**New Custom Property**

EntityVersion  $\geq$  EntityVersion

Level  
PEREntity

Type  
widestring

ReadOnly  
 Script Implementation

Package  
My Package

OK Cancel

2. **New Custom Property** dialog appears. Here you can configure your property. Once, you're done, restart Toad Data Modeler and your property will be created and accessible via scripting.

Option	Description
Caption/Name	Logical and physical name of the property. <b>Name</b> is used to refer to the property in scripting.
Level	Determines on which level the property should be created. For example, if you create a new custom property for an entity (in Oracle 10g model), you can choose from the following options: <ul style="list-style-type: none"> <li>• <b>PEREntityOR10</b> - The property can be used only in Entities in Oracle 10g models.</li> <li>• <b>PEREntityOR</b> - The property can be used only in Entities in any Oracle models.</li> <li>• <b>PEREntity</b> - The property can be used in all Entities.</li> </ul>
Type	Sets the custom property data type. The data type should be chosen depending on what would you like to store in the property: <ul style="list-style-type: none"> <li>• <b>Widestring</b> - Text strings in general, supports Unicode characters.</li> <li>• <b>String</b> - Text strings in general, does not support Unicode characters.</li> </ul>

Option	Description
	<ul style="list-style-type: none"> <li>• <b>Integer</b> - Positive/negative whole numbers.</li> <li>• <b>Boolean</b> - True/False values.</li> <li>• <b>Real</b> - Floating points numbers.</li> <li>• <b>Dispatch</b> - Any Toad Data Modeler object (Entities, Attributes...)</li> </ul>
Readonly	Flags the property as <b>Read Only</b> , meaning its value cannot be changed.
Script Implementation	<p>This option allows you to customize the way your property will behave using scripting. You should check it in two cases:</p> <ul style="list-style-type: none"> <li>• <b>Your custom property value will be set according to other property values</b> - In this case, use the automatically generated <b>Get</b> method to get other properties values and set your custom property value according to them.</li> <li>• <b>Other properties values will be changed depending on your custom property value</b> - This can be done using the automatically generated <b>Set</b> method. From there, you can change the value of any other property.</li> </ul>
Package	Determines in which Package the custom property will be stored. <b>Loading/Disallowing</b> the selected Package will cause the property to be <b>usable/unavailable</b> .

## Custom Property Example 1

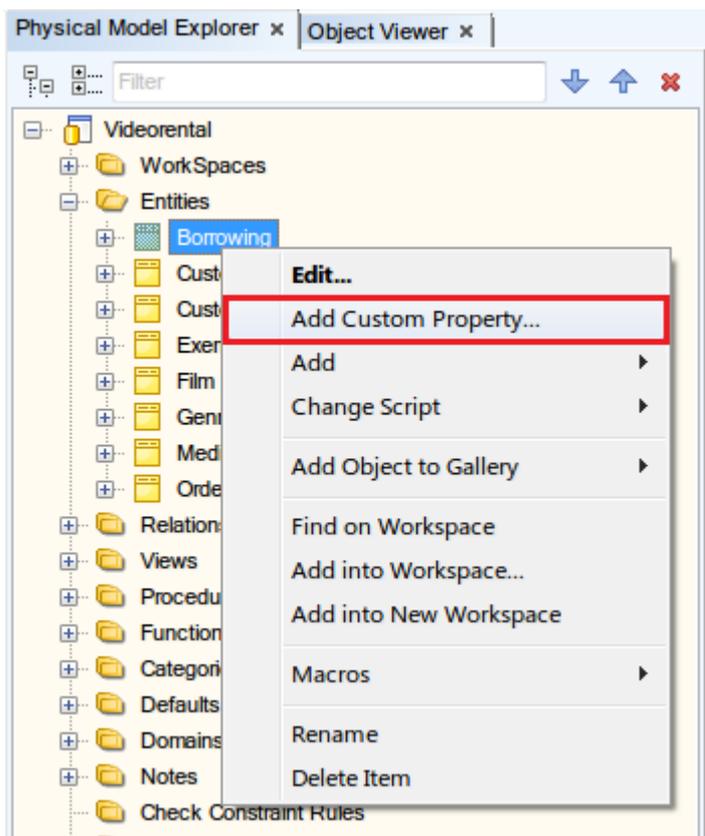
### Scenario

You would like to add an additional property to all entities which would indicate that modeling of a particular entity is not done yet and there are more things to be done before the entity is considered complete. The property could be named **InProgress**.

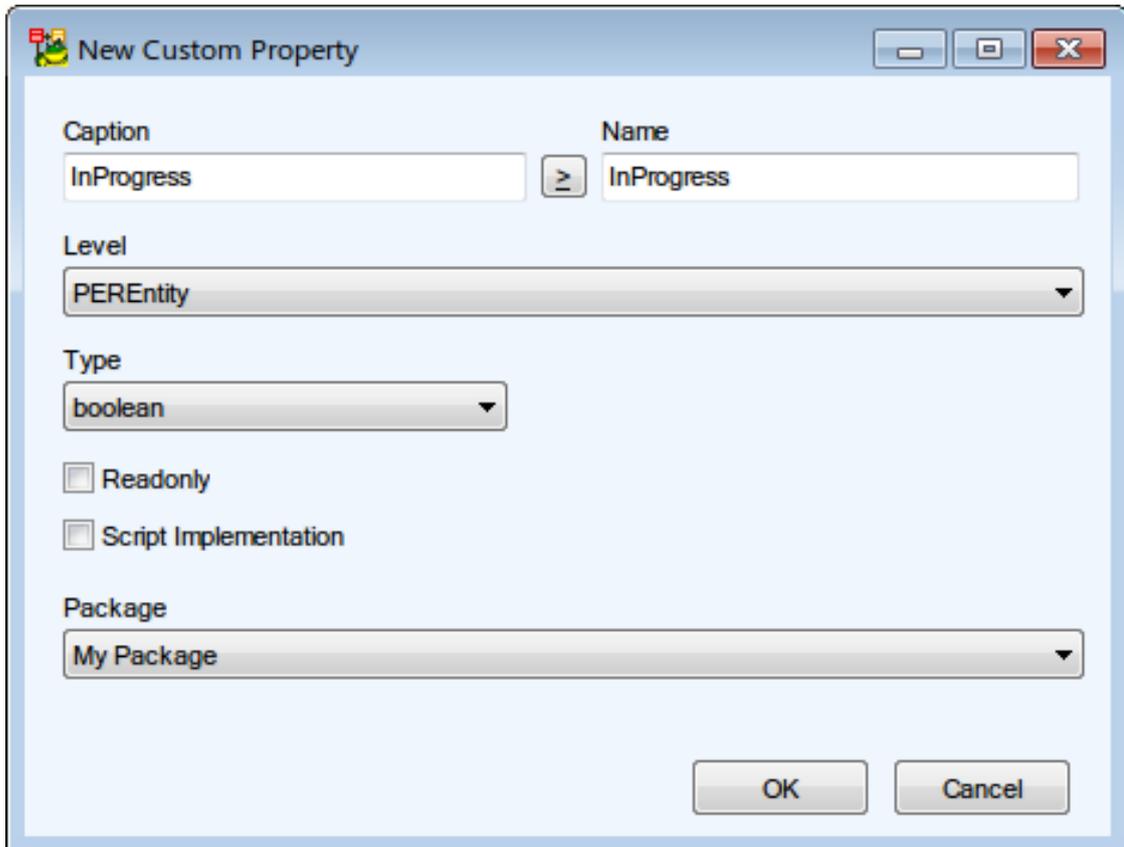
### Solution:

Create a simple custom property and add a checkbox to **Entity Properties** linked to the property. Use the following steps as a guide:

1. **Right-click** any entity in **Model Explorer** and select **Add Custom Property**.

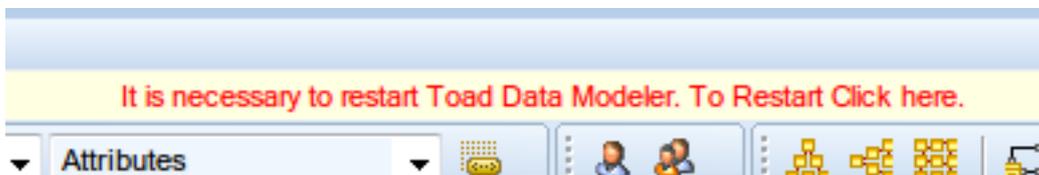


- In the **New Custom Property** dialog, enter the following:
  - Caption/Name** - enter **InProgress** as Name and anything you want as Caption
  - Level** - we assume the property should be available in all entities in all models, select **PEREntity**
  - Type** - the **boolean** type is the most suitable for our type of property (true/false)
  - Readonly** - the property value should be editable, do not check this checkbox
  - Script Implementation** - the property value is not dependent on other property values and it doesn't change any other property value either, do not check this checkbox
  - Package** - feel free to choose any Package

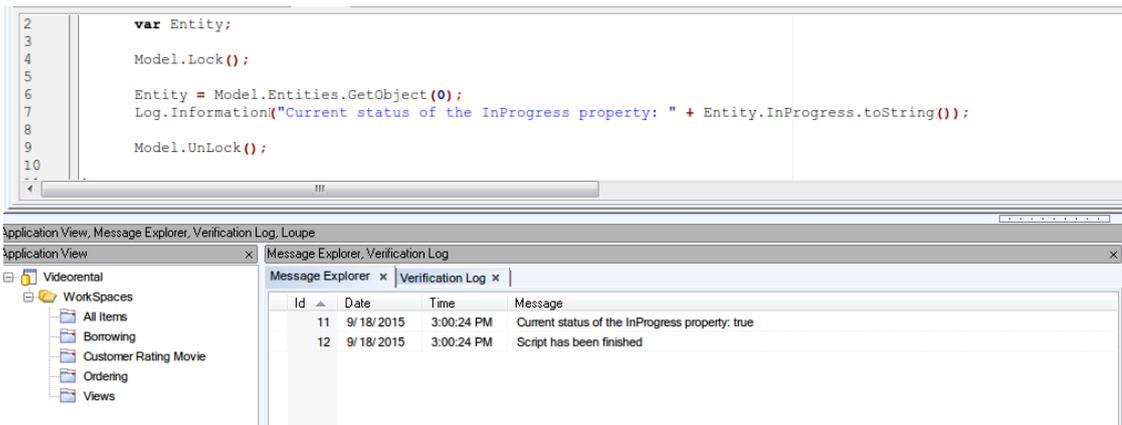


The screenshot shows the 'New Custom Property' dialog box. The 'Caption' field is set to 'InProgress' and the 'Name' field is also set to 'InProgress'. The 'Level' dropdown menu is set to 'PEREntity'. The 'Type' dropdown menu is set to 'boolean'. There are two unchecked checkboxes: 'Readonly' and 'Script Implementation'. The 'Package' dropdown menu is set to 'My Package'. At the bottom right, there are 'OK' and 'Cancel' buttons.

- Confirm the changes, you will need to restart the application in order to access the newly created property.

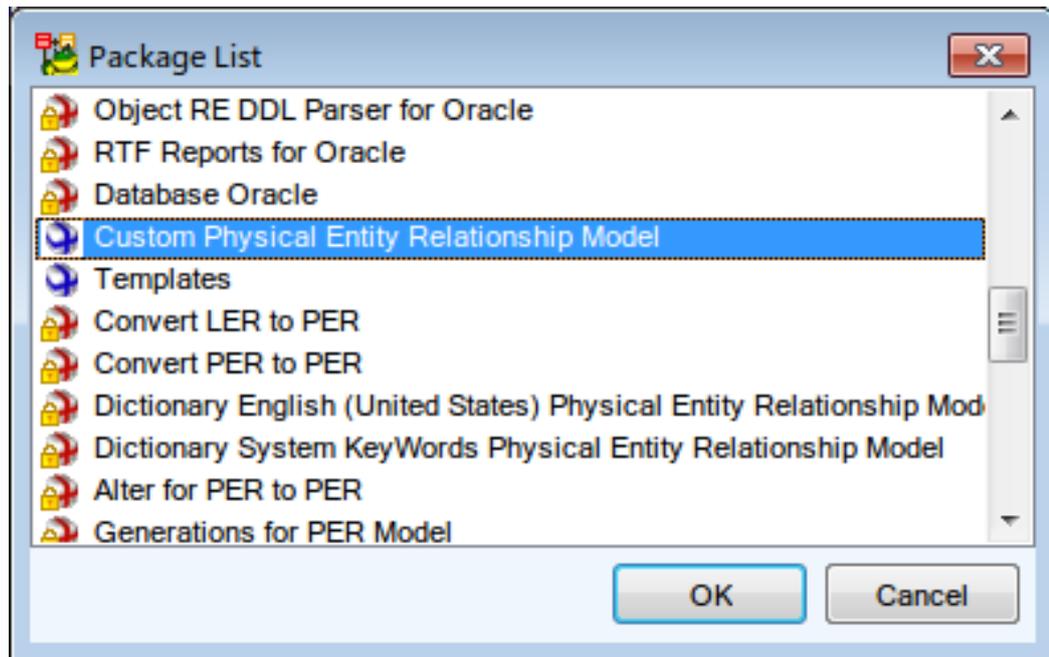


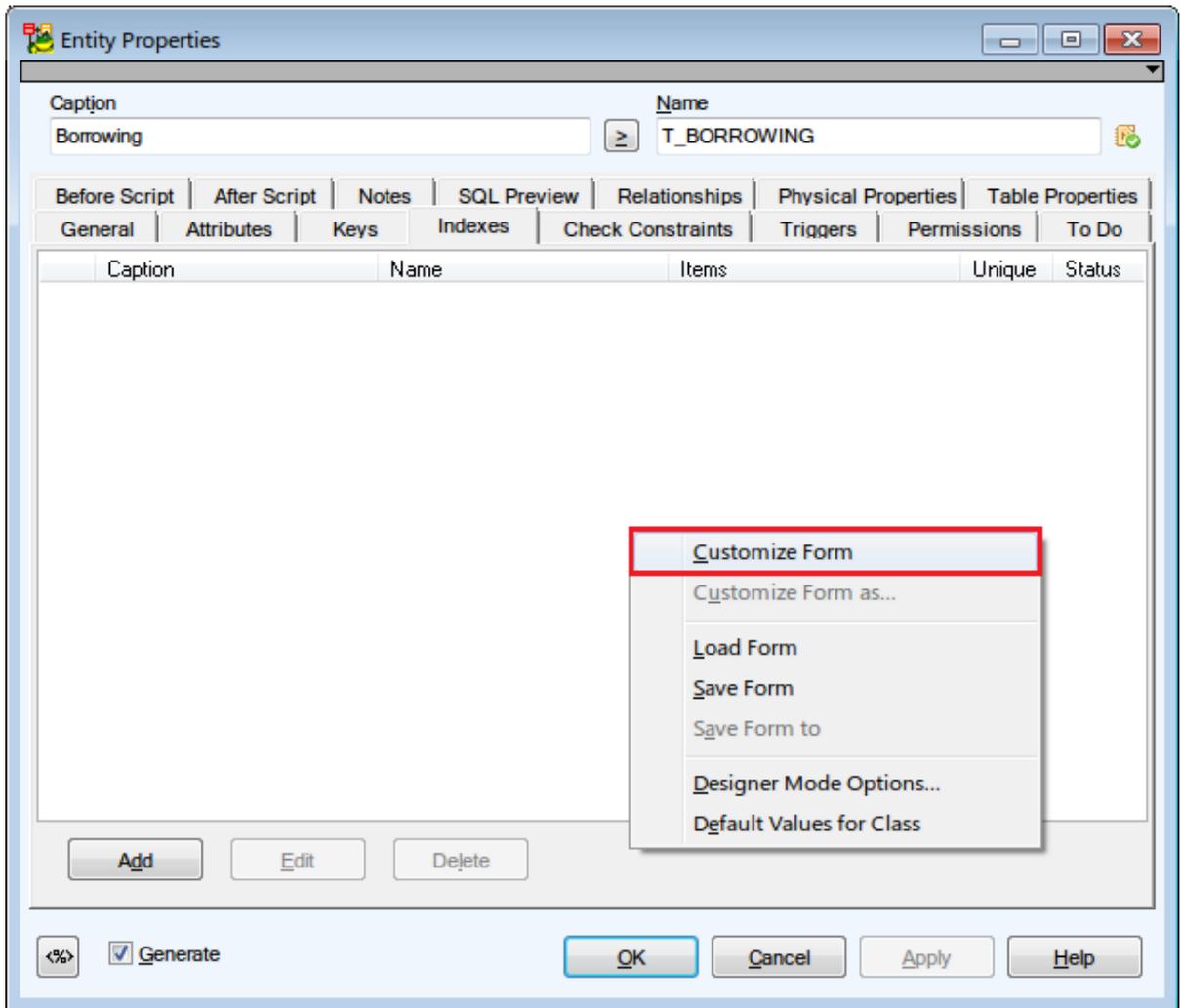
4. The custom property is now available for use. You can verify that by accessing it via scripting.



5. Now it's time to add a checkbox to the **Entity Properties** form. Right-click anywhere in the form and select **Customize Form**.

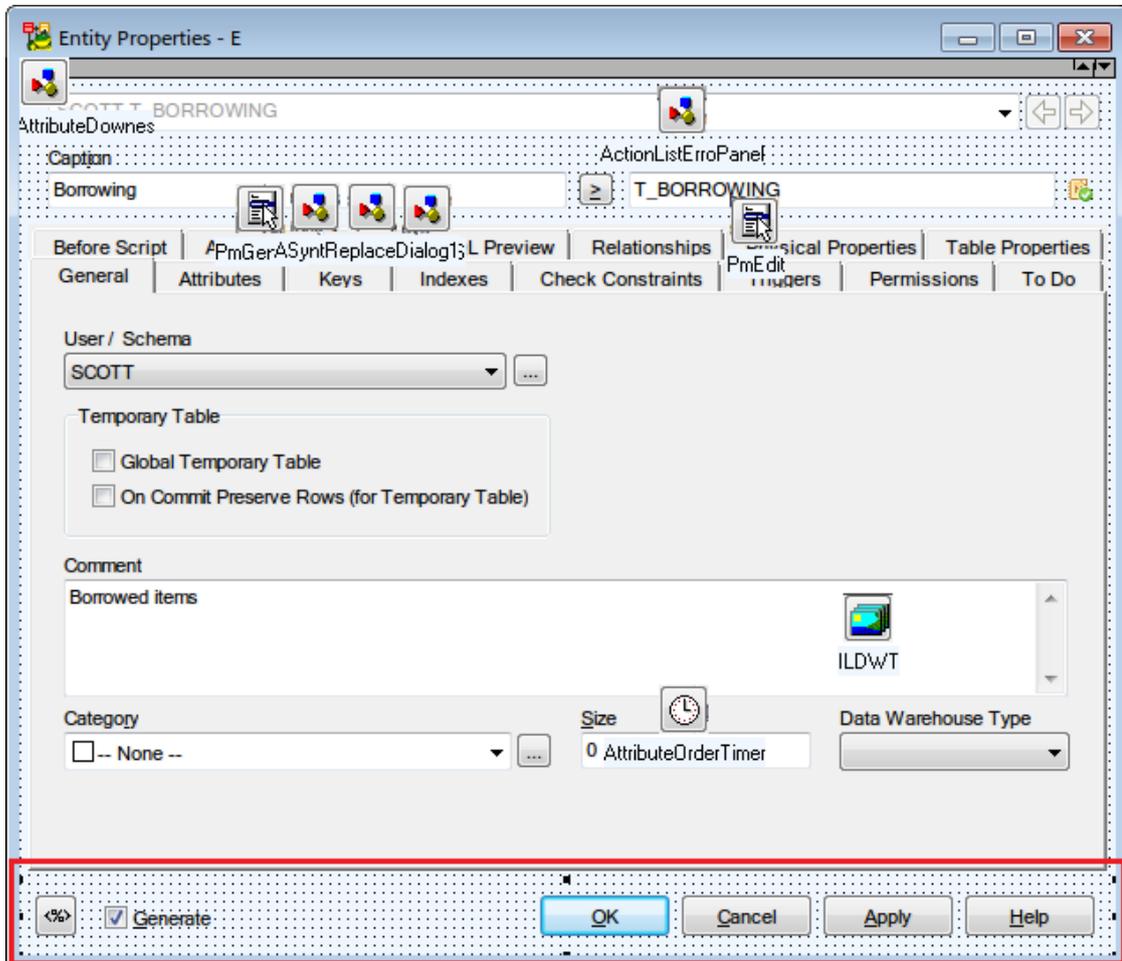
**i** Note: If you have chosen Package other than My Package, you need to select **Customize Form** as option and choose the Package you have selected previously (step 2). This option is disabled by default. To enable it, go to **Settings Menu | Options | Expert Mode** and uncheck the **Save the definitions the the 'My Package'** option.



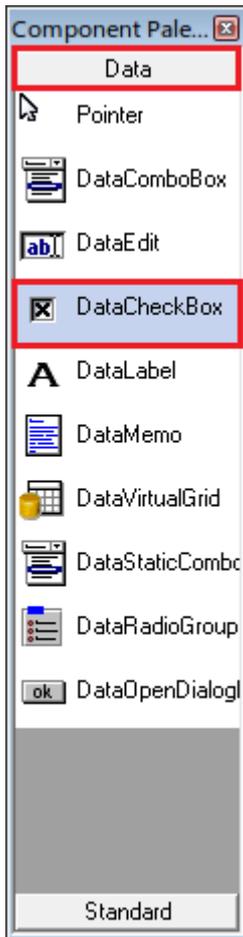


6. The Entity Properties form is now editable and several other forms appear. The checkbox for our property should be placed somewhere where we can see it no matter what tab is currently selected. Let's place it next to the **Generate** option. First, click the bottom section of the form, where the option is placed along with several buttons.

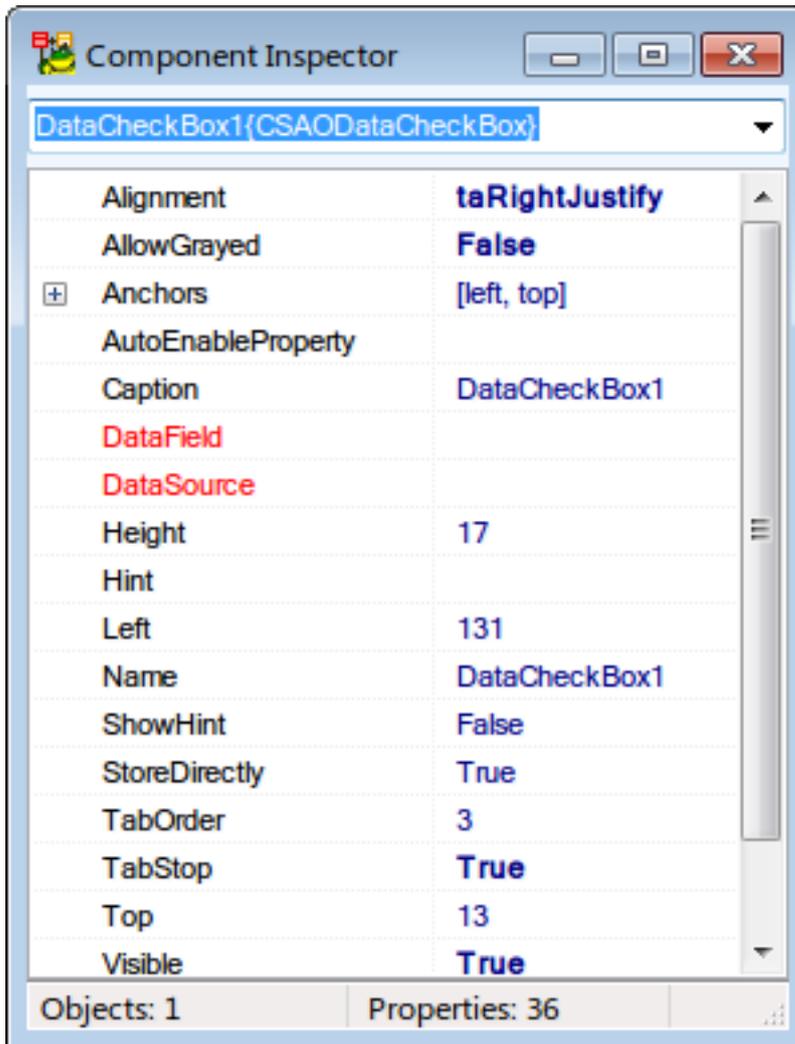
**i** Note: If you accidentally close some of the customization forms, go to **Settings Menu | Options | Expert Mode | Editable Forms** and check the appropriate option to show them again.



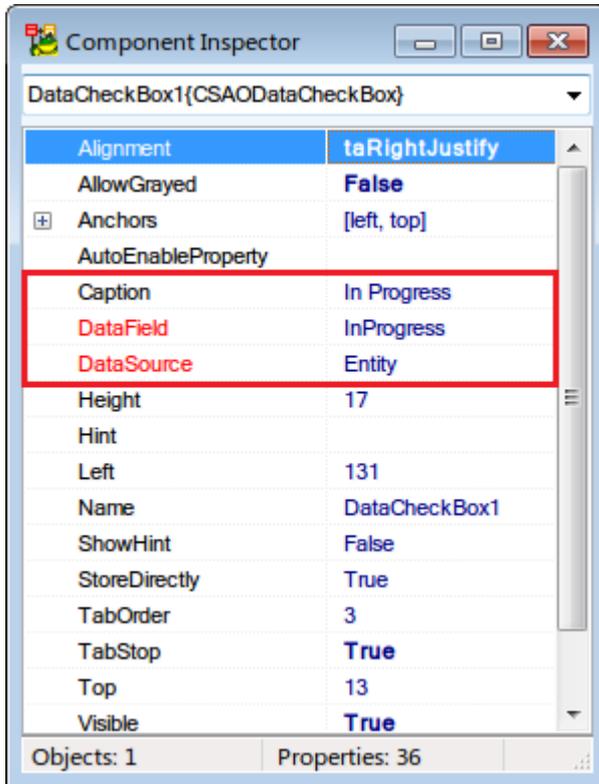
7. Now find the **Component Palette**, go to **Data** section and double-click **DataCheckBox** to add it to the form.



8. Move the newly added checkbox next to the **Generate** option (you can use **Alignment** form). Make sure the checkbox is selected, then locate the **Component Inspector**.

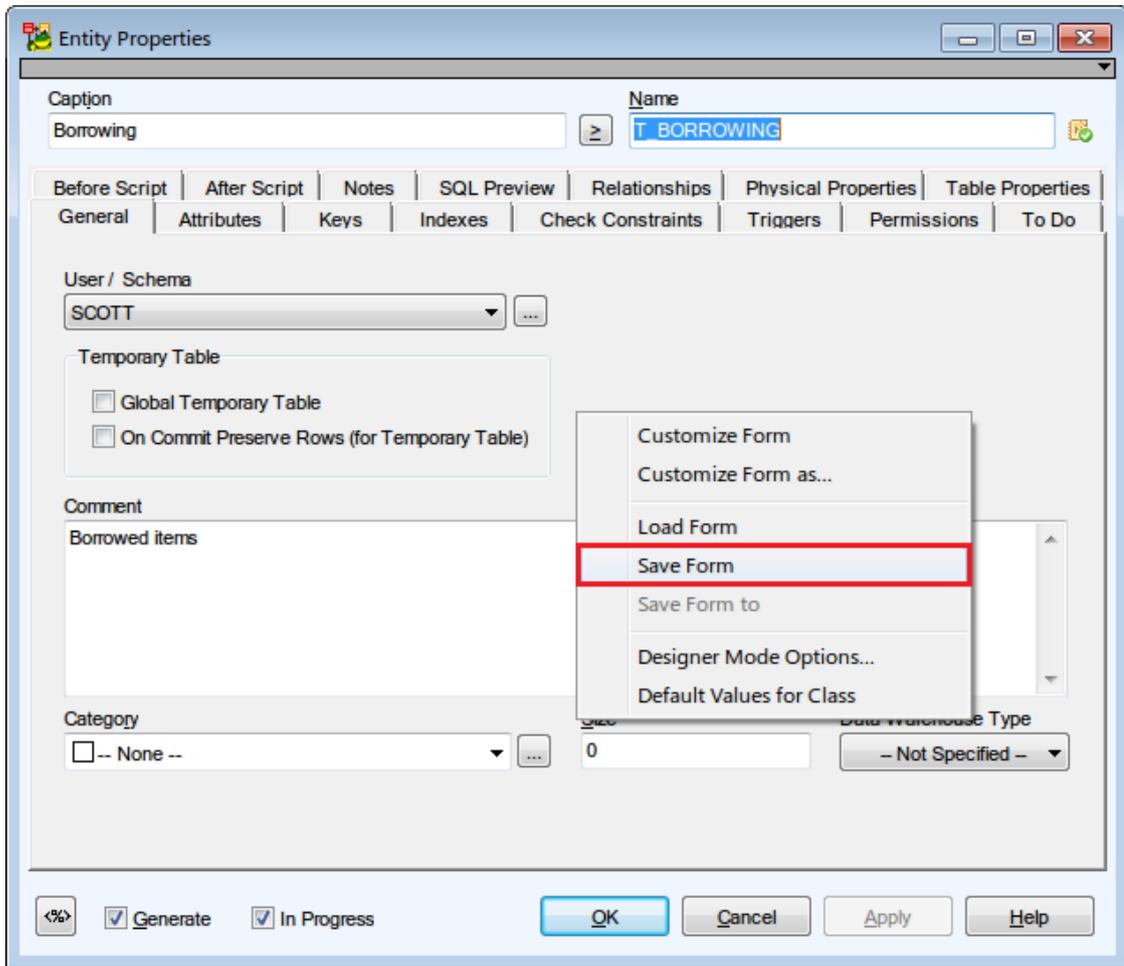


9. In **Component Inspector**, change the following properties:
- Caption** - enter a descriptive caption, e.g. **In Progress**
  - DataSource** - choose **Entity**
  - DataField** - find your custom property in the list and select it (**InProgress** in our case)



10. Close the editable **Entity Properties** via cross button in the **upper-right corner**. This will cause you to exit the editable mode of the form.

11. **Right-click** anywhere in the form and select **Save Form**. The changed form definition will be saved to **My Package**.



12. Your custom property is now available in all entities across all databases. The property and the modified form definition are stored in **My Package** (or in other package, depending on what you chose in step 2).

**i** TIP: If you want to transfer your customizations to another installation of Toad Data Modeler, simply move the package file from the **Packages** folder to the same folder of the target installation (e.g. "C:\Program Files (x86)\Quest Software\Toad Data Modeler 5.5\Packages\User\My Package.txg")

## Sample Scripts and Scripting Tips

In this topic, you can find sample scripts that can be executed via **Scripting Window** or used in user macros.

### Reorder attributes/columns

Option 1:

```
Attributes.InitSort();
```

```

//set ordinal values as you need

var temp = Attr1.Ordinal;

Attr1.Ordinal = Attr2.Ordinal;

Attr2.Ordinal = temp;

Attributes.Sort(); //it will get ordered according to the ordinal values
set for individual items

```

#### Option 2:

```
Attributes.Move(2,0) // Moves the third attribute to the place of the first attribute (parameters are
current_index,
The first parameter is the Current index in the list and the second is the index where an item
should move.
```

## Refresh of Model Explorer and Workspace

RefreshAllWorkspaces

- Refreshes all Workspaces of currently active model.

RefreshWorkspace

- Refreshes the selected Workspaces by ID.

RefreshModel

- Refreshes all Workspaces and also Model Explorer.

## Add a Relationship and Key

//Create Relation with Shortcut on WS workspace

```
var Rel = Model.AddLink(2004,ParentEntShape.ID, ChildEntShape.ID, WS.ID);
```

OR

//Create Relation without Shortcut

```
Model.AddLinkObject(2004, ParentEnt, ChildEnt);
```

## Add Columns to Keys

Key.AddAttribute(Attr)

- Adds an attribute to key.

Key.CommitChanges()

- Propagates the changes.

## Possibility to Set User Rights

//SelectedObject - Every object that can have permission (Entity, Function, Procedure etc.)

//User - User or User Group

```
Model.AddUserRight(SelectedObject, User, 'SELECT', 'Grant');
```

## Accessing First Entity in First Model

```
function main()
```

```

{
    var app = System.GetInterface('Application');
    var Model = app.Models.GetObject(0); // first model in Application View
    var Entity = Model.Entities.GetObject(0); // first entity in a model
    ...
    ...
}

```

## Using Lock and Unlock Methods

It is good to use the Lock and Unlock methods, otherwise you will not see changes on your Workspace directly. Use both methods to make safe modifications and refresh your WS automatically.

```

function main()
{
    ...
    Entity.Lock();
    Entity.Name = "new_name";
    Entity.Unlock();
    ...
}

```

## Accessing Log (Message Explorer)

Use the following to make Log accessible.

```

function main()
{
    ...
    Log = System.CreateObject('Log');
    ...
}

```

## Iterating Entities and Modifying Second Parameter of Attribute Data Type

This sample is for **Oracle** models only and changes the second parameter of Char(x) and Varchar2(x) data types to BYTE.

```

function main()
{

```

```

var app = System.GetInterface('Application');
var Model = app.Models.GetObject(0);
var e, a, Entity, Attribute;
Model.Lock();
for (e=0; e<Model.Entities.Count; e++) // iterate entities
{
    Entity = Model.Entities.GetObject(e);
    Entity.Lock();
    for (a=0; a<Entity.Attributes.Count; a++) // iterate attributes
    {
        Attribute = Entity.Attributes.GetObject(a);
        if ((Attribute.DataType.Caption == "Char(x)") || (Attribute.DataType.Caption
== "Varchar2(x)"))
        {
            Attribute.DataTypeParam2 = "BYTE";
        }
    }
    Entity.Unlock();
}
Model.Unlock();
}

```

## Creating New Entities

```

function main()
{
    var app = System.GetInterface('Application');
    var Model = app.Models.GetObject(0);
    Model.Lock();
    var Entity = Model.CreateNewObject( 2002 );
    Entity.Name = 'MyNewEntity';
    Model.Unlock(); // Refresh all windows associated with your model
}

```

Object type of PEREntity is 2002. More information can be found in the Reference document. Click **Help | Reference** to open the Reference.

## Converting Entity and Attribute Names to Lower Case

```
function main()
{
    var app = System.GetInterface('Application');
    var Model = app.Models.GetObject(0);

    for (e=0; e<Model.Entities.Count; e++)
    {
        Entity = Model.Entities.GetObject(e);
        Entity.Lock();
        Entity.Name = Entity.Name.toLowerCase();
        Entity.Unlock();
        Log.Information("Name of entity "+Entity.Name+" was changed.");
        for (a=0; a<Entity.Attributes.Count; a++)
        {
            Attribute = Entity.Attributes.GetObject(a);
            Attribute.Lock();
            Attribute.Name = Attribute.Name.toLowerCase();
            Attribute.Unlock();
            Log.Information("Name of attribute "+Attribute.Name+" in entity
"+Attribute.Owner.Name+" was changed.");
        }
    }
}
```

**Note:** The toLowerCase function is a JavaScript function.

## Adding a Prefix to Entity, Index and Trigger Names

```
function main()
{
    var prefix = "abc"; // defined prefix
```

```

var regular_expression_prefix = new RegExp(prefix+"_");
var app = System.GetInterface('Application');
var Model = app.Models.GetObject(0);

// Entities
for (e=0; e<Model.Entities.Count; e++)
{
    Entity = Model.Entities.GetObject(e);

    if (Entity.Name.search(regular_expression_prefix) == -1) // if prefix is not
used in name
    {
        Entity.Lock();

        Entity.Name = prefix+"_"+Entity.Name;

        Entity.Unlock();

        Log.Information("Name of entity "+Entity.Name+" was changed.");
    }

// Indexes
for (i=0; i<Entity.Indexes.Count; i++)
{
    Index = Entity.Indexes.GetObject(i);

    if (Index.Name.search(regular_expression_prefix) == -1) //if prefix is not
used in name
    {
        Index.Lock();

        Index.Name = prefix+"_"+Index.Name;

        Index.Unlock();

        Log.Information("Name of index "+Index.Name+" in entity "+Index.Owner.Name+"
was changed.");
    }
}
}

```

```

// Triggers
for (t=0; t<Entity.Triggers.Count; t++)
{
    Trigger = Entity.Triggers.GetObject(t);
    if (Trigger.Name.search(regular_expression_prefix) == -1) //if prefix is not
used in name
    {
        Trigger.Lock();
        Trigger.Name = prefix+"_"+Trigger.Name;
        Trigger.Unlock();
        Log.Information("Name of trigger "+Trigger.Name+" in entity
"+Index.Owner.Name+" was changed.");
    }
}
}
}

```

## Renaming NotNull Constraints

The script is for **Oracle** models only. It goes through all NotNull attributes and sets their notnull constraint name in format NN\_nameoftable\_number. For names exceeding 30 characters, it will truncate the NN\_nameoftable part.

```

function main()
{
    var app = System.GetInterface('Application');
    var Model = app.Models.GetObject(0);

    for (e=0; e<Model.Entities.Count; e++)
    {
        Entity = Model.Entities.GetObject(e);
        count = 0;
        for (a=0; a<Entity.Attributes.Count; a++)
        {
            Attribute = Entity.Attributes.GetObject(a);
            count++;
        }
    }
}

```

```

if (Attribute.NotNull)
{
    ConstraintNotNullName = "NN_" + Entity.Name;

    SumLength = ConstraintNotNullName.length + 1 + count.toString().length;

    if (SumLength > 30)

        ConstraintNotNullName = ConstraintNotNullName.substr(0,30-(count+1));

    ConstraintNotNullName = ConstraintNotNullName + "_" + count.toString();

    Attribute.ConstraintNotNullName = ConstraintNotNullName; // change name
of index

    Log.Information("NotNull Constraint Name of attribute "+Attribute.Name+" in
entity "+Attribute.Owner.Name+" was changed.");

}
}
}
}
}

```

## Selecting Override Identity Checkboxes at Once

This script is valid for **Microsoft SQL Server 2000** and **Microsoft SQL Server 2005** models only.

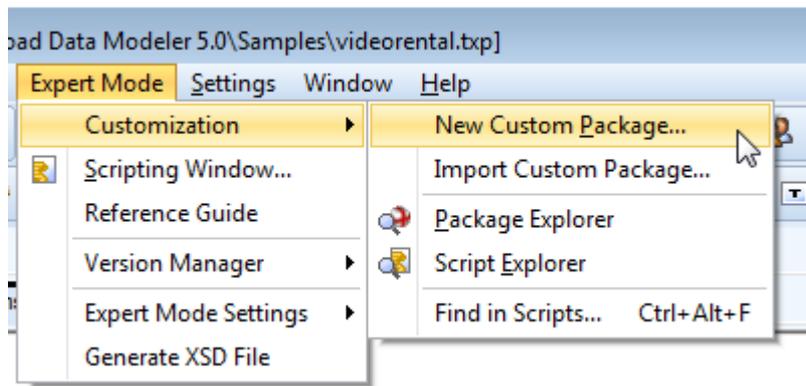
You can use the script provided that you set autoincrement in a domain, use the domain in PK attribute and create relationship to another entity. In this case, FK attribute with the domain (and identity) will be created in child entity. However, you need to override the identity. Not to do it for each FK attribute individually, you can run this script that goes through all FK attributes and automatically selects the checkbox **Override Identity**. After you execute the script, take a look at the Log window where all entities and attributes where the checkbox was selected is written out.

```

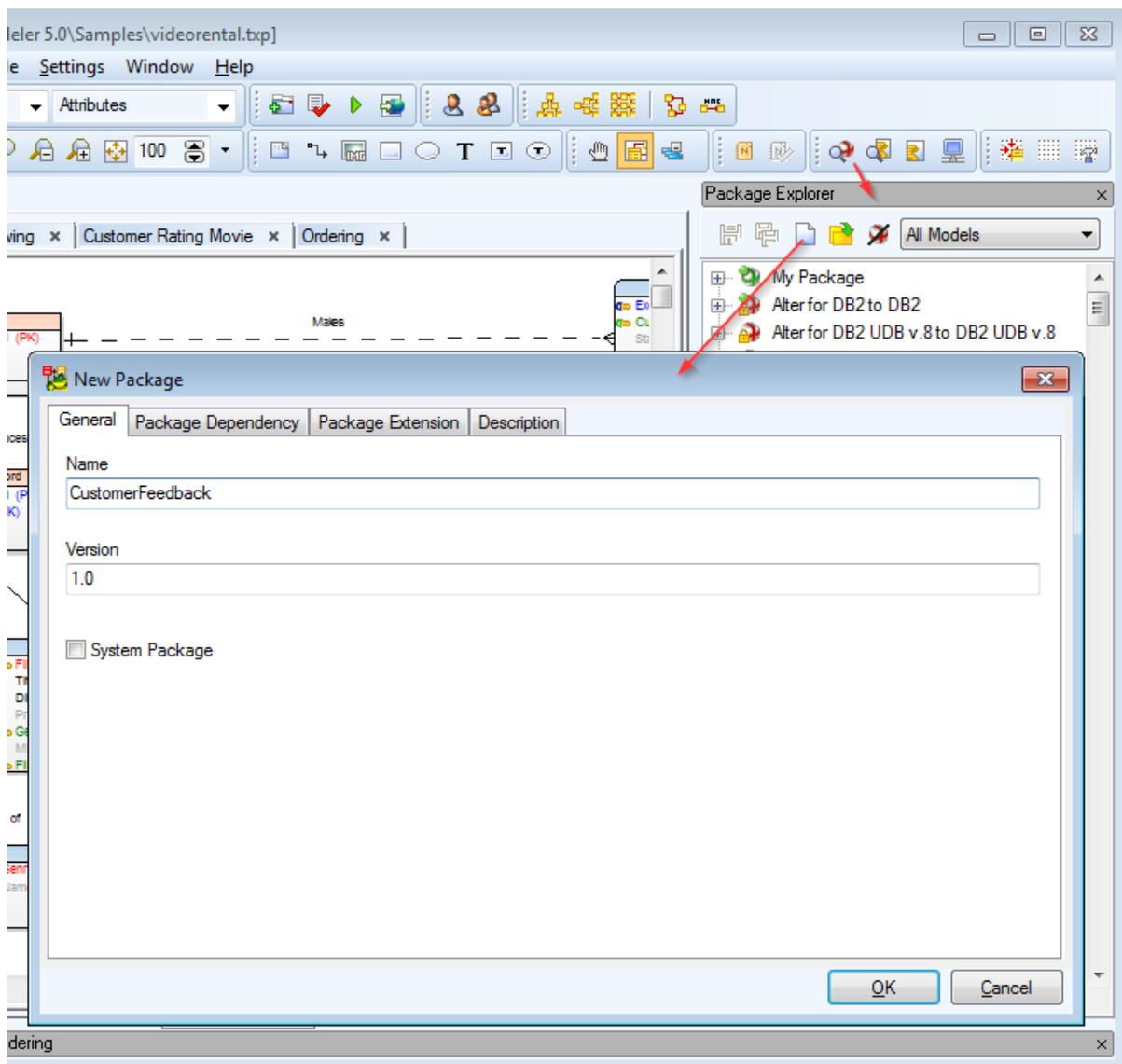
function main()
{
    var app = System.GetInterface('Application');
    var Model = app.Models.GetObject(0);
    for (e=0; e<Model.Entities.count; e++)
    {
        Entity = Model.Entities.GetObject(e);
        for (a=0; a<Entity.Attributes.count; a++)
        {
            Attr = Entity.Attributes.GetObject(a);
            for (i=0; i<Attr.PKForeignKeys.count; i++)

```





Define Name of the package.



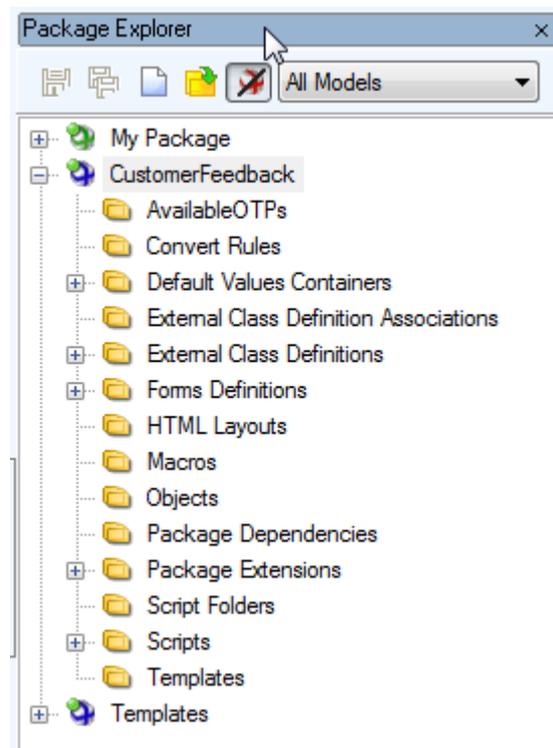
Select packages on which the new package depends. In this example, the Customer Feedback will not be dependent package. It will just extend existing packages.

Example of dependent package: Package **RE Microsoft SQL Server 2005** depends on **RE Microsoft SQL Server** package and extends **Microsoft SQL Server 2005** package. (RE is abbreviation for Reverse Engineering.)

Select packages you want to extend. In our example, we will be extending **Database Oracle 10g** package and **HTML Reports for Oracle 10g**.

You can write description to the **Description** tab.

Newly created package will appear in the **Package Explorer**. You can also see package extensions there. Custom packages have blue icon.



Just to compare, see My Package (where all modifications are stored if you don't use add-on packages) - it has a green icon and is listed at the top.

Well, a new package exists, let's continue adding new properties.

For more information, see [Add New Properties in Metamodel](#) on page 493.

## Add New Properties in Metamodel

Properties and methods can be added visually, via Metamodel.

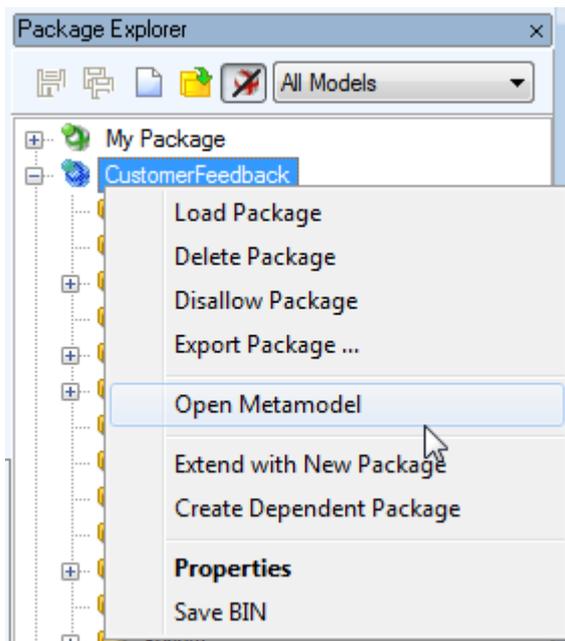
For our purpose, we will need two new properties.

- **ConfirmedByCustomer** (boolean)
- **NotesFromCustomer** (string)

This is where our values will be stored. The properties will be assigned to items that will appear in **Entity Properties** form. The ConfirmedByCustomer property value will be assigned to a checkbox, and the NotesFromCustomer value will be assigned to a text box.

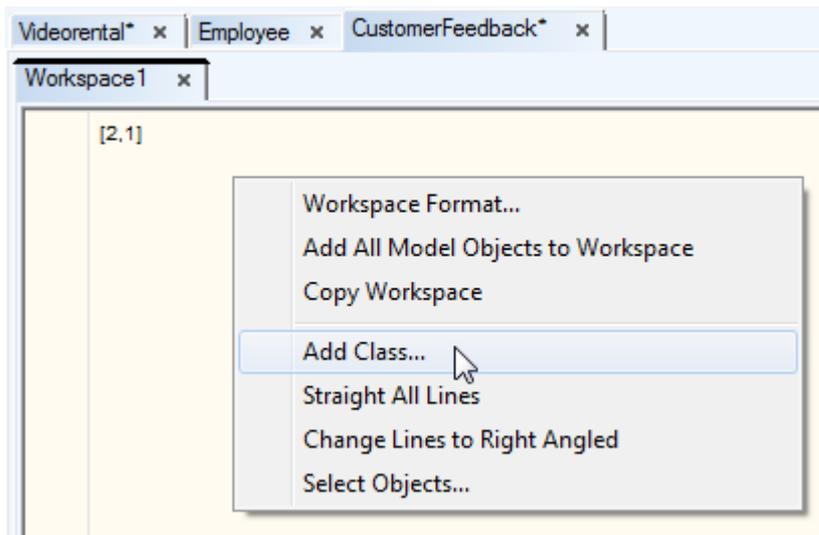
## How To Add New Properties

Right-click the CustomerFeedback package and select **Open Metamodel**.



Empty metamodel diagram opens.

Right-click the workspace and select **Add Class...**



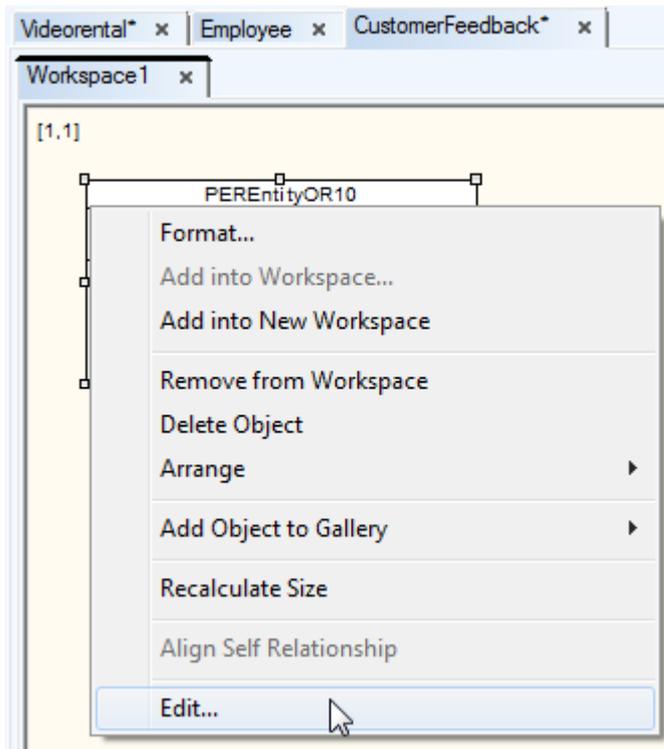
Select class you want to extend. For our purpose, we need to extend **PEREntityOR10** class.

**PER** - Physical Entity Relationship model.

**Entity** - Items must be accessible in the Entity Properties form.

**OR10** - Modification will be made for Oracle 10g only.

Select the newly added class and right-click it. Select **Edit**.

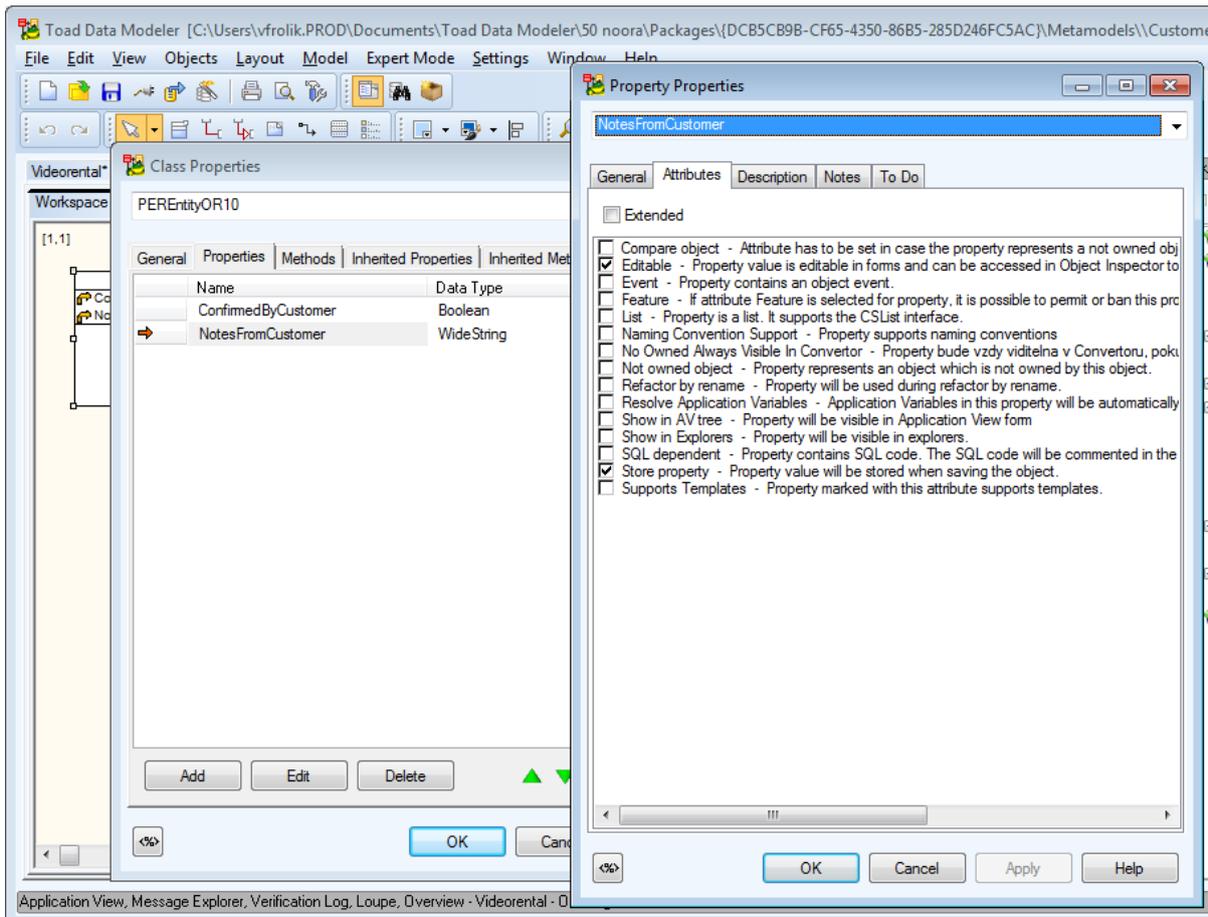


Add two new properties to the class.

**ConfirmedByCustomer**, data type **Boolean**.

On tab **Attributes** of the **Property Properties** dialog, you can define property Attributes. Select **Editable** (we need to be able to edit the values) and **Store Property** (we want to store the values with model).

Now define the second Property **NotesFromCustomer**, on tab **Attributes**, select the same attributes - **Editable** and **Store Property**.



Properties have been added to the metamodel. **Now it's necessary to save the metamodel and restart the application.** Then we can continue modifying **Entity Properties** form.

**i** Note: Metamodels are XML documents stored as .TXM files.

For more information, see [Modify Form](#) on page 498.

## Creating New Objects

In this topic you can find information about how to create new objects via scripting.

### Creating a New Entity

```
function main()
{
var app = System.GetInterface('Application');
var Model = app.Models.GetObject(0);
Model.Lock();
```

```

var Entity = Model.CreateNewObject( 2002 ); // 2002 is object type
of PEREntity

Entity.Name = 'Customer';

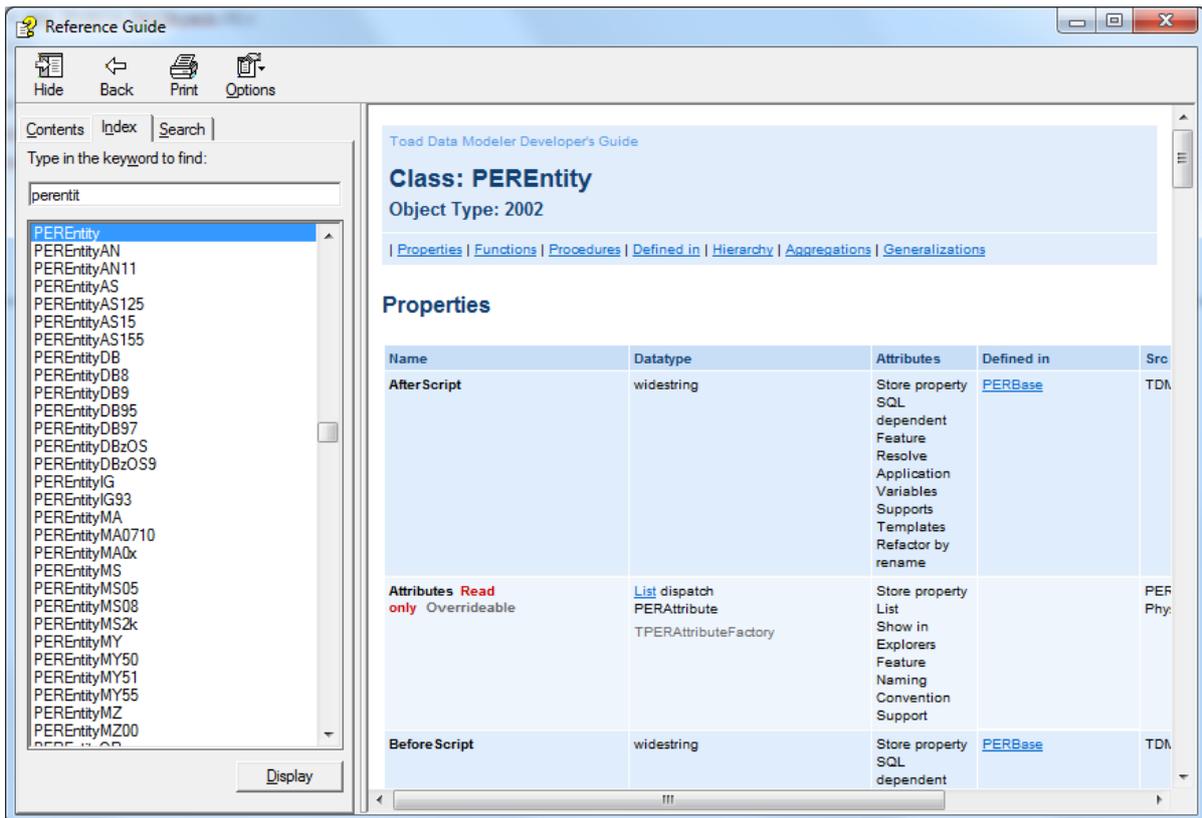
Model.Unlock();

Model.RefreshModel();

}

```

More information about the Entity object can be found in the Reference Guide (in Expert Mode main menu, Expert Mode has to be enabled first). See PEREntity class or PEREntityOR (for Oracle) class etc.



## Creating a New Index

```

function main()
{
var app = System.GetInterface("Application");
var Model = app.Models.GetObject(0); // gets first model in application
Model.Lock();
var Entity = Model.Entities.GetObject(0); // gets first entity in model

```

```

Attribute = Entity.Attributes.GetObject(0); // gets first attribute
in entity

var Index = Entity.CreateNewObject( 2012 ); // 2012 is object type
of PERIndex

Index.Name = "IXName";

var IndexItem = Index.CreateNewObject ( 2013 ); //2013 is object type of
PERIndexItem

IndexItem.Attribute = Attribute;

Model.Unlock();

Model.RefreshModel();

}

```

## Creating a New Domain

```

function main()

{

var app = System.GetInterface('Application');

var Model = app.Models.GetObject(0);

var Domain;

var DataType_Number = Model.ModelDef.DataTypes.GetObjectById("{3A22E4F9-
EE24-4A39-835D-62C3EF76CAA4}"); // Number(x,y);

Model.Lock();

Domain = Model.CreateNewObject( 2006 ); // 2006 is object type of
PERDomain

Domain.Name = 'MyDomain';

Domain.SetLinkedObject("DataType", DataType_Number);

Domain.DataTypeParam1 = "10";

Domain.DataTypeParam2 = "2";

Model.Unlock();

}

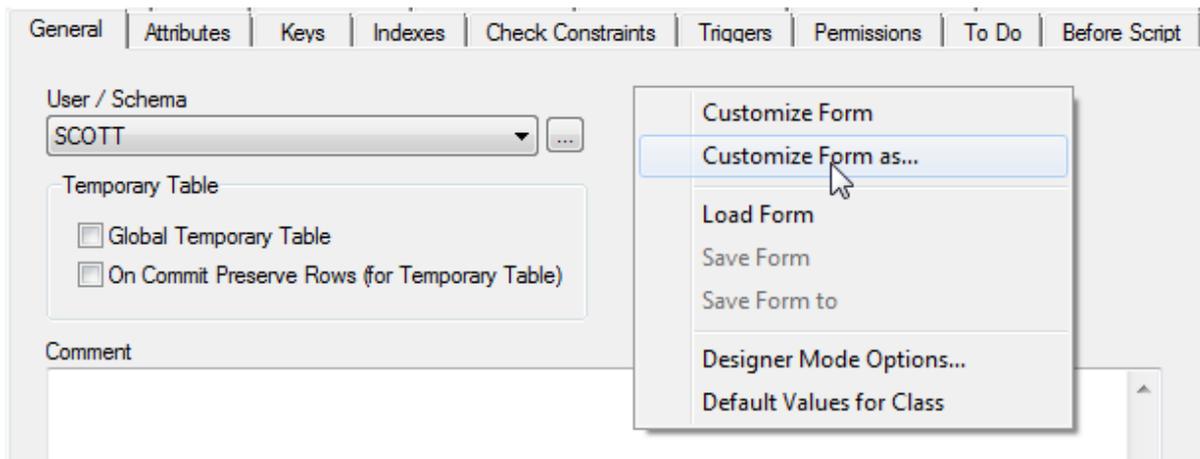
```

## Modify Form

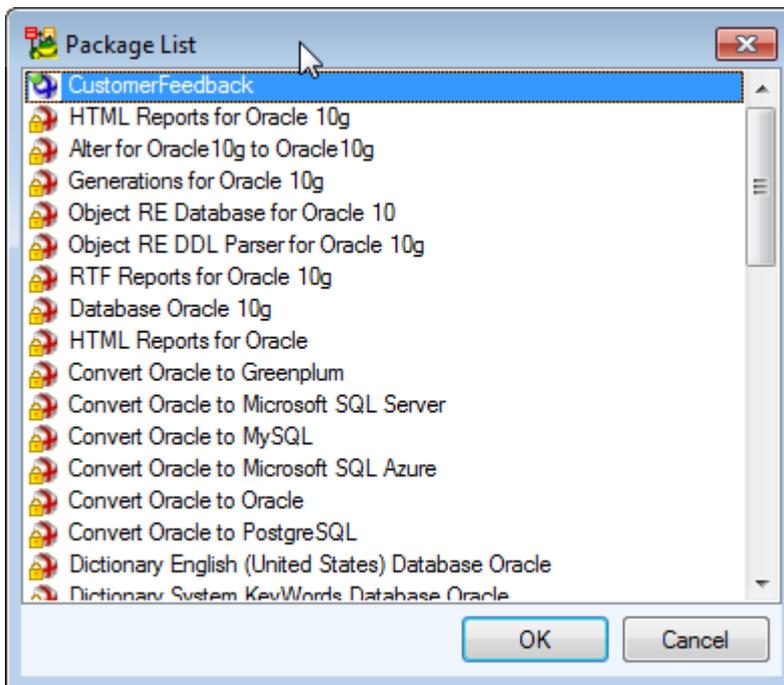
Back to our Oracle 10 physical model.

Edit an entity to open standard **Entity Properties** form.

Right-click the form and select **Customize Form as...**



Select a package you want to store modifications in.



The following forms and palettes will appear. Note that the **Entity Properties** form has dotted grid now. In the **Form Explorer**, see that the form name is **FmPEREntityEdit**. We will need this information later. To add a new tab to the Entity form, right-click any tab in the form and select **New Page**.

Define Caption for the new tab in **Component Inspector**.

Then select **DataCheckBox** item from the **Component Palette**.

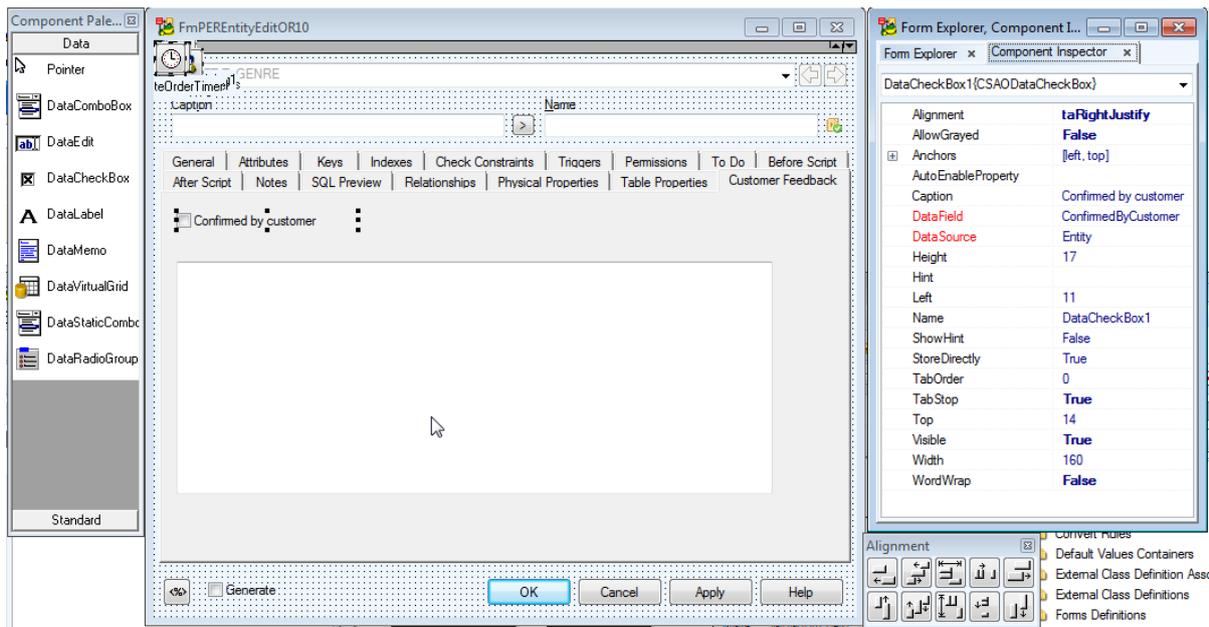
And add a new checkbox to the **CustomerFeedback** tab.

Select the **Entity** item from **DataSource** field in the **Component Inspector**.

Then select **ConfirmedByCustomer** item from **DataField**.

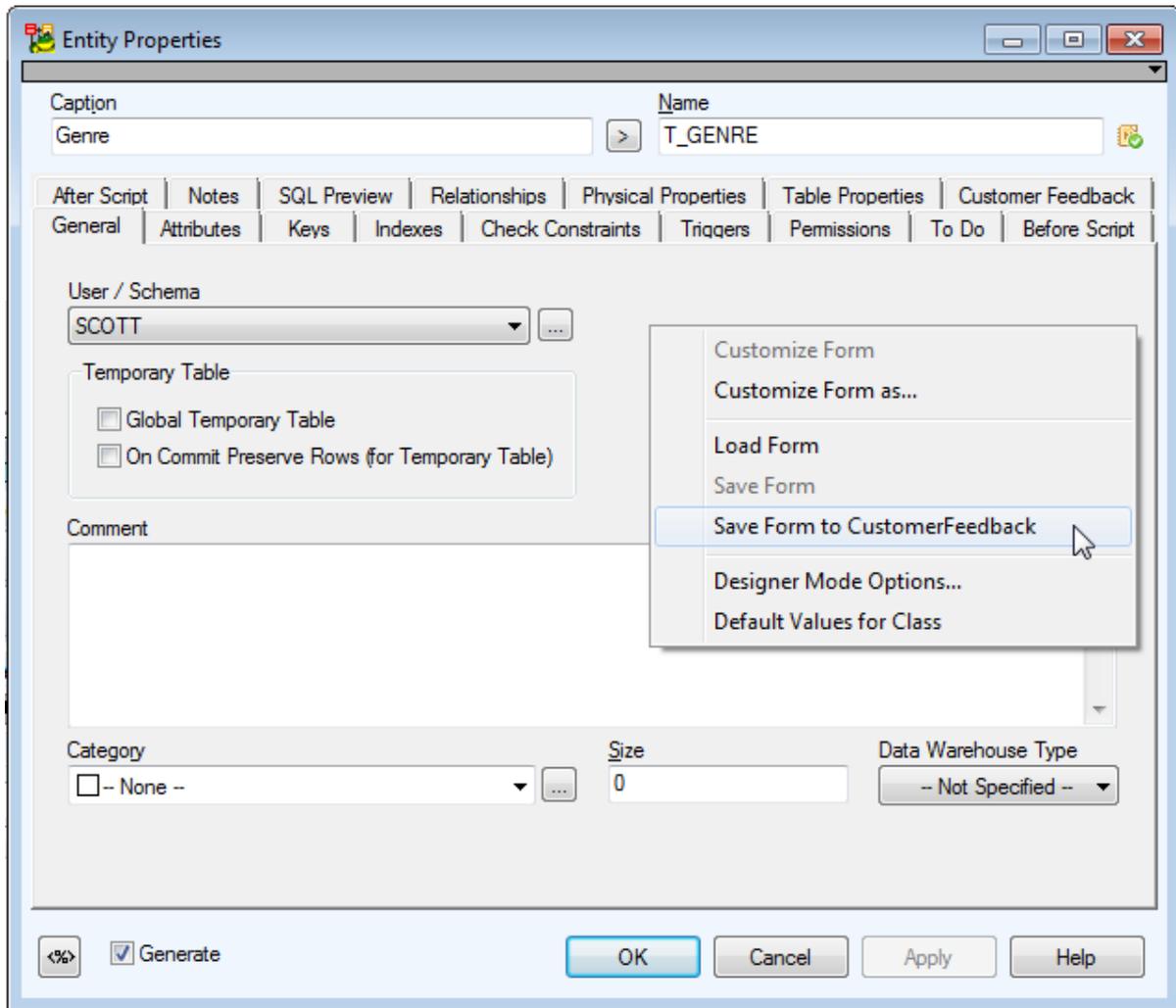
Select **DataMemo** from **Component Palette** and add new text field (datamemo) item to the **Customer Feedback** tab.

Result:



Close the **Entity Properties** form by clicking the red X button at top of the form. Component Inspector, Component palette, Form Explorer will disappear.

Then right-click the form and select **Save Form to CustomerFeedback**.



New items are in the form.

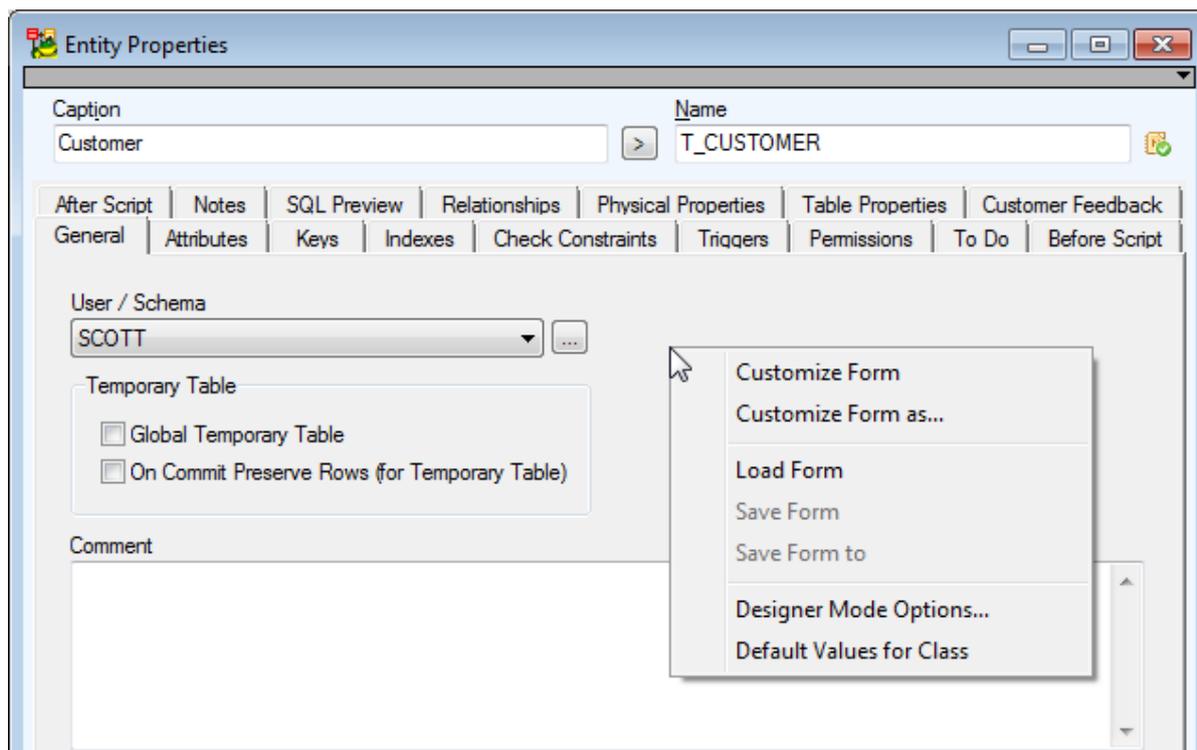
For more information, see [Set Default Values](#) on page 501.

## Set Default Values

### How to Define Default Values

If you need to change the default value for new items, do the following:

Right-click the form (**Entity Properties** form in our example) and select **Default Values for Class**.



We want the **Confirmed by customer** checkbox to be selected by default for new entities. Select property name and click the **Default Value** column. Then press F2 to edit the value. Select where the definition will be stored. In our example, we need to store it into the **CustomerFeedback** package. Click the dialog to confirm your selection in combo box and then confirm **OK**. Done. For more information, see [Add Events](#) on page 502.

## Add Events

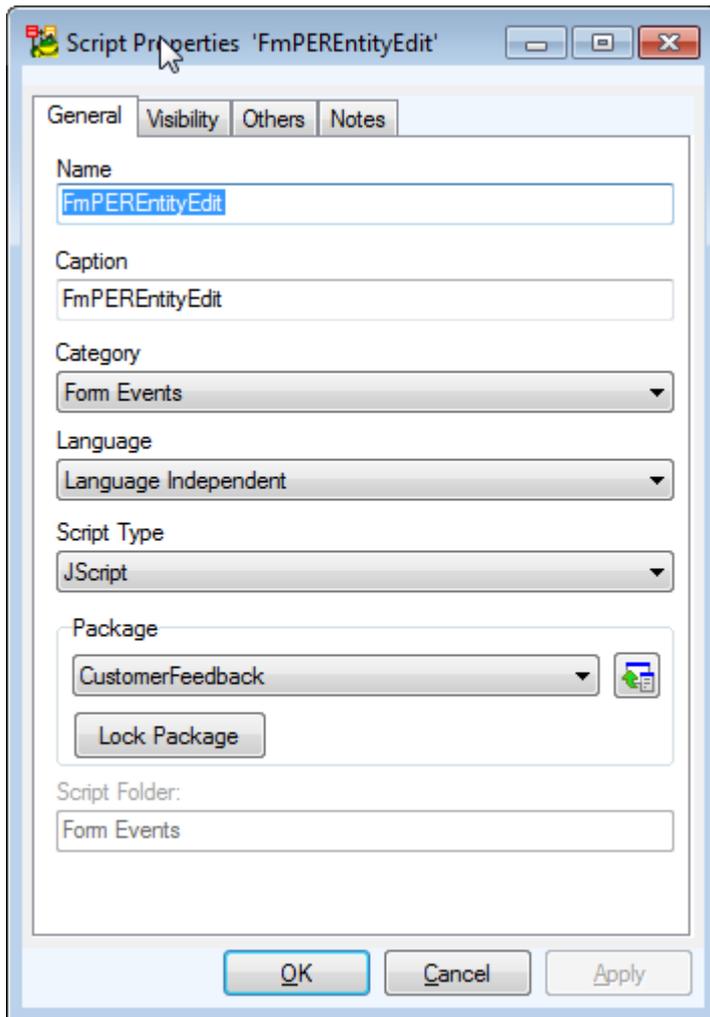
The following events are available in Toad Data Modeler scripting:

- OnCreate
- OnCheck
- OnChange
- OnClick
- OnClickSilent
- OnClose

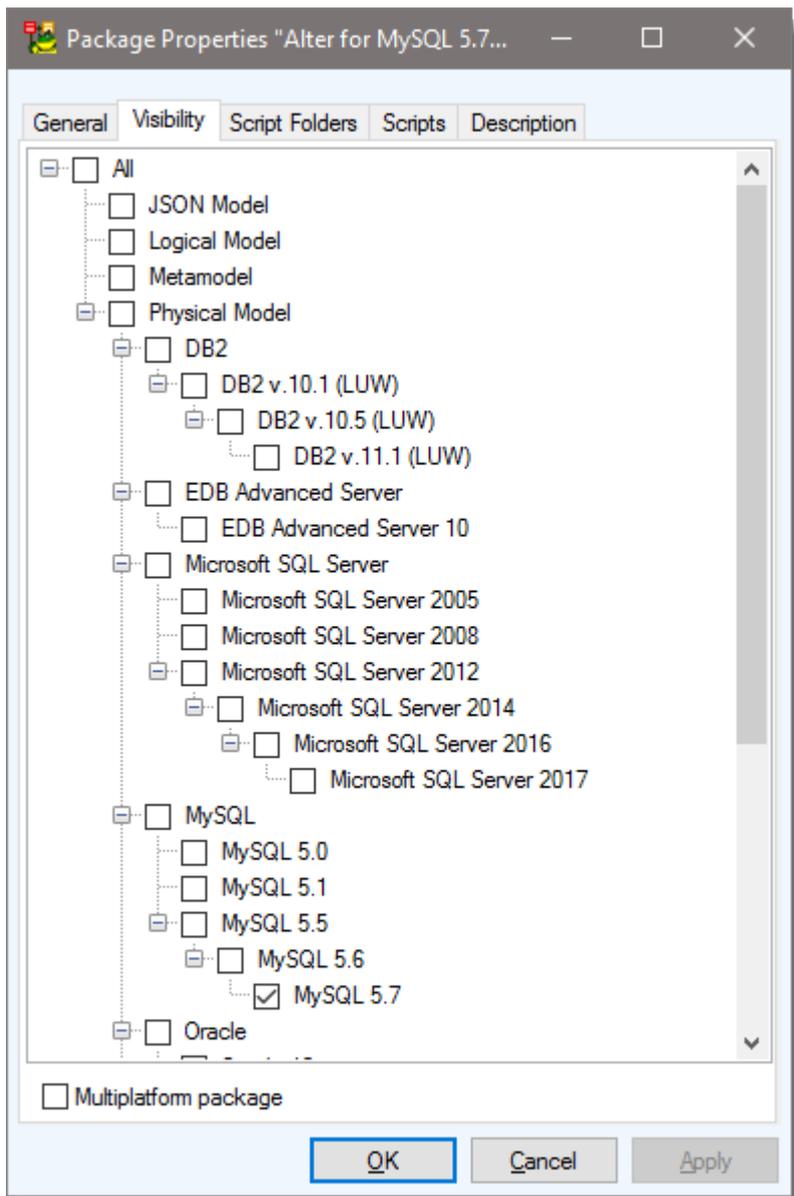
Let's add **OnCheck** event to the checkbox on the **Customer Feedback** tab of the **Entity Properties** dialog. When the checkbox is selected, the text box with **Notes from Customer** will be visible. When the checkbox is unchecked, the text box will disappear.

In Package Explorer, select the Script folder under the CustomerFeedback package. Right-click it and select **AddScript**.

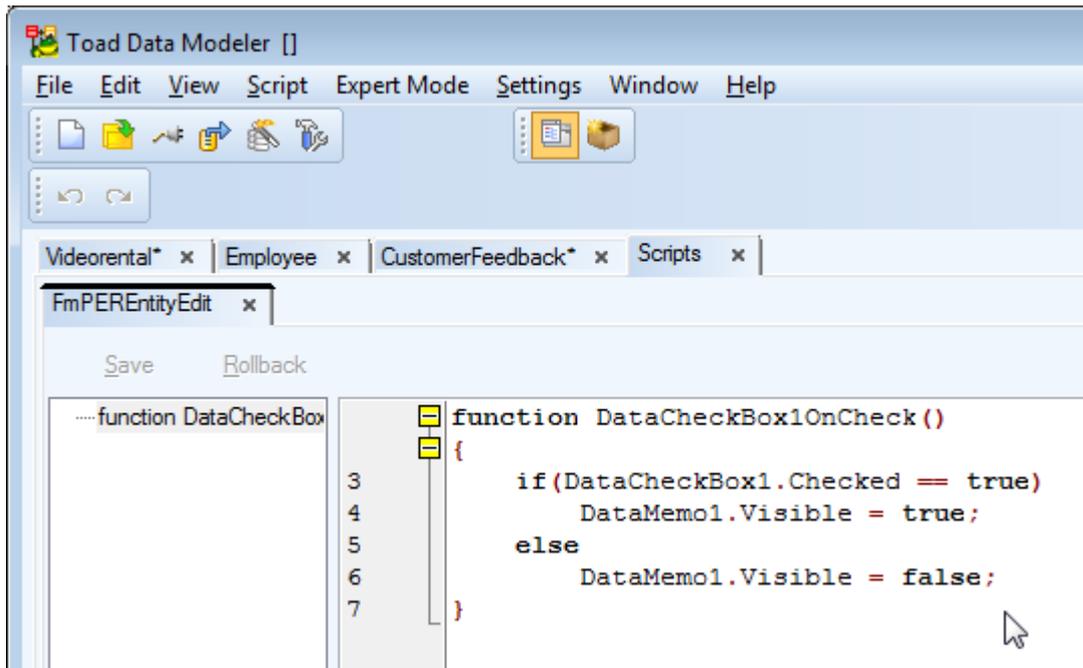
Set the script name to **FmPEREntityEdit**. - This is the name of the form we want to write the script for. See the "Modifying a Form" topic to find out where the form name is defined.



Set visibility to Oracle 10g only.



Close the window, right-click the script again and select **Edit Source Code**.  
Add there the event function.



Code:

```
function DataCheckBox1OnCheck()
{
    if(DataCheckBox1.Checked == true)
        DataMemo1.Visible = true;
    else
        DataMemo1.Visible = false;
}
```

Explanation of items in bold.

#### **DataCheckBox1OnCheck()**

- DataCheckBox1 - name of item that has been added to the Entity form.
- OnCheck - name of event.

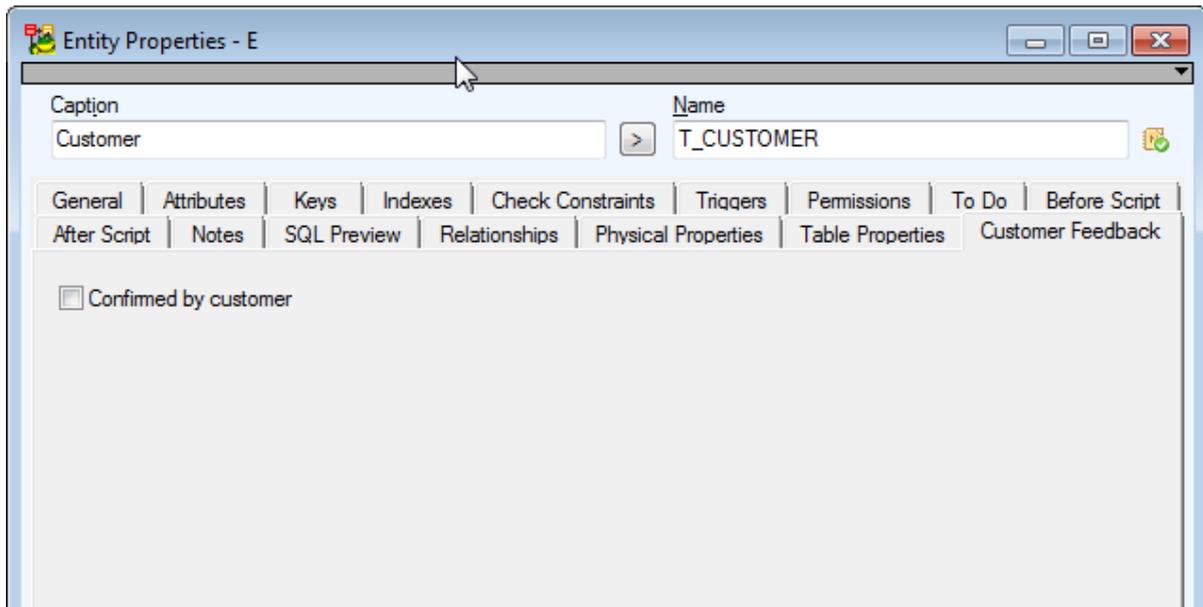
#### **DatacheckboxBox1** and **DataMemo1**

- Both are names of items that were added to the Entity form. [Modify Form](#)

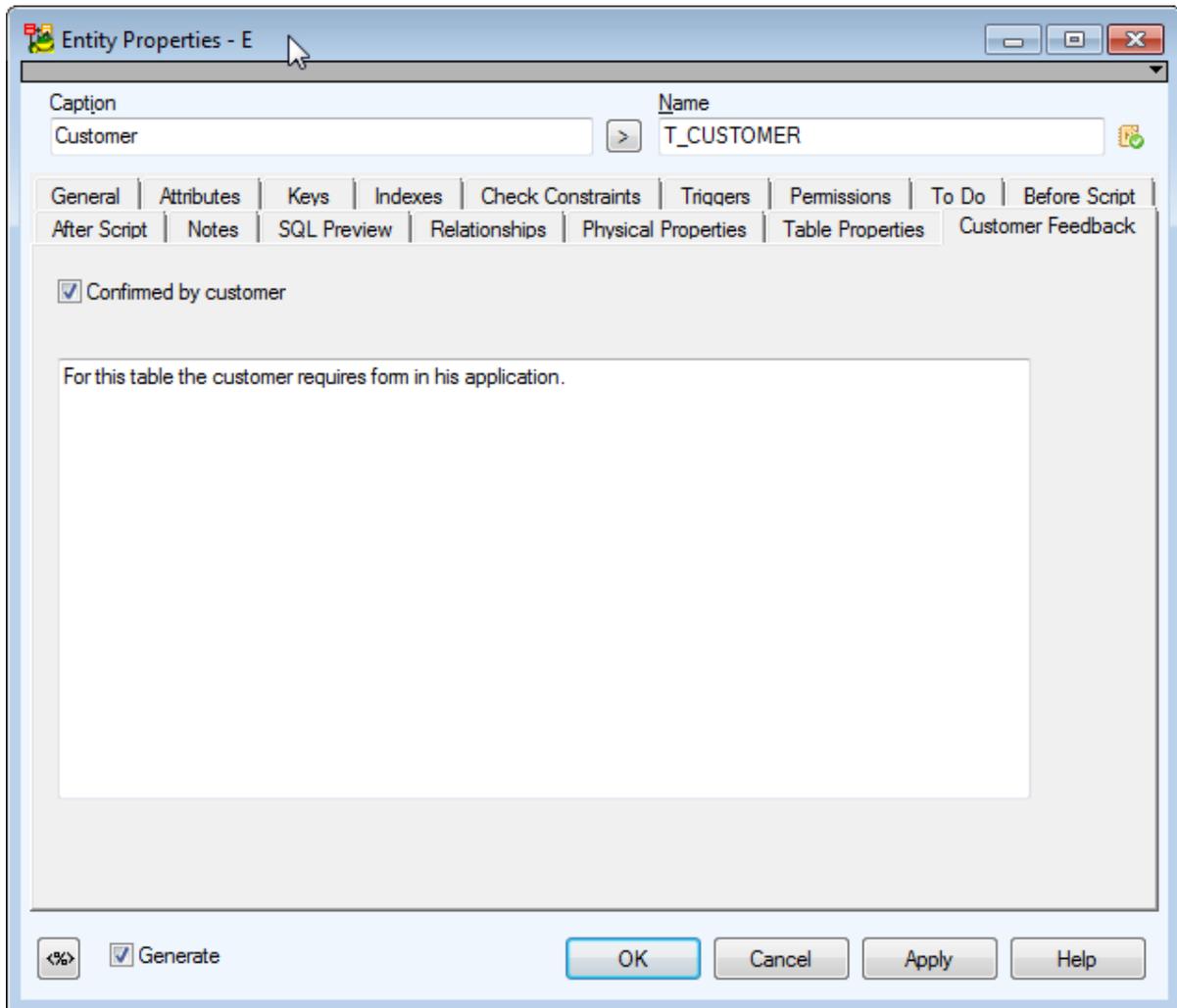
Press **Commit** to confirm the script.

Result:

When the checkbox is unchecked, the text area is hidden.



If you select the checkbox, the text area will display.



For more information, see [Access Property Values via Scripting Window](#) on page 509.

## Dialogs

Samples for Message Dialogs:

### ShowMessageDialog

```
System.ShowMessageDialog(1004,'WarningDialog','Please select shapes on your Workspace before running the macro.',2,4);
```

#### Dialog type index

- 0 - warning
- 1 - error
- 2 - info
- 3 - confirm

4 - no icon

### Dialog buttons index

0 - no button

1 - yes

2 - no

3 - yes/no

4 - ok

5 - yes/ok

6 - no/ok

7 - yes/no/ok

8 - cancel

9 - yes/cancel

10 - no/cancel

11 - yes/no/cancel....

## ShowMessageDialogScript

This way you can create dialog with hyperlinks at the bottom.

```
var DlgParams = System.CreateObject('DialogParams');

DlgParams.Caption = 'Add Entities Info'; // Name appears in Settings |
Options in section Dialog Boxes.

DlgParams.DialogIndex = 202; // Unique number, must be above 200

DlgParams.Msg = 'This macro allows you to quickly add entities to your
model. Specify one entity caption per line. ';

DlgParams.Msg += 'Spaces in entity captions can be converted to entity
names as underscore characters. ';

DlgParams.Msg += 'For more infomation click the Help link at bottom. Do
you wish to continue?';

DlgParams.Buttons = 3;

DlgParams.DlgType = 2;

DlgParams.HyperLink =
'http://www.casestudio.com/help/ProductivityPack.aspx';

DlgParams.HyperLinkCaption = 'Help';

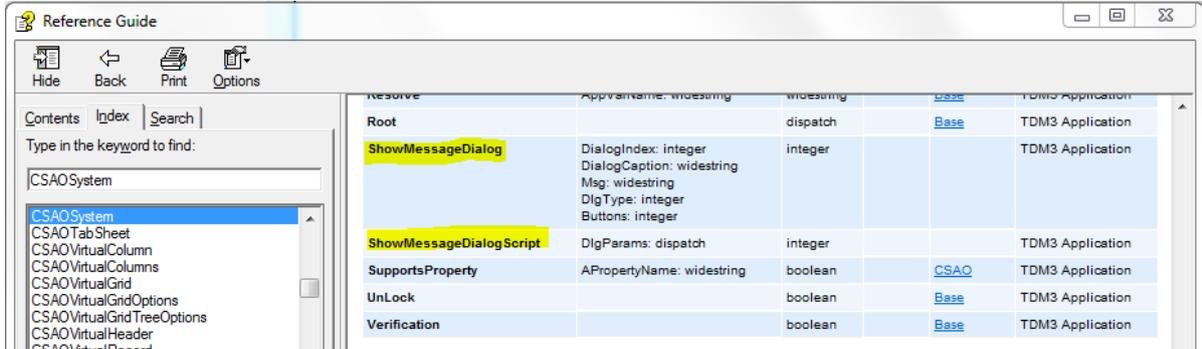
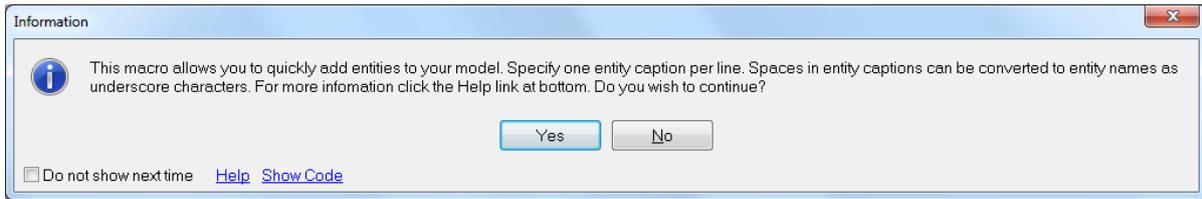
DlgParams.ScriptName = 'AddEntitiesMacro';

if(System.ShowMessageDialogScript(DlgParams) != 6)

{

return;
```

}



## Access Property Values via Scripting Window

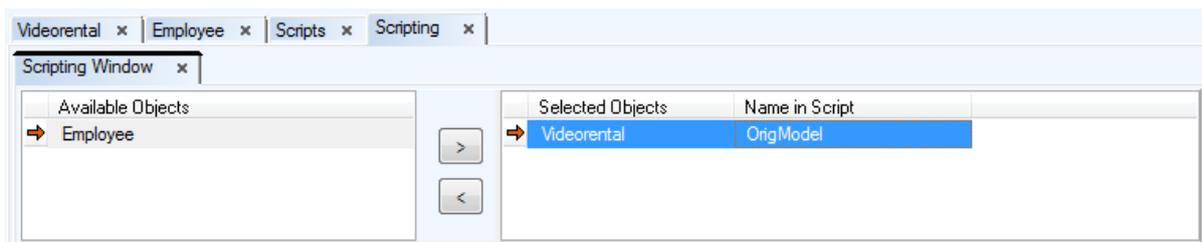
You can write scripts in Toad Data Modeler, save the scripts to packages, distribute the packages etc. - This will be explained later. Now you will see how to work with **Scripting Window** that allows you to run scripts at once, without the necessity to have them stored in packages.

Click **Expert Mode | Scripting Window** to open it. (Of course, Expert Mode has to be turned on.)

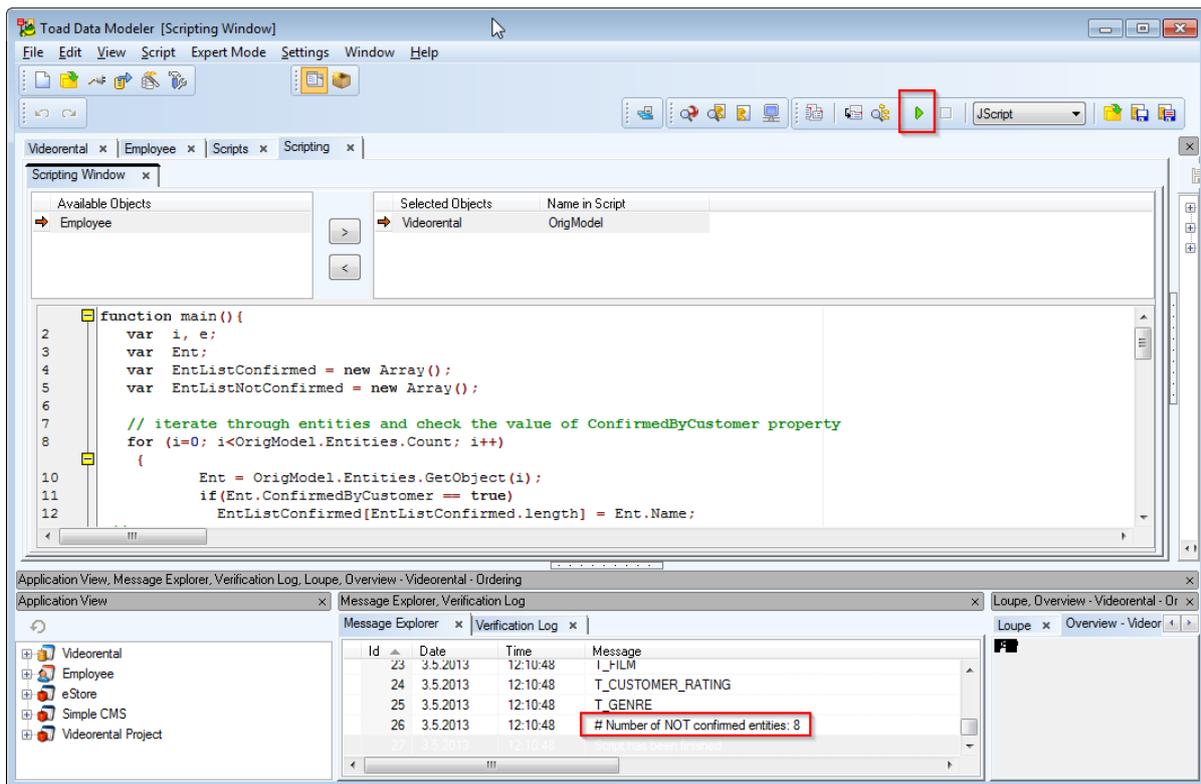
The following dialog appears. If you don't see the upper part of the **Scripting Window**, select **View | Show Registered Objects**.

On the left, you can see available models. Use the arrows to select model you want to work with. In our example, we will execute script for *Videorental model* (for Oracle 10g).

In the **Name in Script** column, you can define name that will be used in the script. Our OrigModel value will represent the selected Videorental model.



Write script to the main() function.



## Code:

```
function main(){

    var i, e;

    var Ent;

    var EntListConfirmed = new Array();
    var EntListNotConfirmed = new Array();

    // iterate through entities and check the value of ConfirmedByCustomer property
    for (i=0; i<OrigModel.Entities.Count; i++)
    {

        Ent = OrigModel.Entities.GetObject(i);

        if(Ent.ConfirmedByCustomer == true)

            EntListConfirmed[EntListConfirmed.length] = Ent.Name; // add to list of
            confirmed entities
    }
}
```

```

else

    EntListNotConfirmed[EntListNotConfirmed.length] = Ent.Name; // add to list of not
confirmed entities

}

// write list of confirmed entities to Log.
Log.Information ("-----");
Log.Information ("List of entities confirmed by customer");
Log.Information ("-----");
for (e=0; e<EntListConfirmed.length; e++)
{
    Log.Information(EntListConfirmed[e]);
}
Log.Information ("# Number of confirmed entities:
"+EntListConfirmed.length.toString());

// write list of NOT confirmed entities to Log.
Log.Information ("-----");
Log.Information ("List of entities NOT confirmed by customer");
Log.Information ("-----");
for (e=0; e<EntListNotConfirmed.length; e++)
{
    Log.Information(EntListNotConfirmed[e]);
}
Log.Information ("# Number of NOT confirmed entities:
"+EntListNotConfirmed.length.toString());

}

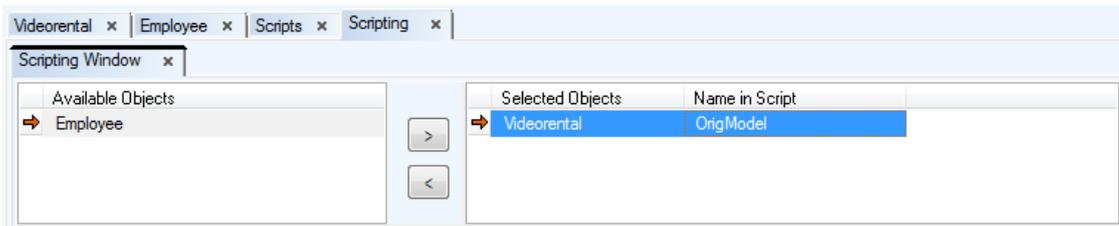
```

## Where to find information about objects and their properties and methods?

### Explanation of Items in Bold:

**OrigModel.Entities.Count**

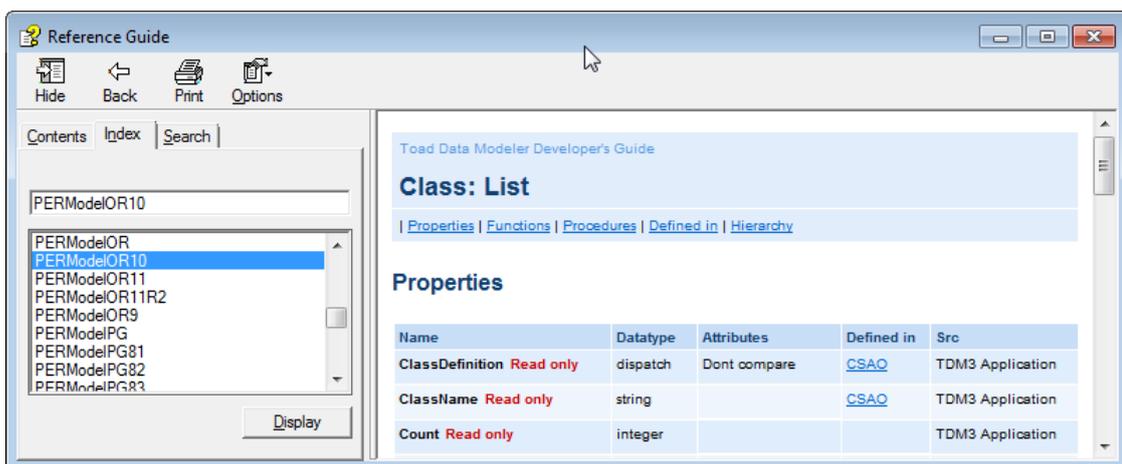
- **OrigModel** - represents object assigned in the upper part of the Scripting Window (Videorental object renamed to OrigModel).



- **Entities** - we work with Physical Entity Relationship model, therefore we need to search for PER object. Model is for Oracle 10g, let's find the PERModelOR10 object in the Reference.



- **Count** - represents a feature that is available for all List objects. On the screenshot above, you can see that the Entities datatype is a List. Let's click the List link and see details of the List class.

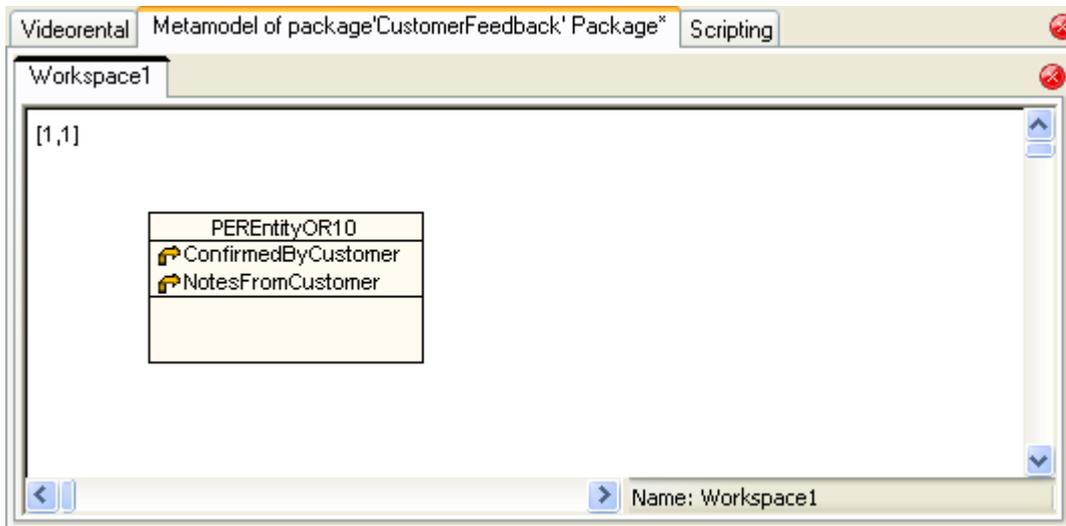


### OrigModel.Entities.GetObject(i)

- **GetObject** - belongs to the List class.

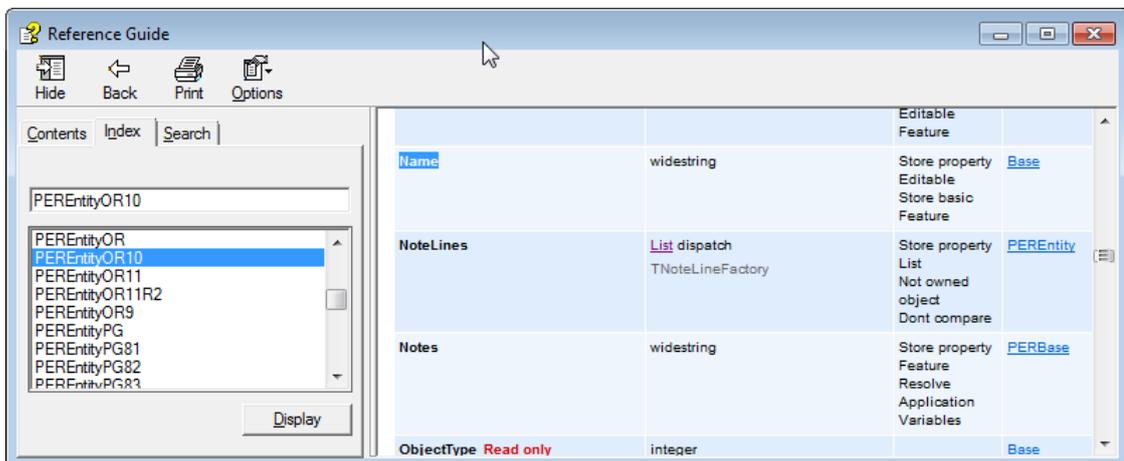
## Ent.ConfirmedByCustomer

- **Ent** - is a variable that holds assigned Entity objects (assigned earlier using the **OrigModel.Entities.GetObject(i)** function).
- **ConfirmedByCustomer** - property of PEREntityOR10 object, added to Metamodel of the **CustomerFeedback** package.



## Ent.Name

- **Ent** - is a variable that holds assigned Entity objects (assigned earlier using the **OrigModel.Entities.GetObject(i)** function).
- **Name** - property of PEREntityOR10 object. We still work with PER model and now we need to find property of Entity in Oracle 10g model. Let's see properties of the PEREntityOR10 object.

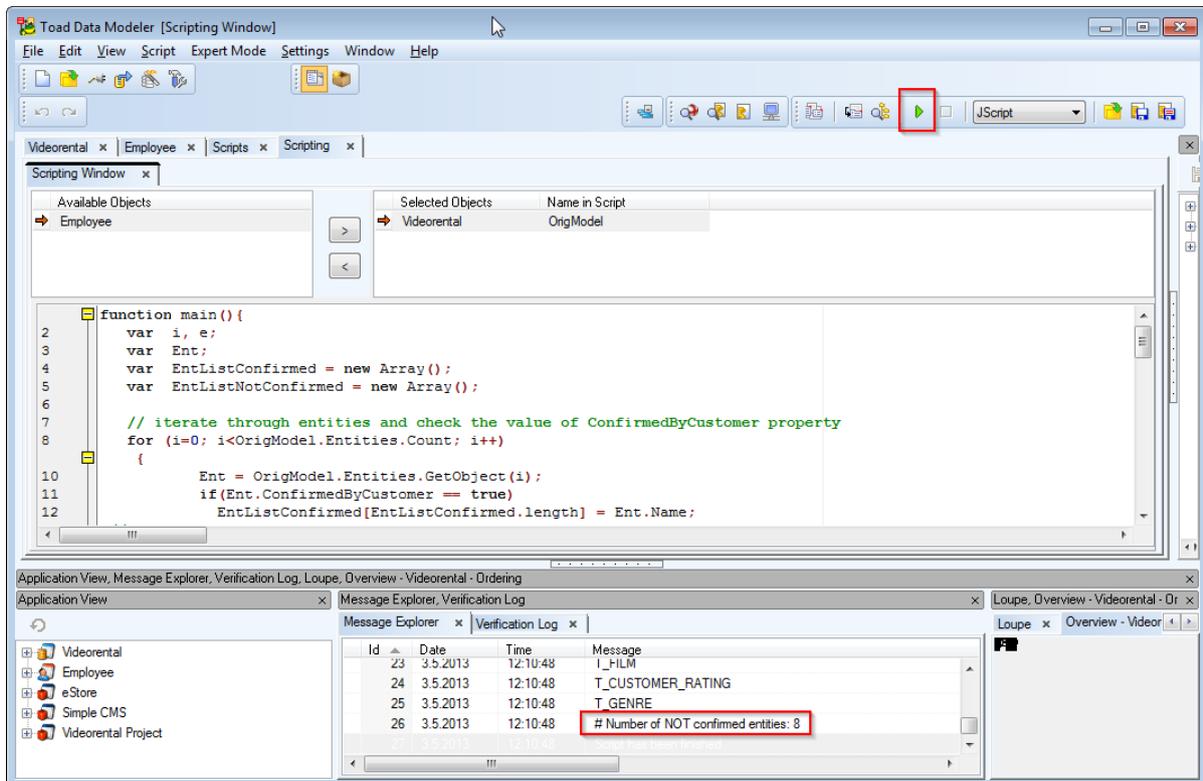


## .length and .toString()

- both are standard JavaScript items.

## Executing the Script

Click **Execute Script** . Result will be displayed in the **Message Explorer** and **Log** area.



For more information, see [Create Script](#) on page 515.

## File System Scripts

You can create new files or folders using simple javascript code.

```
function CreateFolder(folder)
{
    var fso;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    fso.CreateFolder (folder);
}

function CopyFolder(sourceFolder, destinationFolder, overwrite)
{
    var fso;
```

```

fso = new ActiveXObject("Scripting.FileSystemObject");
fso.CopyFolder (sourceFolder, destinationFolder, overwrite);
}

function CopyFile(sourceFile, destinationFile)
{
var fso;

fso = new ActiveXObject("Scripting.FileSystemObject");

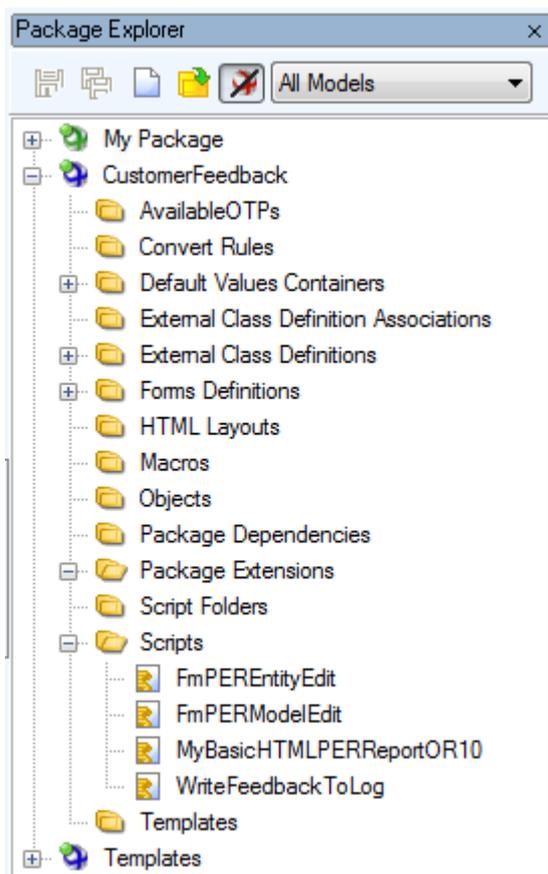
fso.CopyFile (sourceFile, destinationFile);
}

```

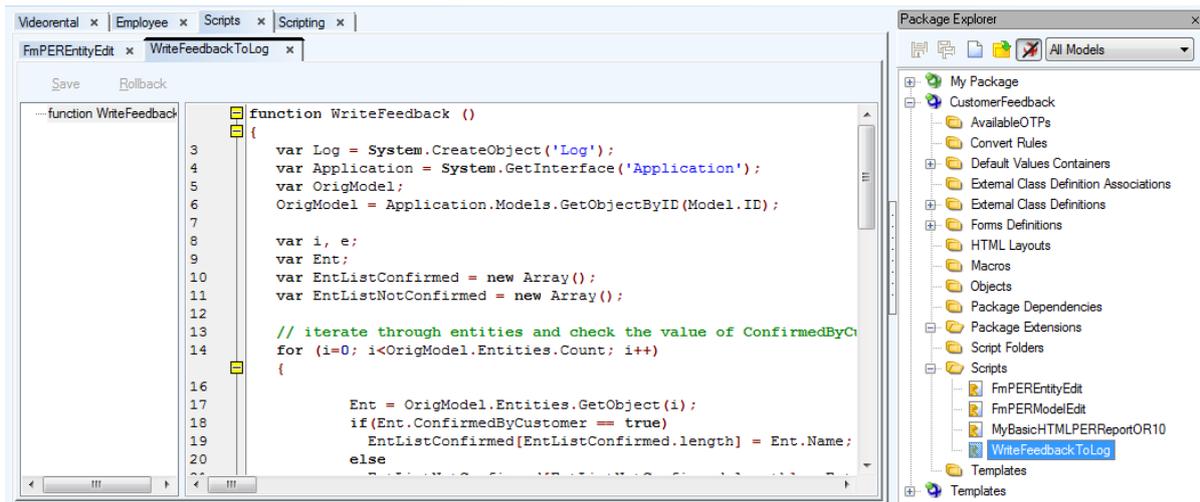
## Create Script

You know how to execute scripts from the Scripting Window. If you want to store the script and call it from another form in the application, for example, do the following:

Create a new script WriteFeedbackToLog. See the "Adding Events" topic to find out how to create new scripts.



Write there function WriteFeedback.



## Code

```
function WriteFeedback ()
{
    var Log = System.CreateObject('Log');
    var Application = System.GetInterface('Application');
    var OrigModel;

    OrigModel = Application.Models.GetObjectByID(Model.ID);

    ....

    ....
}

```

## Explanation

The WriteFeedback function is almost identical to the Main function we were executing from the Scripting Window. The only difference is in the definition of OrigModel object. In the **Scripting Window**, we could select Videorental and define the OrigName name.

However, now we have no means to select the object visually (and we do not need it, the function will be executed for active model). Therefore we need to define the OrigModel object via Application.Models.GetObjectByID method, with parameter Model.ID.

This way we can get the currently active model.

We also need to register object Log. (It is not necessary to register Log in the Scripting Window. Log is registered in the Scripting Window automatically.)

The rest of the script is identical.

For more information, see [Call Existing Script from Model Properties Form](#) on page 519.

## Getting Settings Information

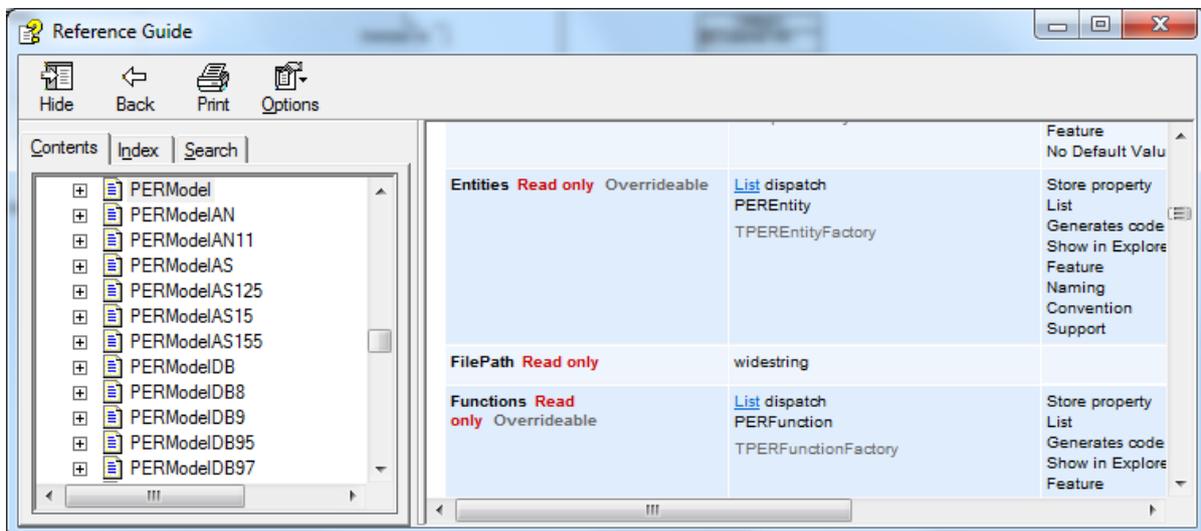
In Toad Data Modeler you can access Application Settings via **ApplicationConfig** class:

```
function main() {  
  
    var App = System.GetInterface("Application");  
  
    var Log = System.CreateObject("Log");  
  
    var Model = App.Models.GetObject(0);  
  
    Log.Information(App.ApplicationConfig.PackagePath);  
  
}
```

If you need to find out the path to the folder where your model is stored, use the property **FilePath** of the Model object (PERModel class):

```
function main() {  
  
    var App = System.GetInterface("Application");  
  
    var Log = System.CreateObject("Log");  
  
    var Model = App.Models.GetObject(0);  
  
    Log.Information(Model.FilePath);  
  
}
```

More information can be found in Reference Guide (in Expert Mode main menu, Expert Mode has to be enabled first).



## Iterate Entity And Attributes

This sample shows you how to iterate entities and attributes and how to recognize PK, PFK or FK attributes.

```

function main()
{
var app = System.GetInterface('Application');
var Model = app.Models.GetObject(0); // gets first model in application
var e, a, iterEntity, iterAttribute;
Model.Lock();
for (e=0; e<Model.Entities.Count; e++) // iterate entities
{
iterEntity = Model.Entities.GetObject(e);
iterEntity.Lock();
for (a=0; a<iterEntity.Attributes.Count; a++) // iterate attributes
{
iterAttribute = iterEntity.Attributes.GetObject(a);
if(iterAttribute.IsPrimaryKey == 1) // check if attribute is PK
{
if(iterAttribute.FKForeignKeys.Count !=0)
Log.Information(iterEntity.Name+'-'+iterAttribute.Name+'-PFK');
else
Log.Information(iterEntity.Name+'-'+iterAttribute.Name+'-PK');
}
else
{
if(iterAttribute.FKForeignKeys.Count !=0)
Log.Information(iterEntity.Name+'-'+iterAttribute.Name+'-FK');
}
}
iterEntity.Unlock();
}
Model.Unlock();
Model.RefreshModel();
}

```

```
}
```

## Call Existing Script from Model Properties Form

Edit the **Model Properties** form. See the "Modify Form" topic to find out how to edit existing form.

Add there a new button and remember the name of the form - **FmPERModelEdit**. The name can be found in the **Form Explorer**.

Defined caption for the button - **Write Customer Feedback To Log**.

Set the name of the button to **FeedbackButton**.

Create a new script with the name of the **Model Properties** form - **FmPERModelEdit**.

Write event function to the script.

## Code

```
function FeedbackButtonOnClick()  
  
{  
  
    WriteFeedbackToLog.WriteFeedback()  
  
}
```

### Explanation

- **FeedbackButton** = name of the button.
- **OnClick** = event.
- **WriteFeedbackToLog** = name of script that contains called function.
- **WriteFeedback** = called function.

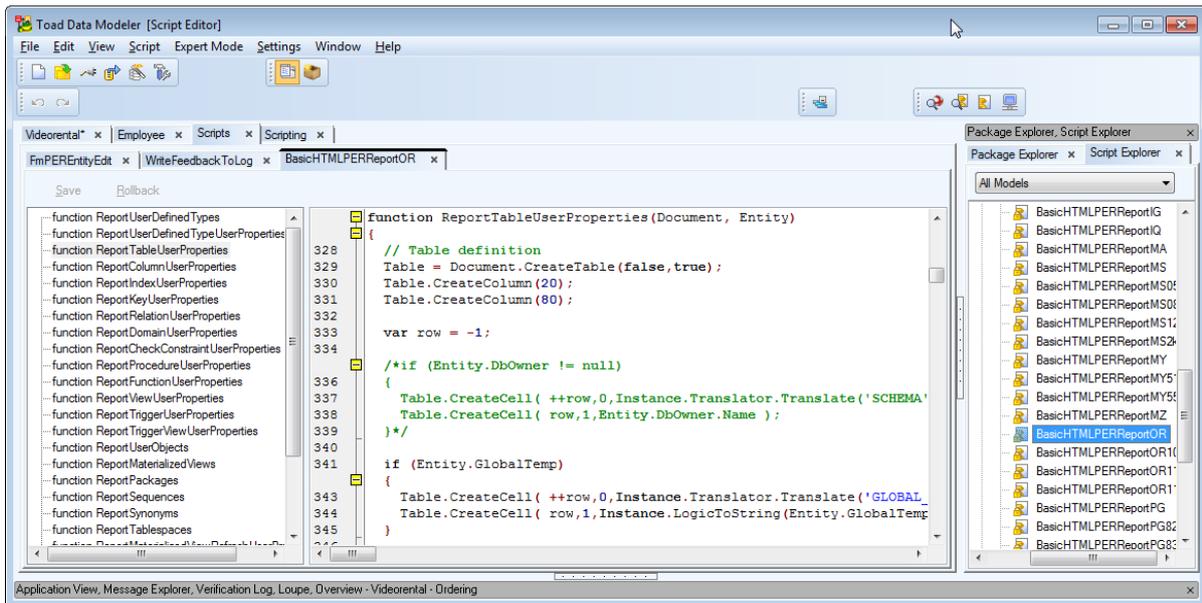
When you click the button, an output will be displayed in **Message Explorer** (Log).

For more information, see [Modify HTML Reports](#) on page 519.

## Modify HTML Reports

To modify HTML reports, we need to extend existing method. The first thing we need to do is to find out what script should be extended.

In **Script Explorer**, you can see **BasicHTMLPERReportOR** script with function **ReportTableUserProperties**. This is the script that generates Tables pages in HTML reports, specifically the section **Table Properties**.



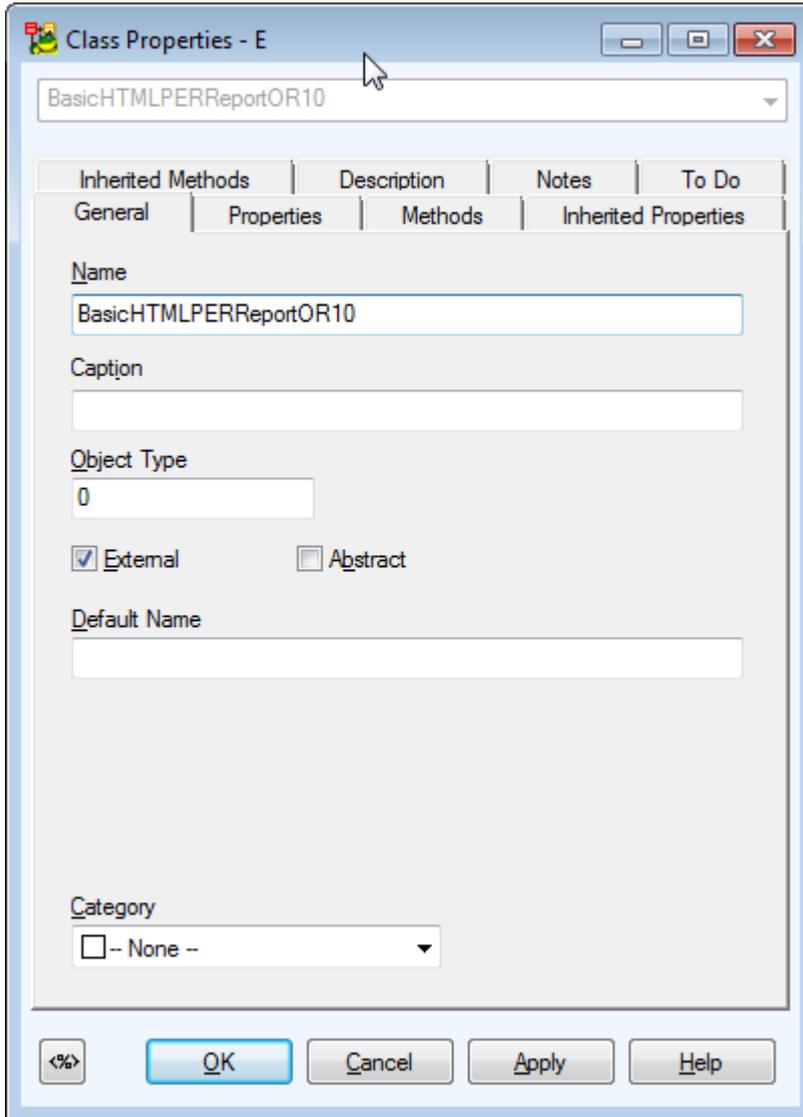
You can also see script **BasicHTMLPERReportOR10** that extends the **BasicHTMLPERReportOR** script.

Now we now need to write a script that will extend the **ReportTableUserProperties** function defined in the **BasicHTMLPERReportOR** script.

For that purpose, we need to open our **CustomerFeedback** metamodel and the method there.

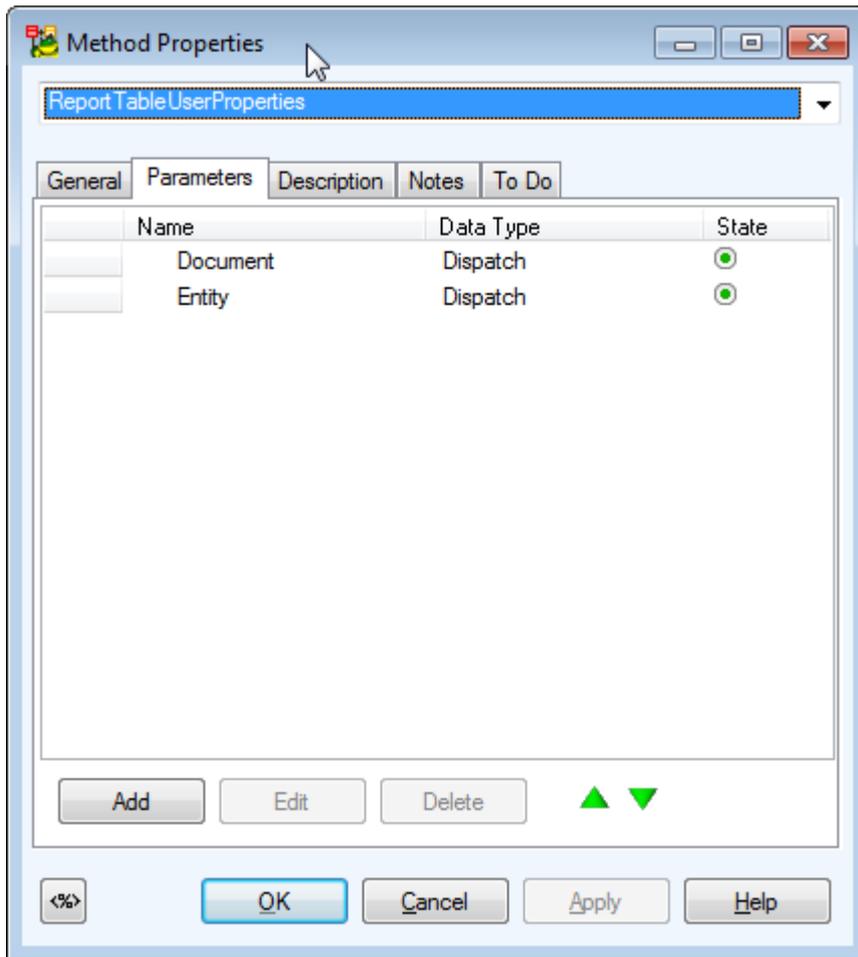
Open the **CustomerFeedback** metamodel, add there a new class (see the **Class** icon in the toolbar), edit the class and set the name to **BasicHTMLPERReportOR10**. (One extension of that class already exists, in our metamodel we will create another extension of the class).

Define **Object Type** (the value can be currently found out in metamodel to **HTML report for Oracle 10g** package).

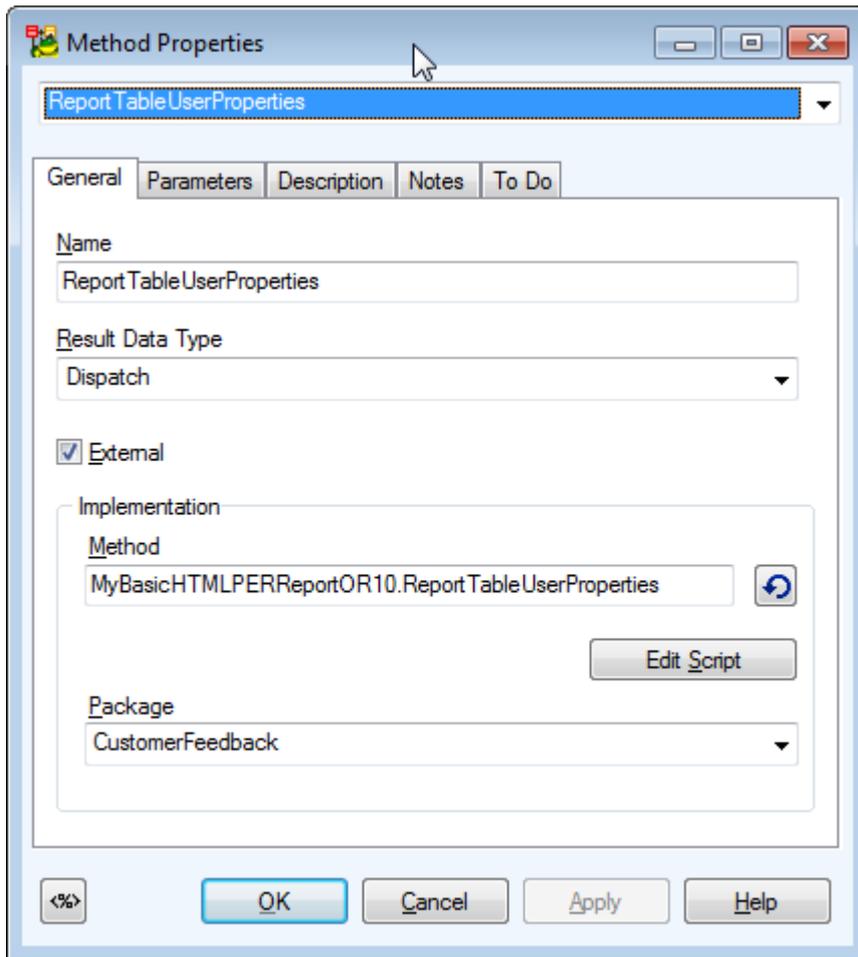


Add the **ReportTableUserProperties** method to the class.

Add two new parameters to the method. (The method name and number of parameters must be identical to the original method - see the first screenshot).

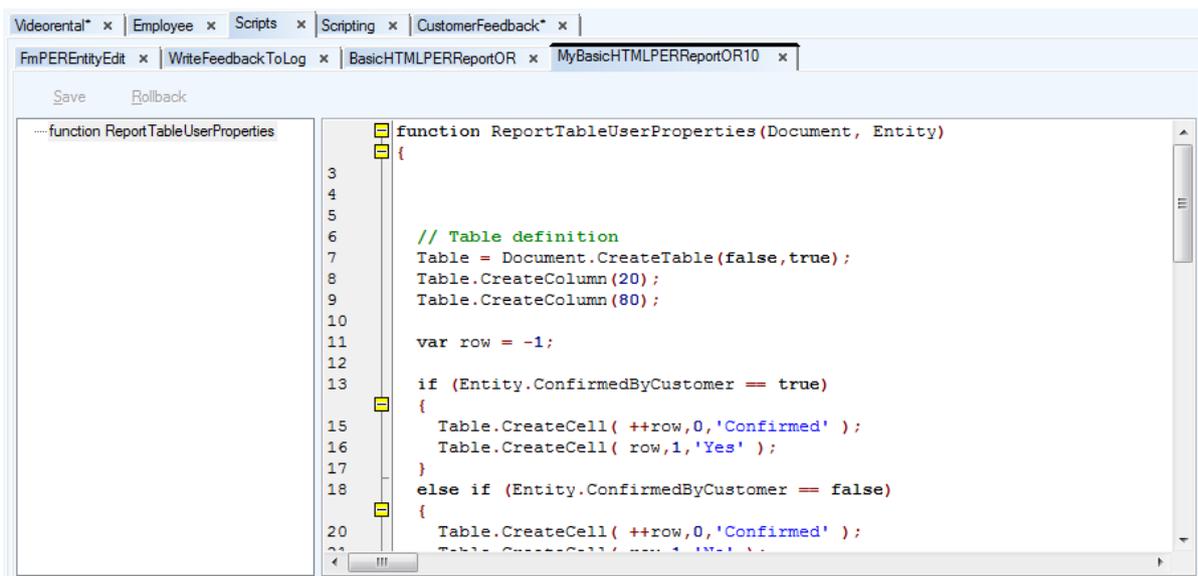


Return back to the **General** tab. Click **Reload**. Script name and method name will appear there. Add a prefix **My** to it (this will be changed in future, no manual modification will be required).



Click **Edit Script**.

Click **OK** and define code for the **ReportTableUserProperties** method that extends the existing method of the same name.



## Code:

```
function ReportTableUserProperties(Document, Entity)
{
    // Table definition
    Table = Document.CreateTable(false,true);
    Table.CreateColumn(20);
    Table.CreateColumn(80);
    var row = -1;

    if (Entity.ConfirmedByCustomer == true)
    {
        Table.CreateCell( ++row,0,'Confirmed' );
        Table.CreateCell( row,1,'Yes' );
    }
    else if (Entity.ConfirmedByCustomer == false)
    {
        Table.CreateCell( ++row,0,'Confirmed' );
        Table.CreateCell( row,1,'No' );
    }
    else
    {
        Table.CreateCell( ++row,0,'Confirmed' );
        Table.CreateCell( row,1,'Undefined' );
    }

    if (Entity.NotesFromCustomer.length > 0)
    {
        Table.CreateCell( ++row,0,'Notes from Customer' );
        Table.CreateCell( row,1, Entity.NotesFromCustomer );
    }
}
```

```

if (row > -1)
{
    Document.WriteStyled( 'CAPTION2', 'Customer Feedback' );
    Table.Draw();
    Table.Close();
}

Instance.ReportTableUserProperties(Document, Entity);

};

```

## Explanation

Table.**CreateColumn(20)** - the CreateColumn function belongs to the HTMLReportTable class. All functions related to the Table object can be found in the Toad Data Modeler Reference.

Entity.**ConfirmedByCustomer** - represents the variable we added earlier to the **CustomerFeedback** metamodel.

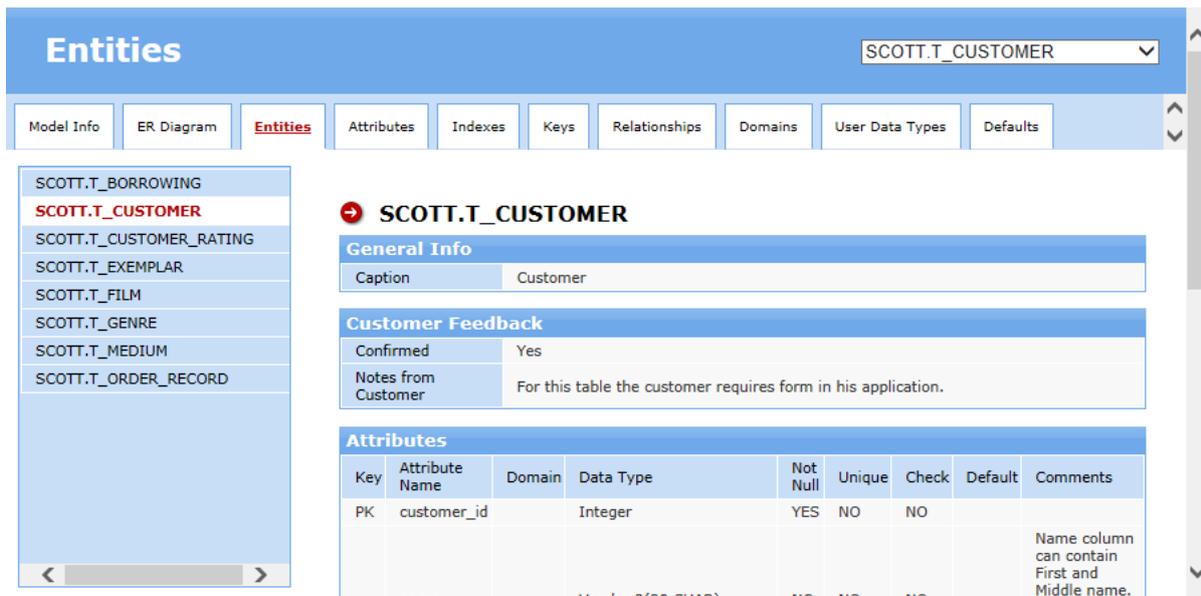
Entity.NotesFromCustomer.length - standard JavaScript function that returns number of characters of the NotesFromCustomer string.

Document.**WriteStyled** - represents function that belongs to the HTMLDocument class.

**Instance.ReportTableUserProperties(Document, Entity);**

- **Instance** - using the Instance keyword, we can call existing function we extended. We could copy and paste the content of the **ReportTableUserProperties** function defined in the **BasicHTMLPERReportOR** script. However, if a change was made to the script later, we would have to update our script too, which would be difficult to maintain. That's why it's better to write code that will extend the existing functionality only, and call the rest from existing script via the **Instance** keyword.
- **ReportTableUserProperties** - represents existing function we call.

When you generate HTML reports now, you will see the following output. New section **Customer Feedback** is generated on top, followed by the **Table Properties** part, as originally defined in the ReportTableUserProperties function in script BasicHTMLPERReportOR.



## Editable Forms and Frames

Toad Data Modeler allows you to edit some forms and frames for more comfortable work.

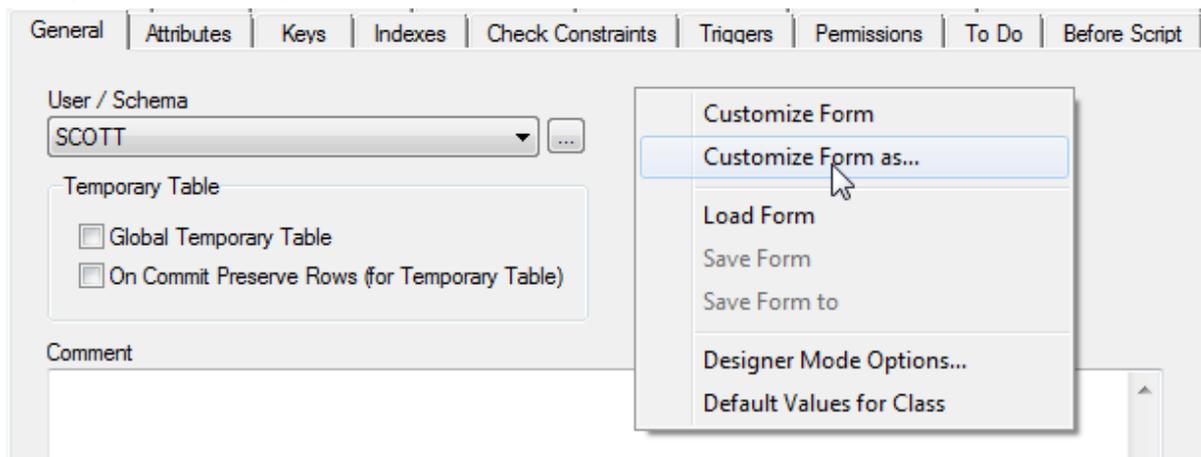
### What Is Editable Form?

It is a form of which appearance you can edit to meet your needs and requirements. You can add, change components and delete user components (those you created).

### How Do I know the Form is Editable?

Right-click the form to see the pop-up menu. If there's the **Customize Form (as)** option, the form is editable.

**Example:** The **Entity Properties** form



## What Is Editable Frame?

Properties of editable frames are similar to properties of editable forms, however frame represents only a part of a form, not complete form. E.g. the **Options** form has editable frames **Application**, **Graphics** etc.

## What Is Editable Form for?

To customize appearance and functionality of forms to suit your needs, e.g. you want to have different options for different database types.

If you model your own property in metamodel, you will probably want to visualize it somehow (write out its value to edit etc.) Therefore it's necessary to edit appropriate form and insert to it a new component to which you will assign appropriate property then. Also, to inserted components, you can add values by scripting.

### To edit a form

Right-click the form and select **Customize Form** or **CustomizeForm as**.

Customize Form	The changes will be saved to your user package <i>My Package</i> .
Customize Form as	The changes will be saved to another package than <i>My Package</i> . This option is disabled by default. To enable it, select <b>Settings   Options   Expert Mode</b> and uncheck the <b>Save the definitions to the My Package</b> checkbox.

Forms can be edited via the following tools :

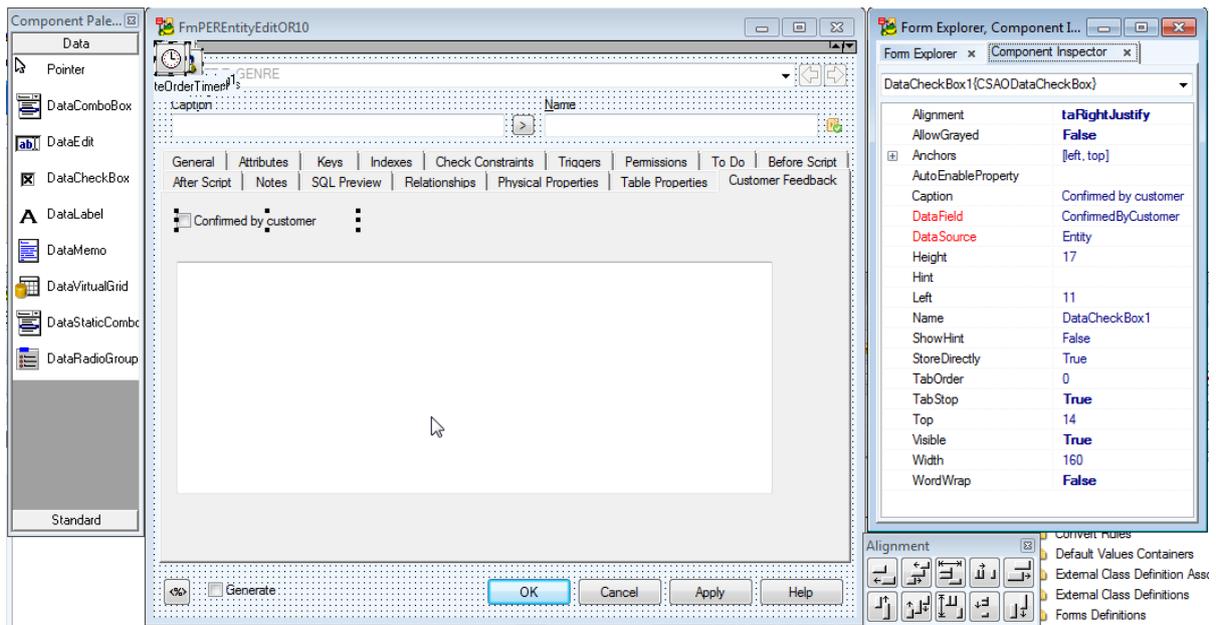
- Component Inspector - list of properties of selected component in the form. Here, you can edit properties
- Component Palette - list of components that you can add to a form
- Form Explorer - tree structure of the edited form
- Alignment - alignment options

Forms of these three tools can be docked in Toad Data Modeler environment at your convenience.

If you want to hide/show any of these tools permanently, you can clear/select appropriate option in:

- Right-click the form and select **Designer Mode Options**.
- From the **Settings** menu, select **Options | Editable Forms**.

**Example:** The **Entity Properties** form that is being edited now.



### To finish editing a form

Close the form that you have just edited (click **x** in the right-hand corner, e.g. in the **Entity Properties** form). After you close the edited form, you need to save the changes to package.

**i** Note: You will not finish editing the form by closing any of the tool forms (Inspector, Explorer or Palette Component). If you close any of them, the **Show** option for the tool (Show Inspector, Show Explorer...) will be automatically cleared. If you edit a form next time, the form of this tool will not display. To display it again, you need to enable the **Show** option either in the **Designer Mode Options** or in the **Settings menu | Options | Editable Forms**.

### To save the changes

1. Right-click the form and select **Save Form** or **Save Form to**.

The changes will be saved to appropriate packages, however to preserve them even after you close the application, you have to save also the packages.

2. Save the packages in Package Explorer.

Changes will be saved either to actual package, or user package (according to the settings in the **Settings menu | Options | Paths**).

If you want to save the changes to another package, you have to start the edit process with option **Edit Form as**, then select the package and finally click **Save Form to** to save the changes.

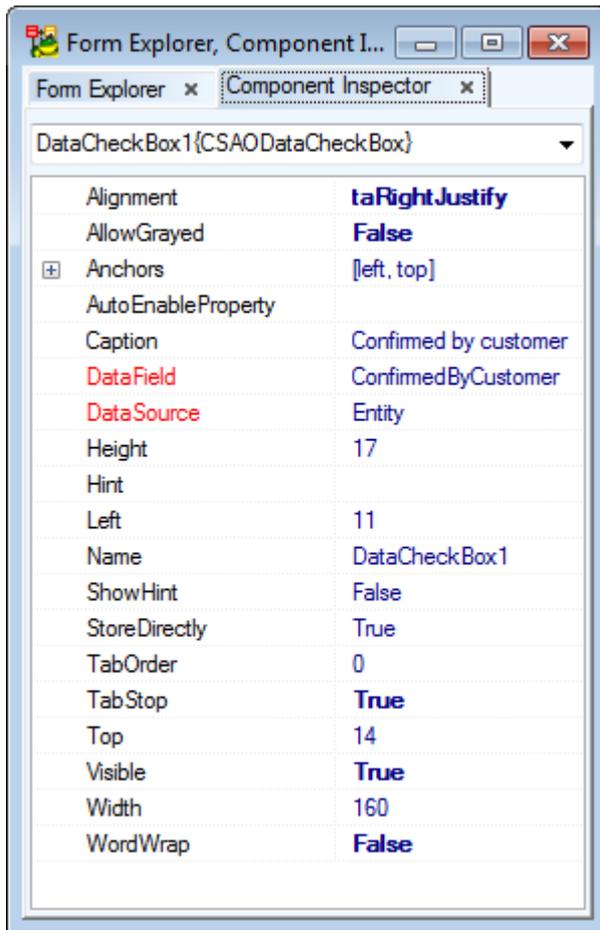
## Editable Forms Right-Click Options

**Right-click the form to see the following options:**

Option	Description
Customize Form	Edits a form. All changes will be stored to <i>My Package</i> user package (file <i>My Package.txg</i> ).
Customize Form as	Edits a form in a state according to the selected package. (Only the components that are saved in the selected package and its child packages will be available.) This option is available only if checkbox <b>Save the Definitions to the 'My Package'</b> is clear. - See the checkbox in the <b>Settings</b> menu   <b>Options</b>   <b>Expert Mode</b> .
Load Form	Reloads a form.
Save Form	Saves changes in the default package (see the <b>Settings</b> menu).
Save Form to	Saves changes to previously selected package (see <b>Edit Form as</b> ).
Designer Mode Options	Here, you can hide/show the tools for next form edit.
Default Values of Object	Opens the dialog where you can define or edit default values of appropriate form/dialog/frame.

## Component Inspector

**Component Inspector** is a list of properties of the selected component in the form. Here, you can edit properties.



Option	Description
Top combo-box	Select a component to display in this dialog.
Left Column	Displays property names.
Right Column	Displays values. Most of the values can be changed in inplace editors.

**i** Note: If you select components directly in the form you edit, you can select more components using the SHIFT key. Then, only common properties will be displayed in Component Inspector.

Right-click the **Component Inspector** form and select **Properties**. Here you can edit colors of specific parts of the dialog, and select other options.

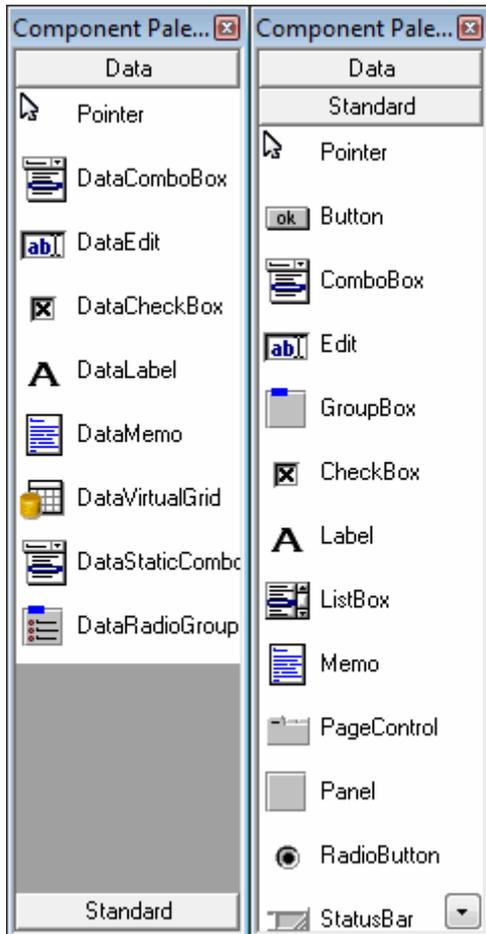
## Component Palette

**Component Palette** is a list of components that you can add to a form. It is divided into two parts:

- **Data** - here you can find components that can follow up with already existing properties
- **Standard** - other components that can be used for change of appearance or better classification of components

### **To insert a component to a form**

1. Select a component from the **Component Palette**.
2. Click to a place where you want to insert it.



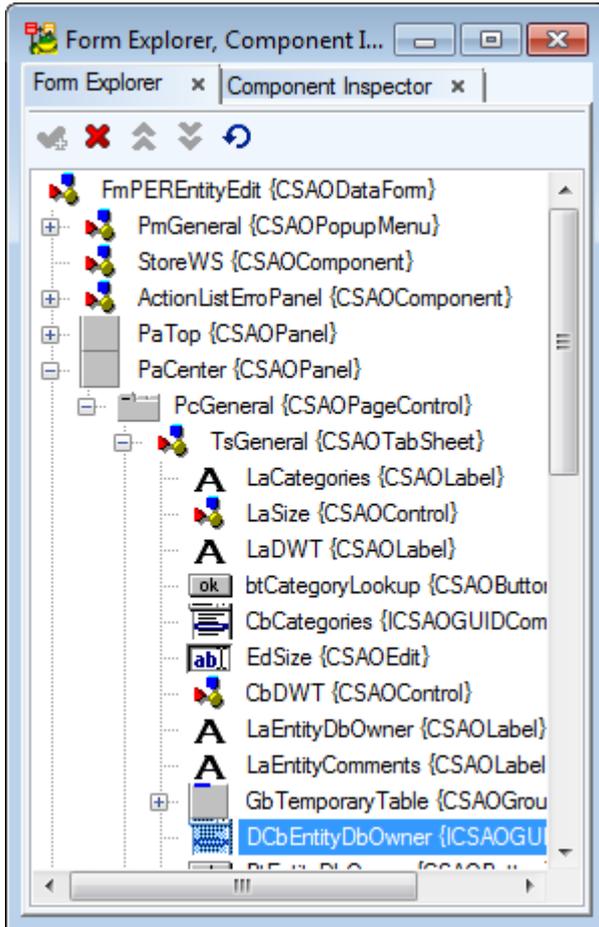
### **To display components with small buttons**

Right-click the **Component Palette** form and select **Properties | Small Buttons**.

## **Form Explorer**

**Form Explorer** displays tree structure of components in a form.

In **Form Explorer**, you can move components, select them for edit in **Component Inspector** (via mouse), and delete them (Delete key).



Right-click the **Explorer** form and select **Properties**.

## Macros

Toad Data Modeler supports macros. You can use sample macros available in the **Macros** menu or create your own macros. You can create a macro in Package Explorer or Script Explorer and modify its properties to display the macro either in the main menu or pop-up menu (of specific object or on the Workspace etc.). Visual components for creation of macros are also available (User Forms).

Macros menu contains:

- All - The selected macro will be applied either on all objects on all Workspaces or all objects of the active Workspace.
- Selected Objects - The selected macro will be applied only on the selected objects on the currently active Workspace.
- [Productivity Pack](#)
- [Rename Objects Pack](#)

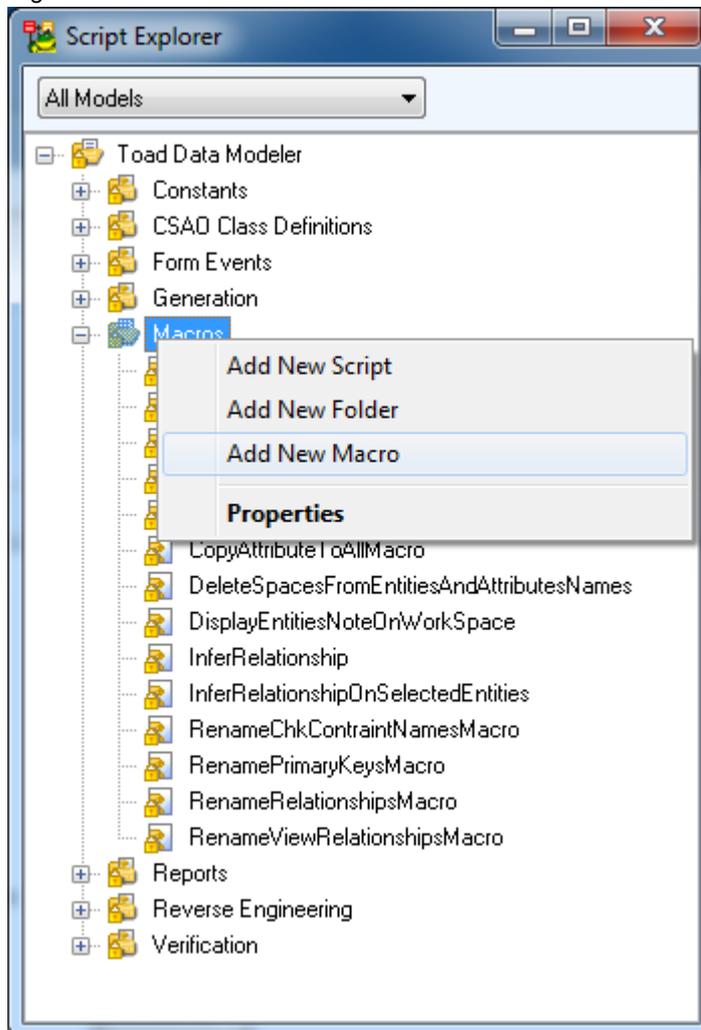
---

Scenario

You want to create a macro *Add Attribute to PK* and add this macro to pop-up menu of attributes in Model Explorer.

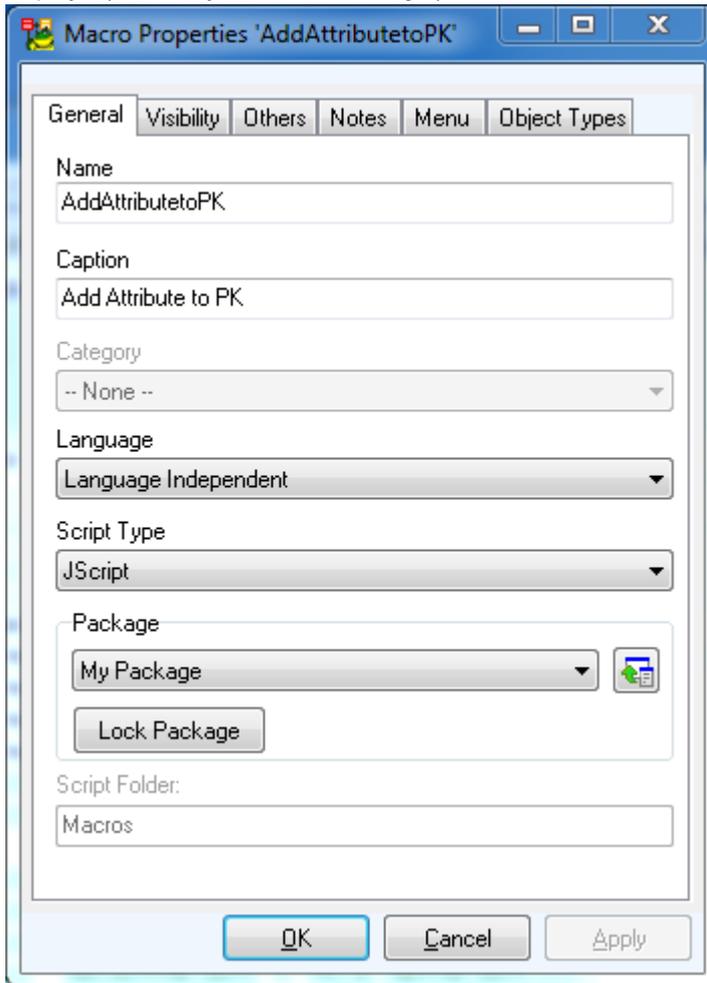
### To create your own macro

1. Open **Script Explorer** (**Tools** menu, Expert mode must be enabled).
2. Right-click the **Macros** item and select **Add New Macro**.

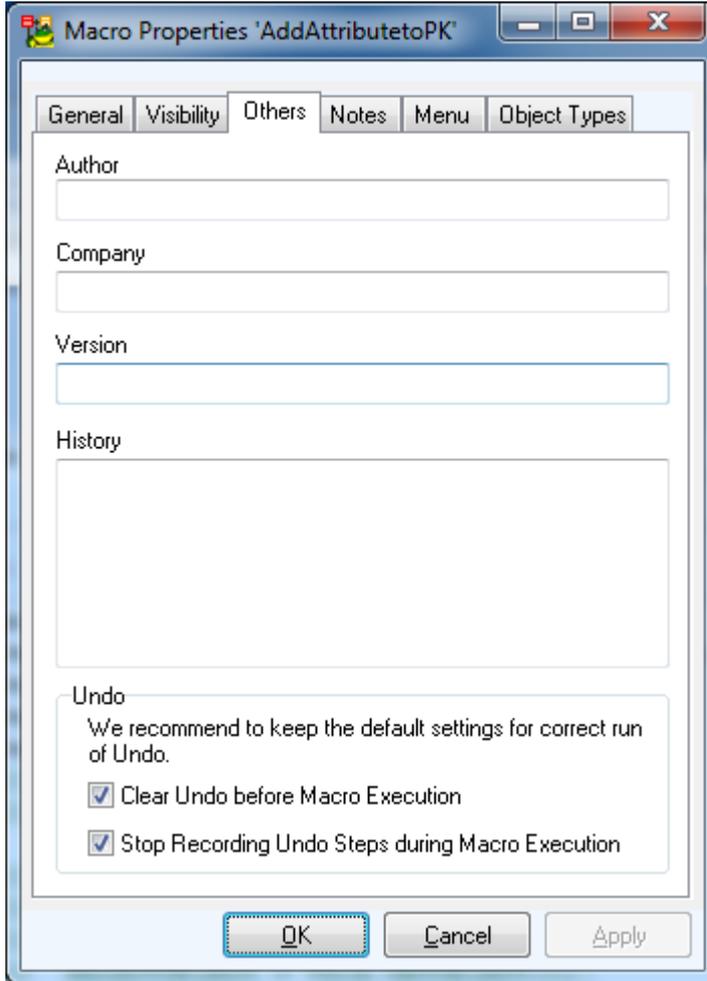


3. Right-click the newly created macro and select **Properties**.

4. Define properties of the new macro. Remember to define its caption (macro name that will be displayed), visibility on tab **Visibility** (if for all databases etc.).

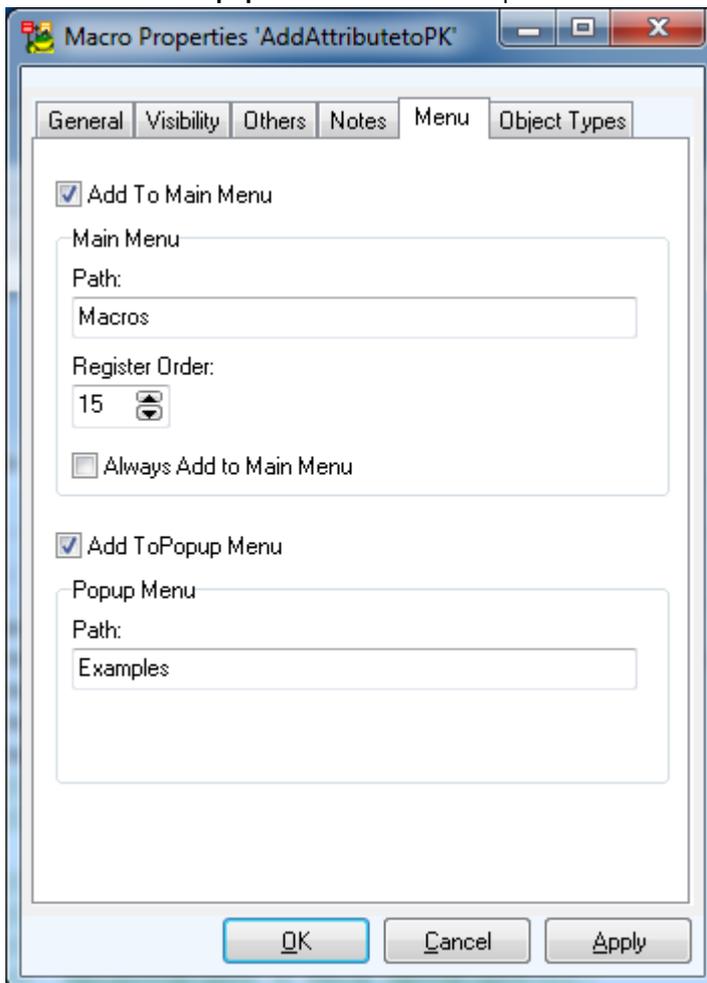


5. On tab **Others**, you can define Undo options. It is recommended to keep the default settings.



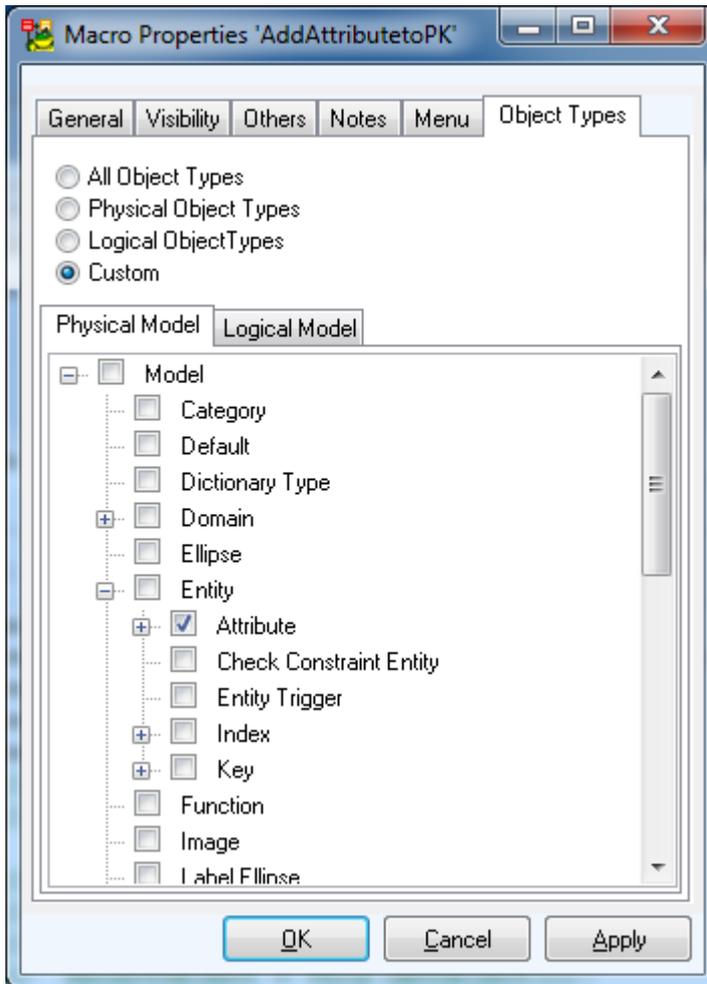
6. On tab **Menu**, you can define where you want to display the macro - in Main menu, pop-up menu or both.

Select **Add to Popup Menu** and write the path name.



7. On tab **Object Types**, select object types for which the macro will be available in their pop-up menu.

Select **Attribute** and confirm **OK**.

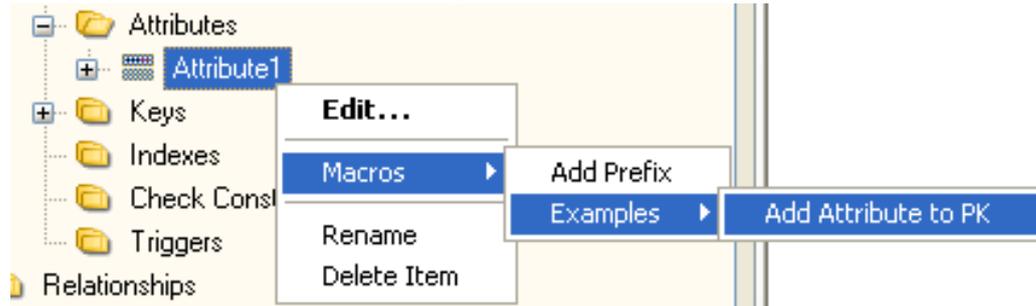


8. Double-click the new macro to open the **Script Editor**. Modify the default code at your convenience.

```
function IsAttrInKey(Attr, Key){  
  
    var i, item;  
    for(i=0; i<Key.KeyItems.Count;i++){  
        item = Key.KeyItems.GetObject(i);  
        if (item.Attribute.Id==Attr.Id)  
        {  
            return true;  
        }  
    }  
  
    return false;  
}  
  
function Main(){  
    var App = System.GetInterface("Application");  
    var Model = App.ActiveModel;  
    var WS = App.ActiveWorkSpace;  
    var Log = System.CreateObject("Log");  
  
    var i, SelectObject;  
    var PK;  
    for(i=0; i<This.Count;i++)  
    {  
        SelectObject = This.GetObject(i);  
        if (SelectObject.ObjectType == 2003)  
        {  
  
            PK = SelectObject.Owner.PK;  
            if (!IsAttrInKey(SelectObject, PK)){  
                {  
                    PK.Lock();  
                    SelectObject.Owner.Lock();  
                    PK.AddAttribute(SelectObject);  
                    PK.CommitChanges();  
                    SelectObject.Owner.Unlock();//Refreh Entity  
                    PK.Unlock();//Refresh Key  
                }  
            }  
        }  
    }  
}
```

9. Confirm **Commit and Save**.

10. The **Macros** will be available accordingly.



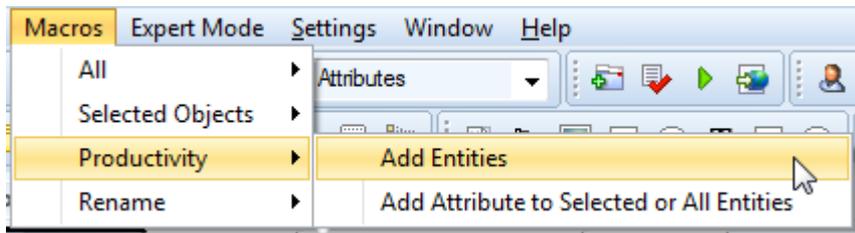
## Productivity Pack

Productivity pack contains three macros.

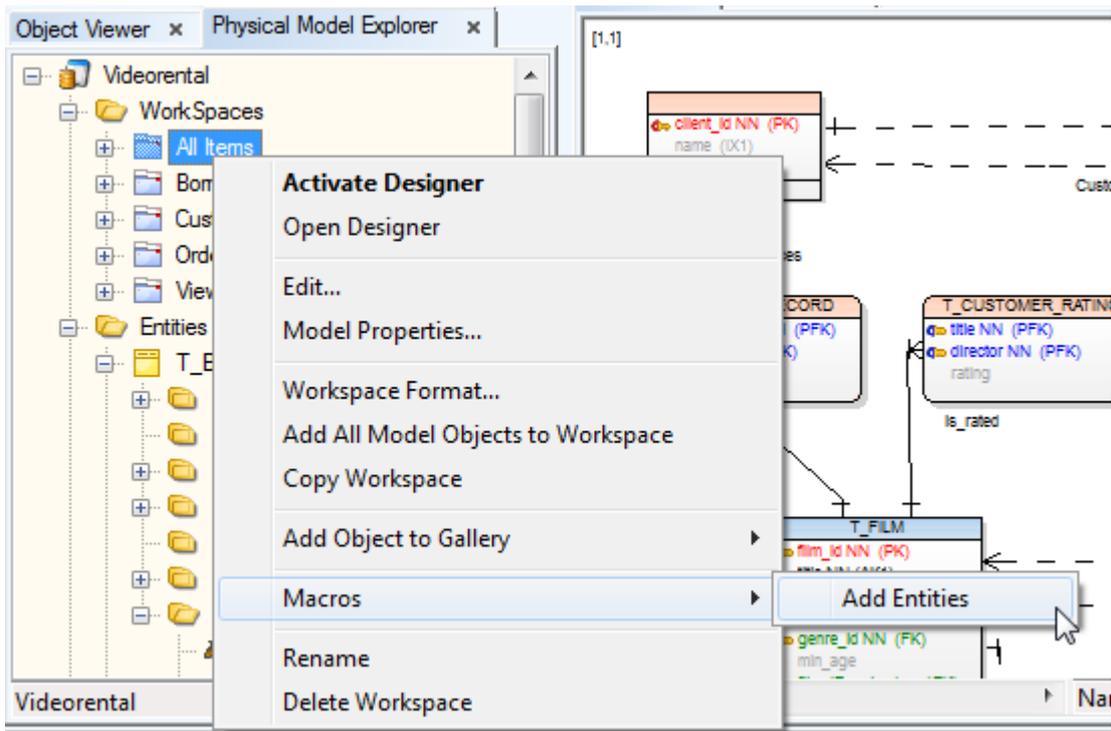
- Add Entities
- Add Attribute to Selected or All Entities
- Copy Attribute to All Entities

### Add Entities

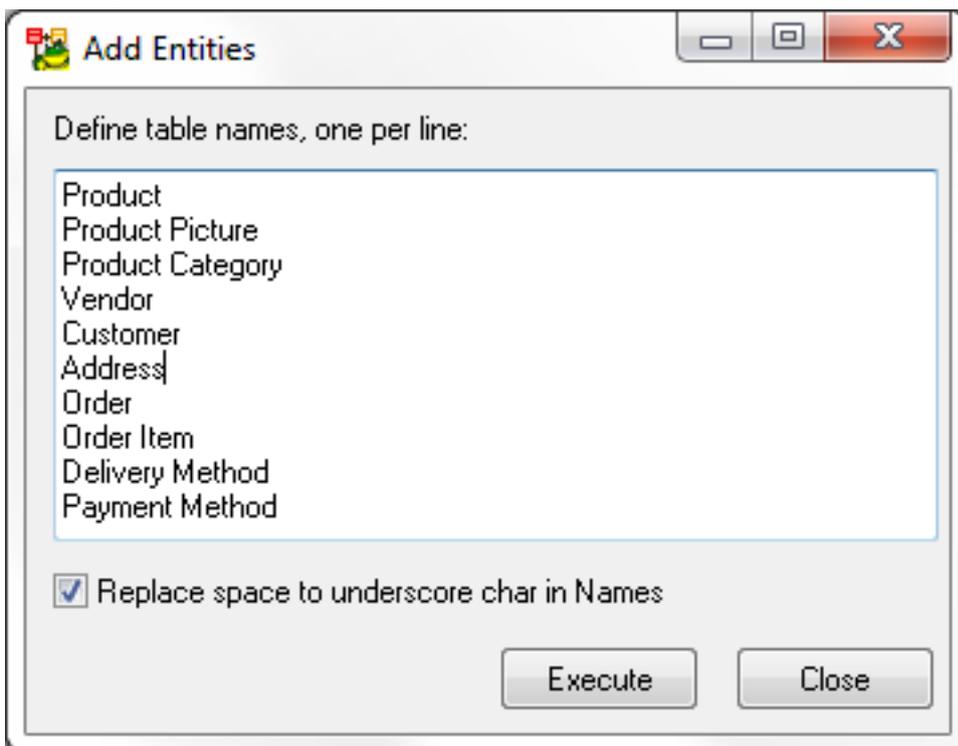
Select **Macros | Productivity | Add Entities** to run the macro.



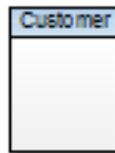
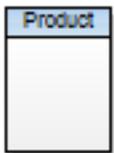
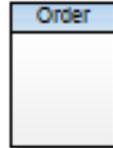
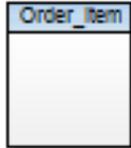
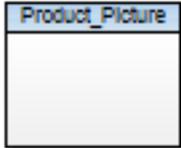
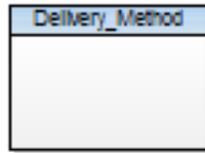
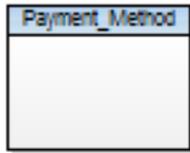
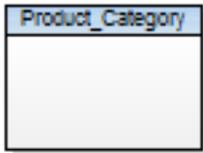
The same macro can be executed from Workspace.



New dialog opens. Specify entity captions, one per line.



Click **Execute** to create the new tables.



The tables will be added to your model.

In case your workspace option **Autocomplete** is activated the entities will appear also on our workspace.

## Add Attribute to Selected or All Entities

Select **Macros | Productivity | Add Attribute to Selected or All Entities** to run the macro.

New dialog opens. Specify caption, name, data type and other options.

Tip: If you want to use table caption/name as prefix, type a "space" as the first character in Attribute Caption field and an underscore as the first character to Attribute Name field.

**Add Attribute**

Attribute Caption: ID      Attribute Name: \_ID

Data type: Integer

Add to Key (Attribute will be added to first found key)

Use Table Name as Prefix

Not Null

Execute      Close

Note that if you select the **Add to Key** checkbox, the Not Null checkbox disappears.

**Add Attribute**

Attribute Caption: ID      Attribute Name: \_id

Data type: Integer

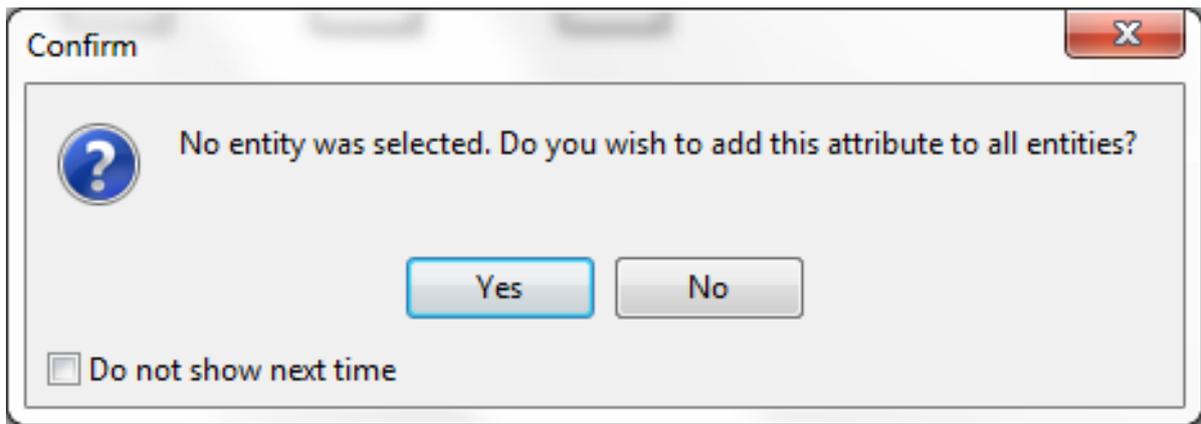
Add to Key (Attribute will be added to first found key)

Use Table Name as Prefix

Execute      Close

In case you selected an entity, the attribute will be added to the selected entity. Otherwise the following dialog opens.

Click **Yes** to add attribute to all entities.



Result:

Product_Category	
Product_Category_id	Integer (PK)

Payment_Method	
Payment_Method_id	Integer (PK)

Product_Picture	
Product_Picture_id	Integer (PK)

Order_Item	
Order_Item_id	Integer (PK)

Customer	
Customer_id	Integer (PK)

Address	
Address_id	Integer (PK)

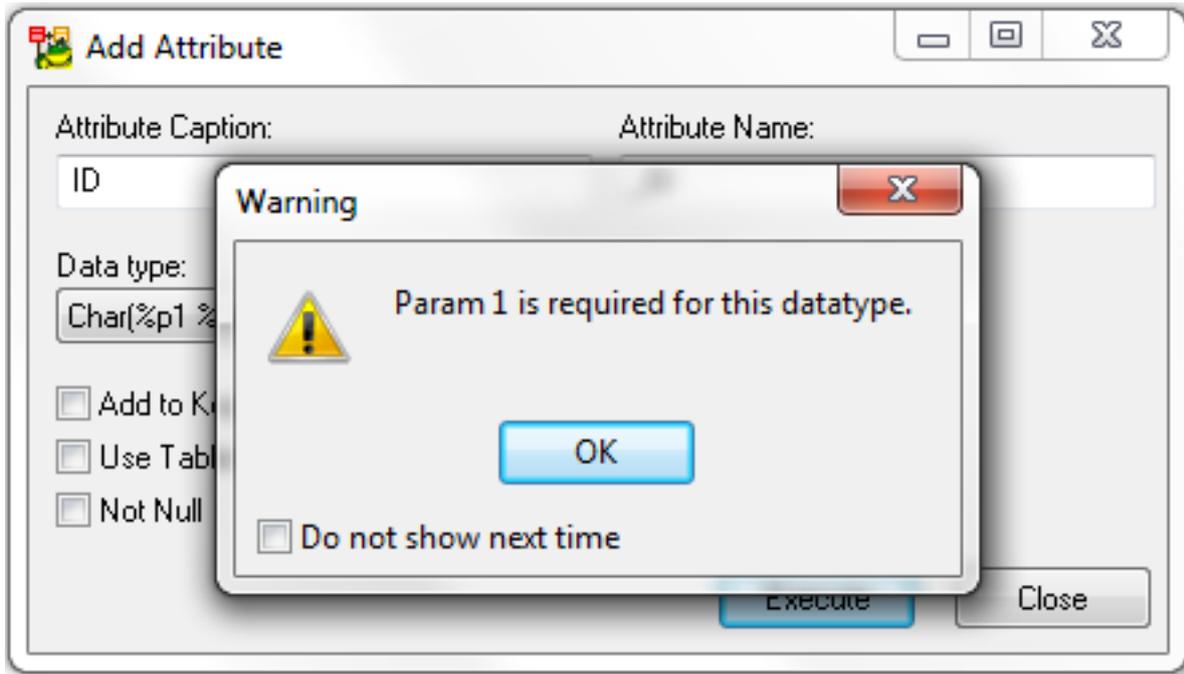
Delivery_Method	
Delivery_Method_id	Integer (PK)

Product	
Product_id	Integer (PK)

Vendor	
Vendor_id	Integer (PK)

Order	
Order_id	Integer (PK)

Verification: There is some basic verification of the required items.

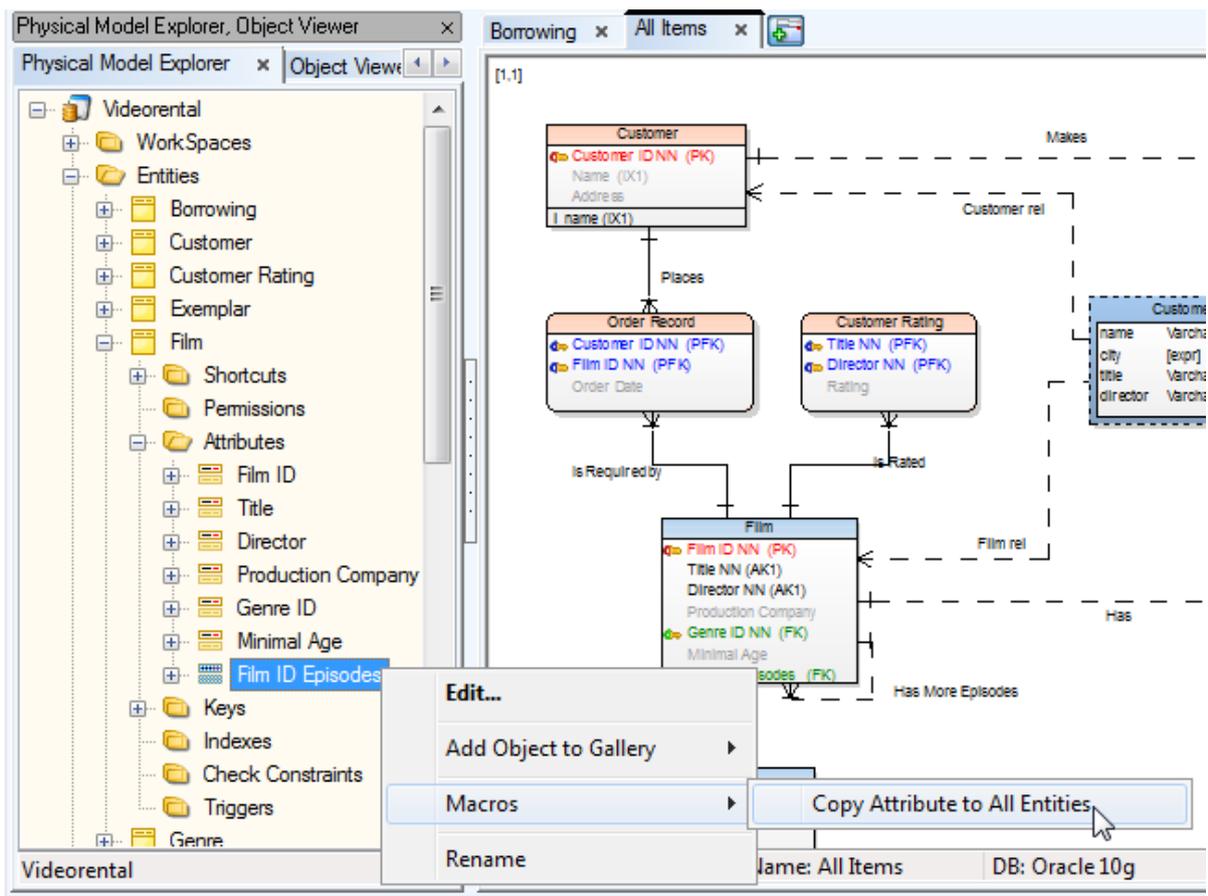


Notes:

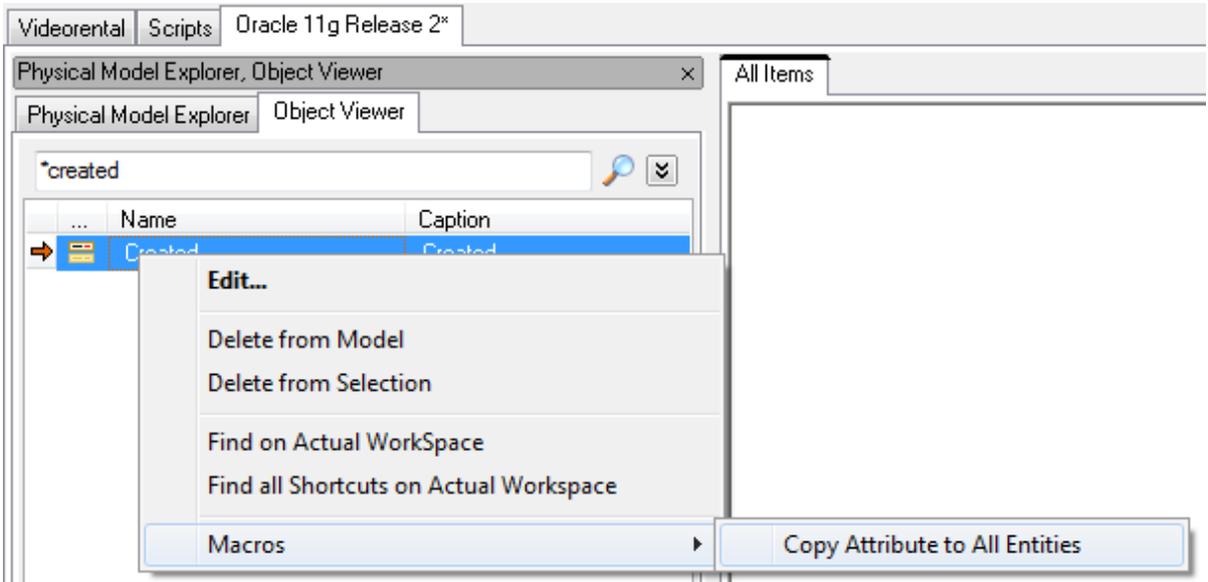
- If you run the macro on entities with identically named attributes, the attribute will not be added to the entity more than once.
- If you specify Varchar(%p1 %p2) as the data type, you need to know whether the second parameter is BYTE or CHAR. There is no verification for parameter values.

## Copy Attribute to All Entities

Select an attribute in physical Model Explorer, right-click and select **Macros | Copy Attribute to All Entities**.



It is possible to run the macro also from **Object Viewer**.



Result:

Product_Category		
Product_Category_id	Integer	(PK)
Created	Timestamp(6)	

Payment_Method		
Payment_Method_id	Integer	(PK)
Created	Timestamp(6)	

Product_Picture		
Product_Picture_id	Integer	(PK)
Created	Timestamp(6)	

Order_Item		
Order_Item_id	Integer	(PK)
Created	Timestamp(6)	

Customer		
Customer_id	Integer	(PK)
Created	Timestamp(6)	

Address		
Address_id	Integer	(PK)
Created	Timestamp(6)	

Delivery_Method		
Delivery_Method_id	Integer	(PK)
Created	Timestamp(6)	

Product		
Product_id	Integer	(PK)
Created	Timestamp(6)	

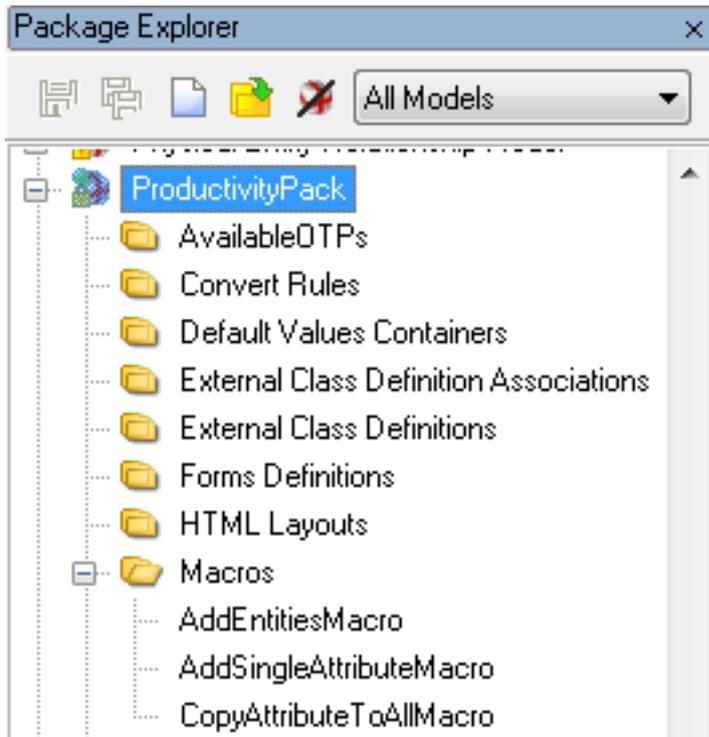
Vendor		
Vendor_id	Integer	(PK)
Created	Timestamp(6)	

Order		
Order_id	Integer	(PK)
Created	Timestamp(6)	

**i** Note: The attribute will be added only to the entities which don't have an identically named attribute.

## Sources

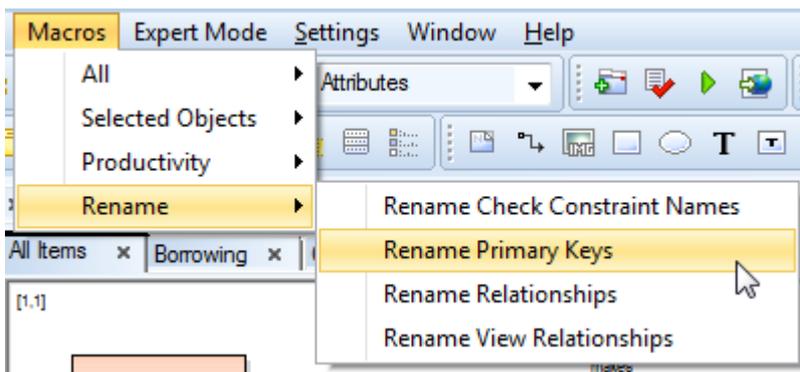
You can find them in **Package Explorer** in section Productivity Pack | Macros.  
Right-click any macro and select **Edit Source Code** to see JavaScript code.



## Rename Objects Pack

Rename Objects Pack contains the following macros:

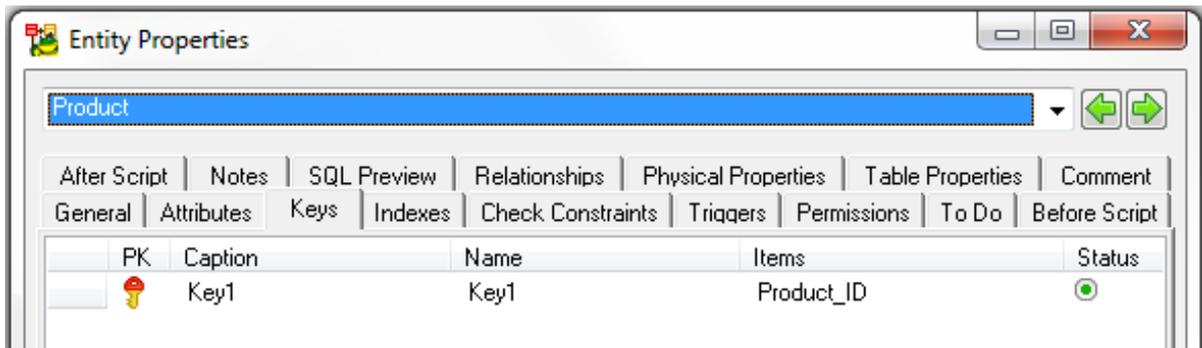
- Rename Primary Keys
- Rename Relationships
- Rename Check Constraint Names
- Rename View Relationships



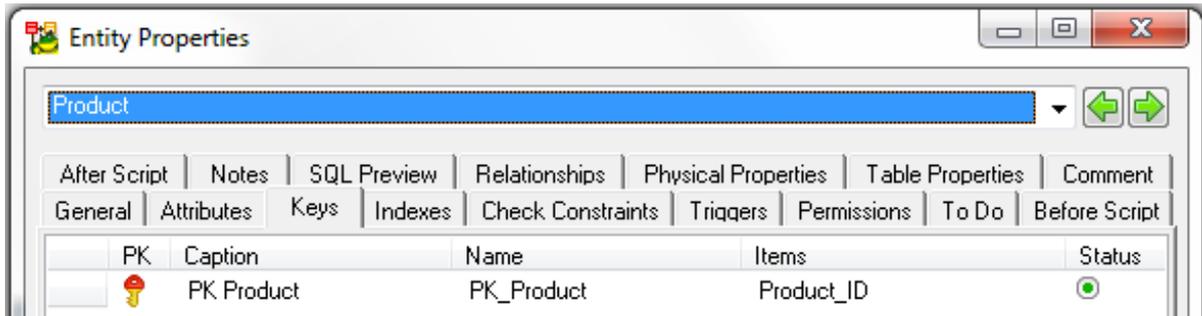
### Rename Primary Keys

This macro renames primary keys to **PK\_<tablename>**.

When you create a new entity, its primary key is named Key1.



When you execute the macro, the key name and caption changes.



Notes:

- If you select an entity, the macro will modify the primary key name and caption of the selected entity only.
- If you need to change the prefix or if you want to use suffix, create a new macro and modify its JavaScript code accordingly.

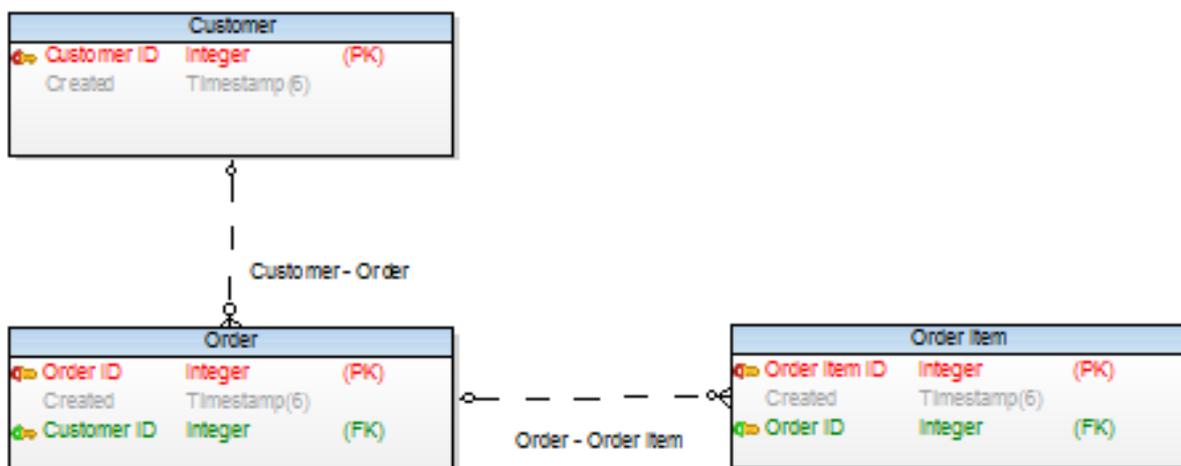
## Rename Relationships

This macro renames relationship captions and names and modifies them to:

New caption: <parenttablecaption> - <childtablecaption>

New name: <parenttablename>\_<childtablename>

By default, relationships are named as Relationship1, Relationship2 etc. When you execute the macro, the following result is achieved:



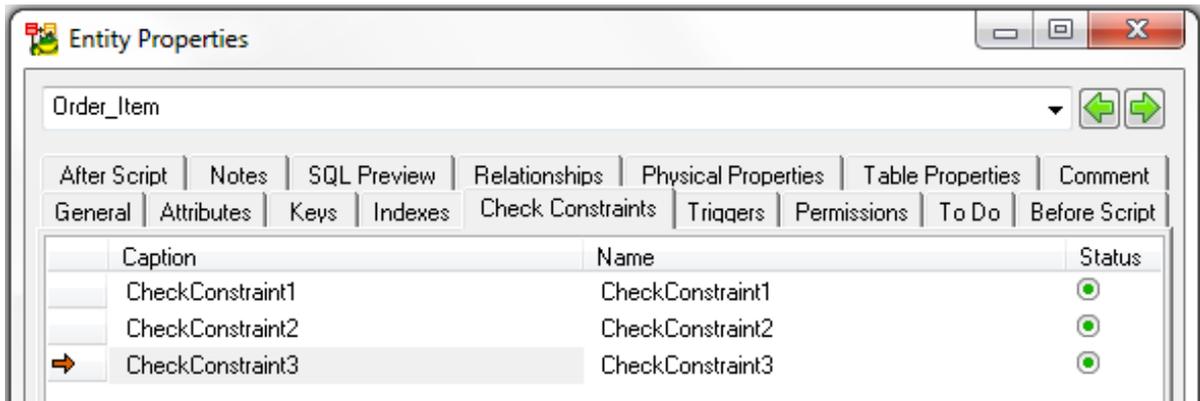
Notes:

- If you select a relationship, the macro will modify only the selected relationship, otherwise you will be asked if you want to run the macro on all relationships.
- If multiple relationships exist between two entities, a random number will be added at the end of the relationship name.

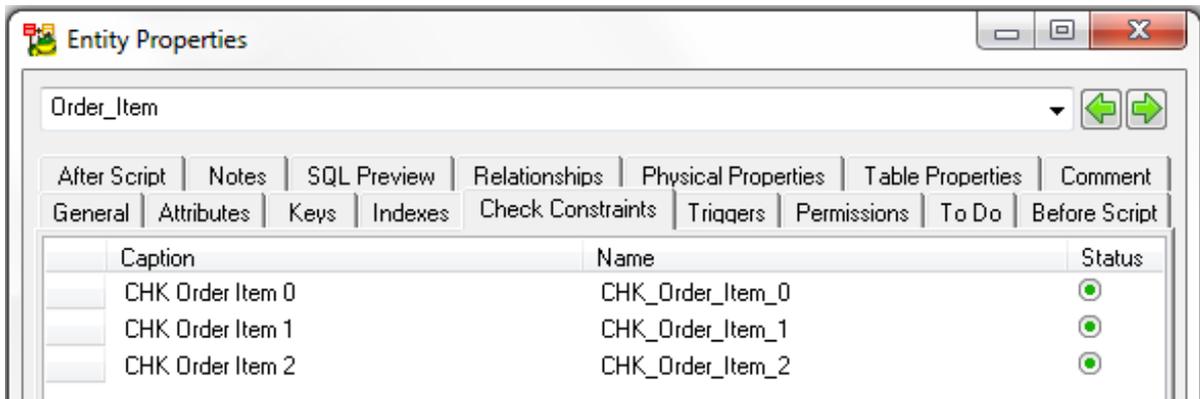
## Rename Check Constraint Names

This macro renames Table and Column check constraint names and captions.

By default, Toad Data Modeler names check constraints as CheckConstraint1,2,3 etc.



After you execute the macro, check constraints will be renamed to CHK\_<tablename>\_<index> and CHK\_<columnname>\_<index>.



Notes:

- If you select an entity, the macro will modify check constraint names and captions of the entity and its attributes.
- If you need to change the prefix or if you want to use suffix, create a new macro and modify its JavaScript code accordingly.

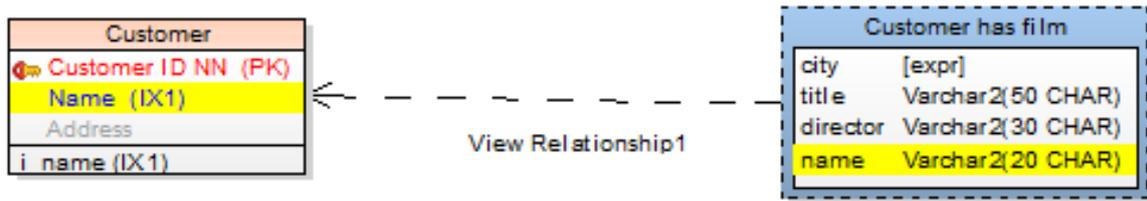
## Rename View Relationships

This macro works similarly to the Rename Relationships macro.

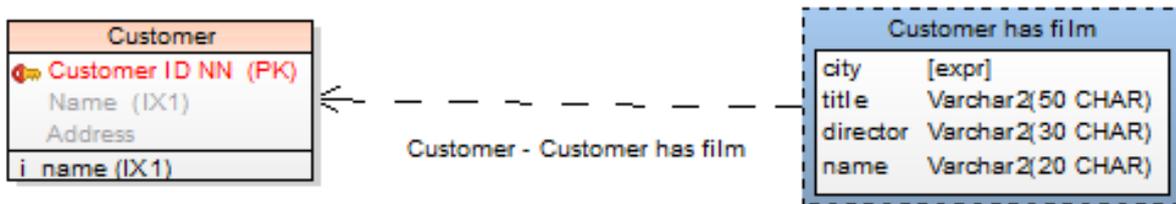
New caption: <object1caption> - <object2caption>

New name: <object1name>\_<object2name>

The macro renames existing View relationships.



Result:



**i** Note:

- If you select a View relationship, the macro will modify only the selected View relationship, otherwise you will be asked if you want to run the macro on all View relationships.

## Macros and User Forms

Toad Data Modeler allows you to create a macro in Package Explorer or Script Explorer and modify its properties to display the macro either in main menu or pop-up menu (of specific object or on the Workspace etc.).

Older versions allowed you to define such macros via a script written in Script Editor. To execute the script directly, you simply selected the macro in the appropriate menu.

Toad Data Modeler version 3.5 is bringing some improvements for using macros - visual components for macros (User Forms). So, now when you select a macro, a user form can display.

### User Forms - Brief Information:

- You can create and use user forms to interact with Toad Data Modeler during script and macro execution. You can enter input parameters or see some output information.
- Function Main only creates and displays the user form. Other functionalities must be implemented/added via form events or its controls. So, a form is not a dialog.

### Create a Form

To create a form, use the object *System* that is registered in every script.

The method you need is called *CreateForm* and has four optional parameters:

**Example:**

```
var form = System.CreateForm('FormName', 'Form Caption', 200, 150);
```

1. First Parameter – Name of form (it mustn't contain spaces and other invalid/not permitted characters).
2. Second Parameter – Caption that will be displayed in the heading of the form.

3. Third Parameter – Width of the form.
4. Fourth Parameter – Height of the form.

## Functions of Form

### AddControl(ControlName: widestring, ControlType: Integer): IDispatch;

- *ControlName* – Name under which the control is accessible.
- *ControlType* – Number of control type that should be created.

See the following table:

- 1	- Edit Box
- 2	- Check Box
- 3	- Memo
- 4	- Panel
- 5	- Label
- 6	- Group Box
- 7	- Radio Button
- 8	- Combo Box
- 9	- List Box
- 10	- Button

This function adds control on the form.

### ShowModal()

This function displays the form.

## Procedures of Form

### AddUserVariable(AName: widestring, DefaultValue)

- *AName* – Name under which a variable is accessible in events of forms.
- *DefaultValue* – Default value. It can be of types integer, widestring or boolean.

This procedure adds a variable on the form. The variable is then accessible in events via calling the *Instance.VariableName*. The variable is accessible across events. If you change a content of the variable in one event, the changed status will be accessible in another event.

### RegisterObject(AName: widestring, AObject: IDispatch)

- *AName* – Name of object via which it will be accessible in events.
- *Aobject* – Object that is registered.

Use this procedure to register objects in events.

## Properties of Form

**Caption** – Heading of the form.

**CloseAfterExecute** – *True* – When you click **Execute**, the code will be executed and the form closes. *False* – The form will not close after execution. Default: *False*.

**ExecuteMethodName** – Name of method that should be executed when you press the **Execute** button.

**ExecuteScriptName** – Name of script for calling out the method when you click the **Execute** button.

**i** | Note: If you don't want to use the button **Execute**, do not set up the properties *ExecuteMethodName* and *ExecuteScriptName*. The button will not be visible on the form then.

## EVENTS

To assign events, assign the component of particular event to properties of names *NameEventScriptName*, *NameEventMethodName* with reference to particular service method.

### Example:

```
Button.OnClickScriptName = 'MyScript';
```

```
Button.OnClickMethodName = 'DoOnClick';
```

## CONTROL

Control is an ancestor from which all controls, including the form, inherit.

### Properties of Control

**Align** – Alignment of control. Possible values to use:

0 - No alignment

1 - Alignment - Top

2 - Alignment - Bottom

3 - Alignment - Left

4 - Alignment - Right

5 - Alignment – All client

**AnchorTop, AnchorBottom, AnchorLeft, AnchorRight** – Determines the position of control. Default place – top left-hand corner.

**Parent** – Control on which a control is placed. Default position of all controls is on the form and this property is not set up.

**Note:** Description of value *Align 0..5*:

*alNone* - The control remains where it was placed. This is the default value.

*alTop* - The control moves to the top of its parent and resizes to fill the width of its parent. The height of the control is not affected.

*alBottom* - The control moves to the bottom of its parent and resizes to fill the width of its parent. The height of the control is not affected.

*alLeft* - The control moves to the left side of its parent and resizes to fill the height of its parent. The width of the control is not affected.

*alRight* - The control moves to the right side of its parent and resizes to fill the height of its parent. The width of the control is not affected.

*alClient* - The control resizes to fill the client area of its parent. If another control already occupies part of the client area, the control resizes to fit within the remaining client area.

## Button

### Event

*OnClick* – Occurs when you click the button.

## Checkbox

### Event

*OnClick* – Occurs when the check in checkbox is changed.

## Combobox

### Event

*OnSelect* - Occurs when combo box is selected.

## Edit

### Event

*OnChangeText* – Occurs when text in edit box is changed.

## Memo

### Event

*OnChangeText* – Occurs when text in memo is changed.

## Radio Button

### Event

*OnClick* – Occurs when the button is selected.

For more properties, please read the Reference Guide (Expert Mode main menu, Expert Mode has to be enabled.). See objects: *UserButton*, *IUserCheckBox*, *IUserComboBox*, *UserControl*, *UserEdit*, *UserFormBasic*, *UserForm*, *UserGroupBox*, *IUserLabel*, *UserListBox*, *UserMemo*, *IUserPanel*, *UserRadioButton*, *UserStrings*.

## Macros and User Forms - Use Case

---

### Scenario

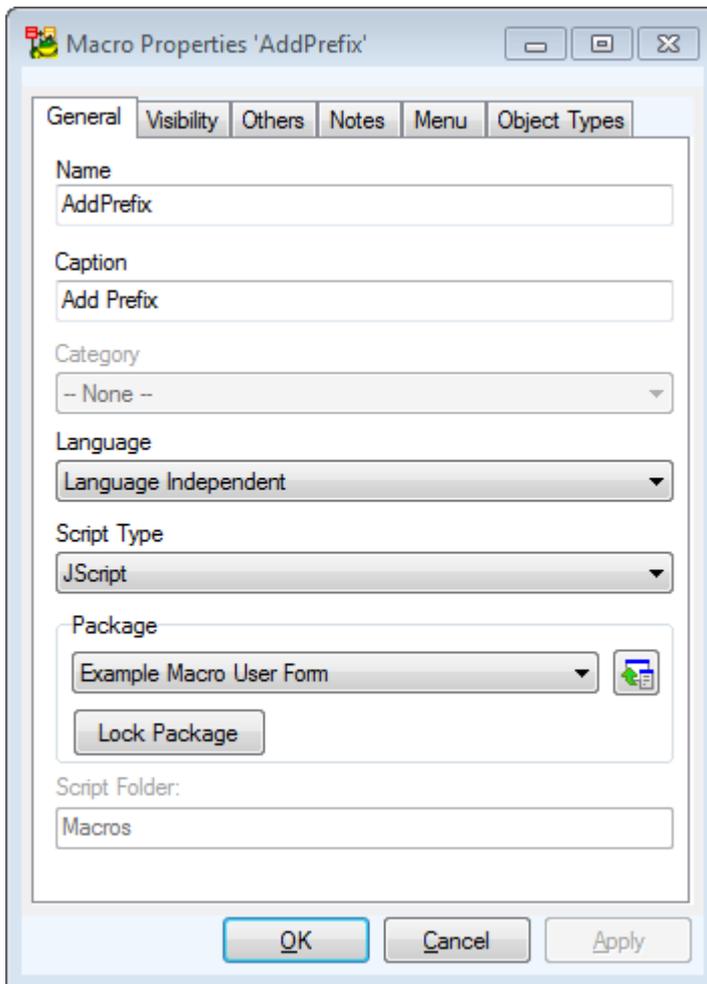
You want to create a macro that will add a prefix to all attributes in your model.

Solution: You will create a macro *Add Prefix*. The macro will be available via right-click menu on the Workspace. You want to create a user form where you will define the prefix and decide if you want to apply the change in Caption of attributes too.

---

1. Open Script Explorer.
2. Right-click the **Macros** item and select **Add New Macro**.
3. Right-click the new item and select **Properties**.
4. On tab **General**, define properties of the macro.

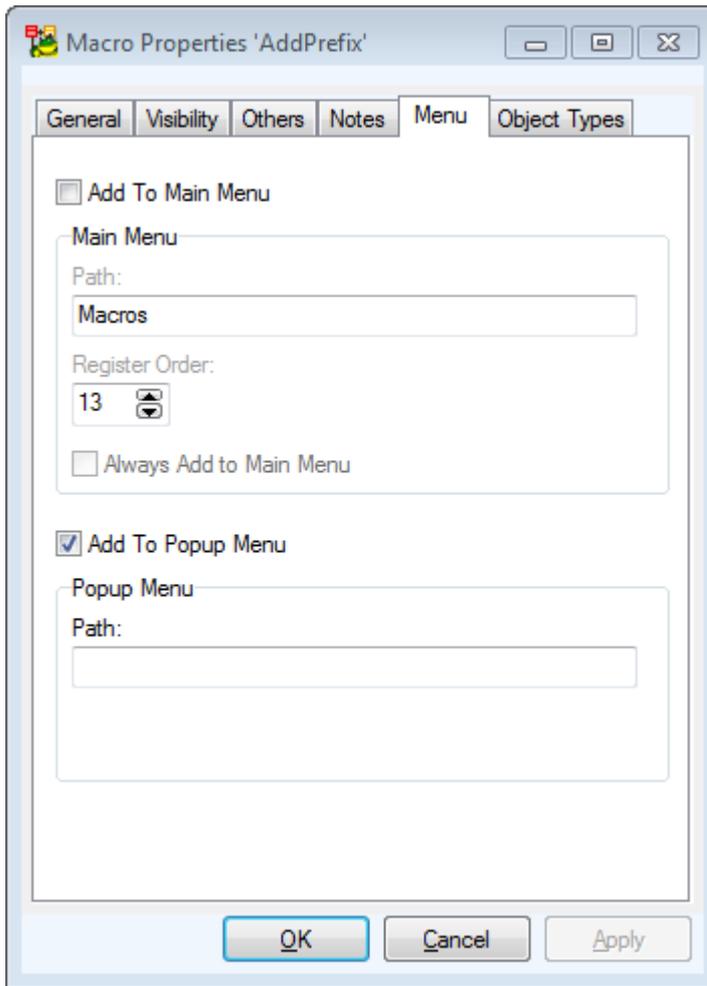
**i** Important: Name of macro mustn't contain spaces and other forbidden characters. The name must start with a character (not number). Then you can use characters, numbers or possibly '\_.'. The rules don't refer to caption. Caption can be any title you want.



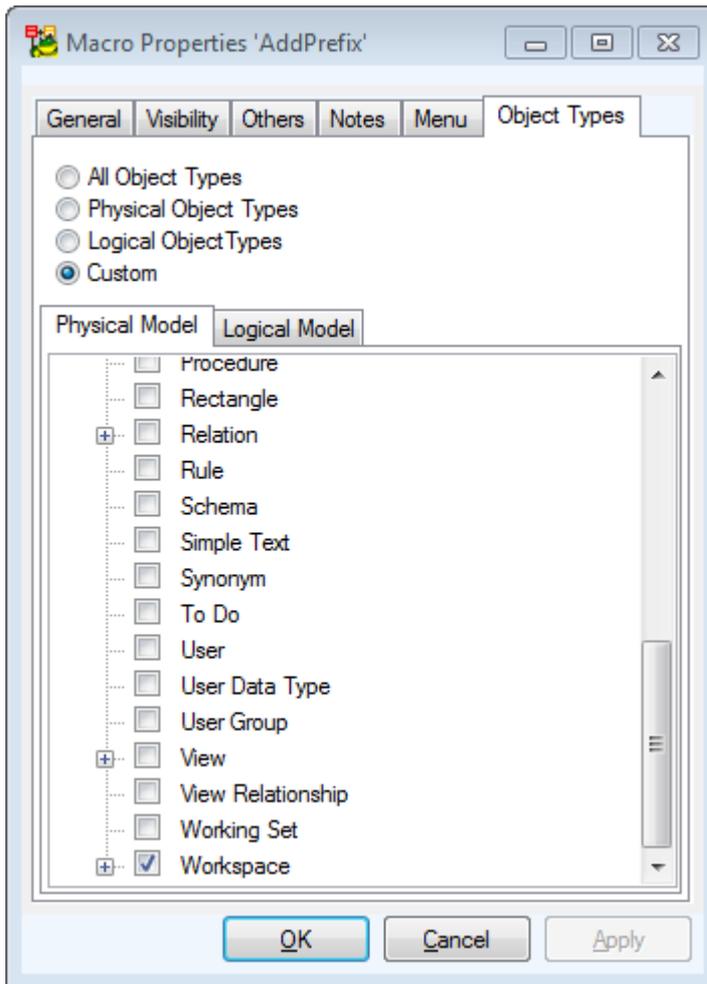
5. On tab **Visibility**, select where you want to apply the macro – **Physical Model**.
6. On tab **Menu**, define whether you want to display the macro in:
  - Macro menu,
  - pop-up menu,
  - both places.

Parameter **Path** specifies position in main menu (or pop-up menu). Example: '*Test\My Items*'. In this example, you decide to display it only in pop-up menu.

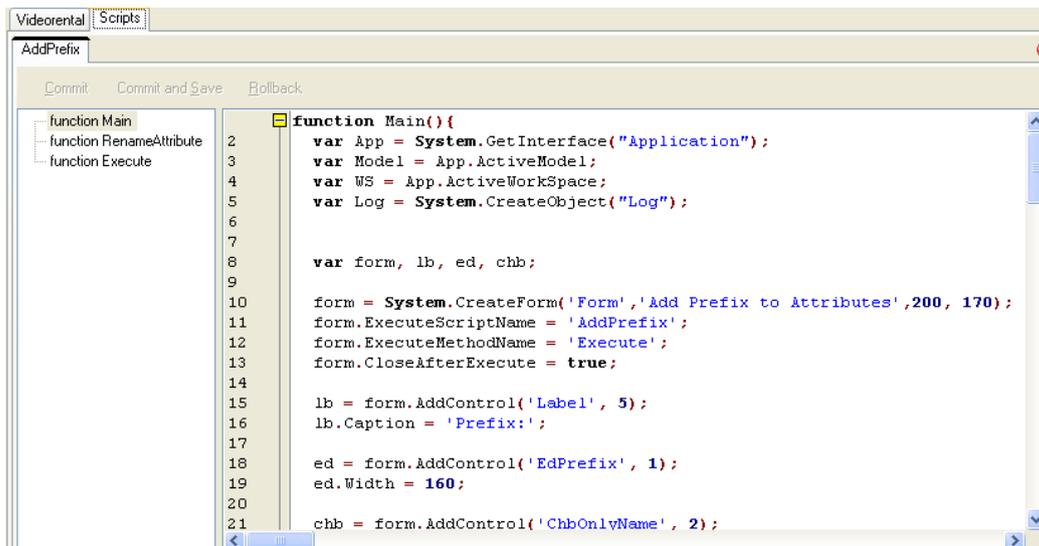
Path box is empty as 'Macros' item is set as default.



7. On tab **Object Types**, select in which object pop-up menu you want to display it. Select *Workspace*. Confirm **OK**.



8. Double-click the macro in Script Explorer to open Script Editor. Modify the default code.



```

function Main(){
    var App = System.GetInterface("Application");
    var Model = App.ActiveModel;
    var WS = App.ActiveWorkspace;
    var Log = System.CreateObject("Log");

    var form, lb, ed, chb;

    //Create form
    form = System.CreateForm('Form', 'Add Prefix to Attributes', 200, 170);
    //Add script that should be executed after you click the Execute button
    form.ExecuteScriptName = 'AddPrefix';
    form.ExecuteMethodName = 'Execute';
    form.CloseAfterExecute = true;

    //Add component Label on the form
    lb = form.AddControl('Label', 5);
    lb.Caption = 'Prefix: ';

    //Add component Edit on the form
    ed = form.AddControl('EdPrefix', 1);
    ed.Width = 160;

    //Add component Checkbox on the form
    chb = form.AddControl('ChbOnlyName', 2);
    chb.Caption = 'Modify Caption';
    chb.Checked = true;

    //Macro can be executed for Attributes, Model or Workspace
    //If macro is executed only for attributes, it relates only to selected attributes.
    var OnlyAttributes = true;
    var i, SelectObject;
    for(i=0; i<This.Count;i++)
    {
        SelectObject = This.GetObject(i);
        if (SelectObject.ObjectType!=2003) //2003 = Attribute
        {
            OnlyAttributes = false;
        }
    }

    //Variable will be accessible also in event via calling Instance.VariableName (Instance.OnlyAttributes)
    form.AddUserVariable('OnlyAttributes', OnlyAttributes);
    //Registered objects will be accessible in events.
    form.RegisterObject(This, 'SelectedObjects');
    form.RegisterObject(Model, 'Model');
    form.RegisterObject(Log, 'Log');

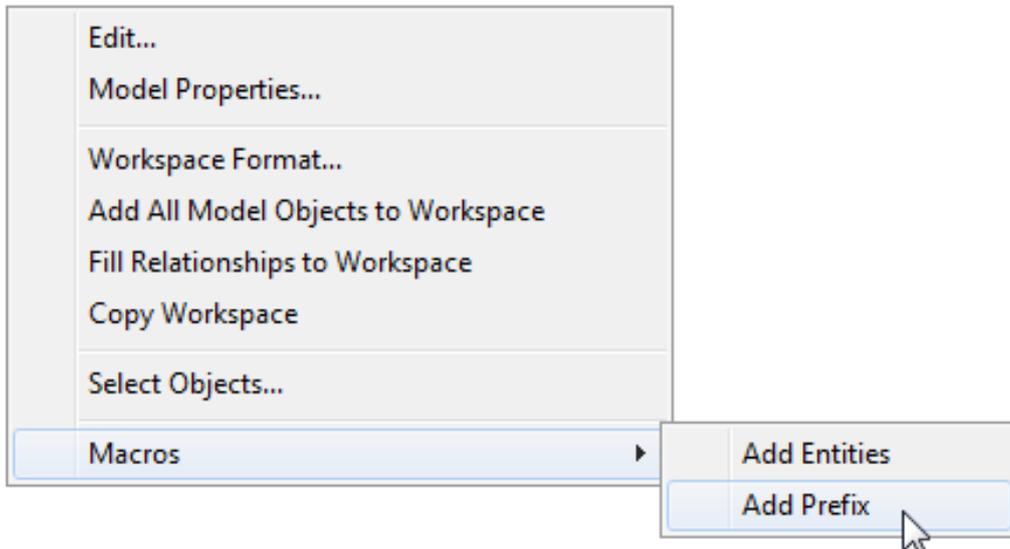
    form.ShowModal();
}

function RenameAttribute(Attribute)
{
    Log.Information('Attribute has been renamed from ''+Attribute.Name+''' to ''+EdPrefix.Text+Attribute.Name+'''');
    if (ChbOnlyName.Checked)
    {
        Attribute.Caption = EdPrefix.Text+Attribute.Caption;
    }
    else
    {
        Attribute.Name = EdPrefix.Text+Attribute.Name;
    }
}

function Execute()
{
    var i, j, SelectObject, Ent;
    if (Instance.OnlyAttributes)
    {
        for(i=0; i<SelectedObjects.Count;i++)
        {
            SelectObject = SelectedObjects.GetObject(i);
            RenameAttribute(SelectObject);
        }
    }
    else
    {
        for(i=0; i<Model.Entities.Count; i++)
        {
            Ent = Model.Entities.GetObject(i);
            for(j=0; j<Ent.Attributes.Count; j++)
            {
                SelectObject = Ent.Attributes.GetObject(j);
                RenameAttribute(SelectObject);
            }
        }
    }
    Model.RefreshModel();
}

```

9. Click **Commit and Save**.
10. Right-click the Workspace |**Macros** |**Add Prefix** to open the user form.



## About Metamodel in Toad Data Modeler

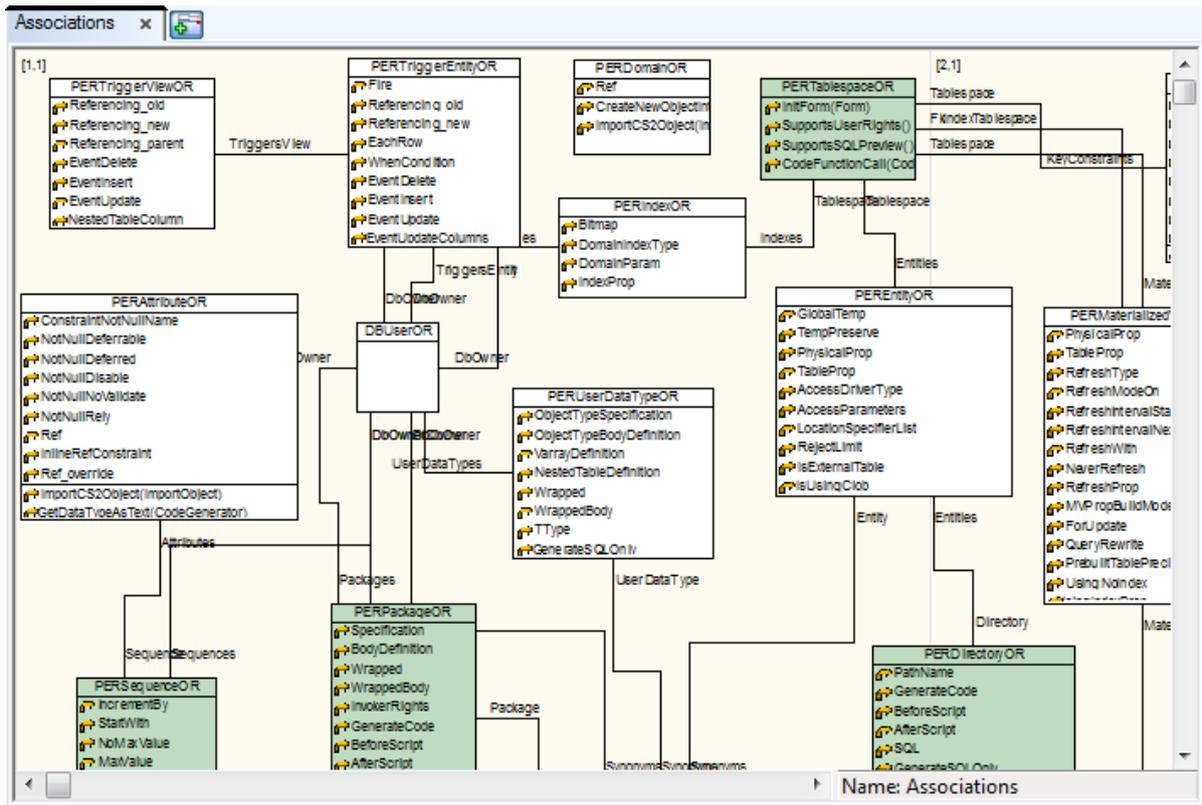
Toad Data Modeler implements the concept of metamodels. Metamodels are accessible only in **Expert Mode** (via **Package Explorer**).

Metamodel is a graphical representation of objects, classes, methods and relationships between them in a specific **Package**.

Using metamodels, you can create your own classes, methods and properties and define relationships between those items and items that are created in the application by default.

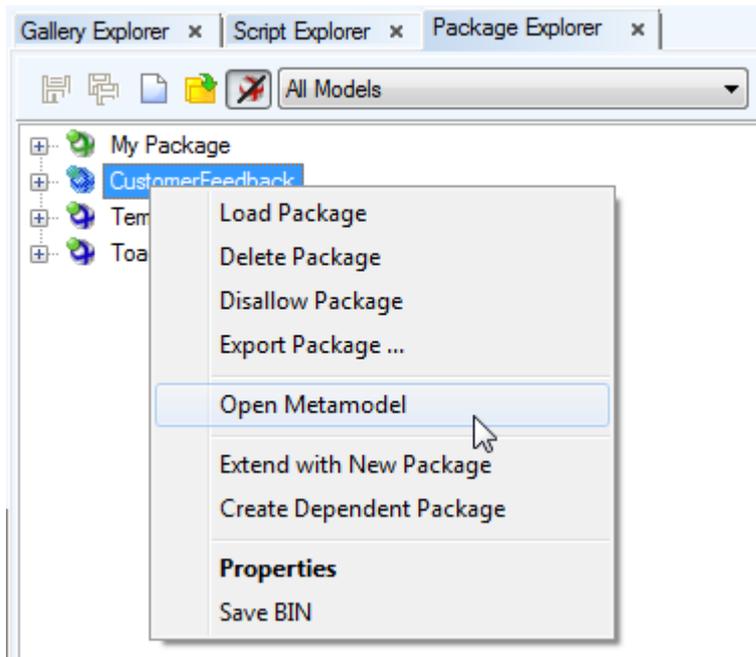
- i** Important: It is recommended to only use lower and uppercase letters for naming your objects in metamodels (no numbers and special characters).

**Example:** The metamodel of Oracle package



## Open Metamodels

1. Open  (or select **Expert Mode Menu | Customization | Package Explorer**).
2. Select a package.
3. Right-click the package and select **Open Metamodel**.



## Add Existing Classes to Metamodel

1. Right-click the work area and select **Add Class**.
2. In the **Class Selection** dialog, select a class and define settings on tab **Settings**.

Result: The selected class will be added to your metamodel and you will be able to modify it.

## Create Classes

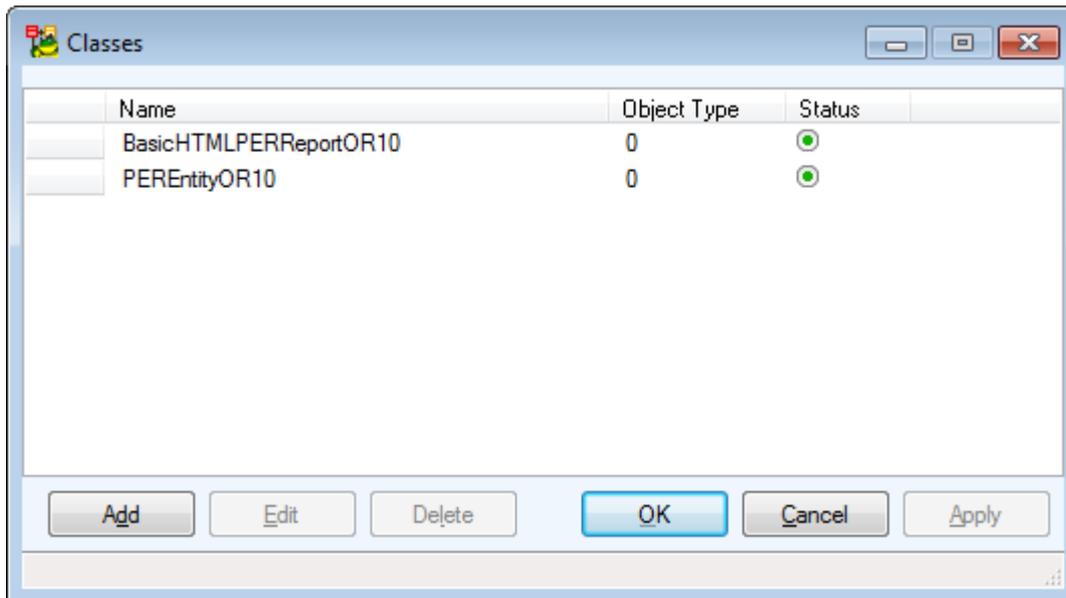
Click **Class**  on the toolbar and then anywhere in the work area.

### *To add multiple classes on the Workspace*

1. Press SHIFT and click the **Class** icon. A blue frame displays in the icon.
2. Click the work area as many times as many classes you want to add.
3. Right-click the work area (or click the **Class** icon again) to turn this function off.

### *To see all classes of your model*

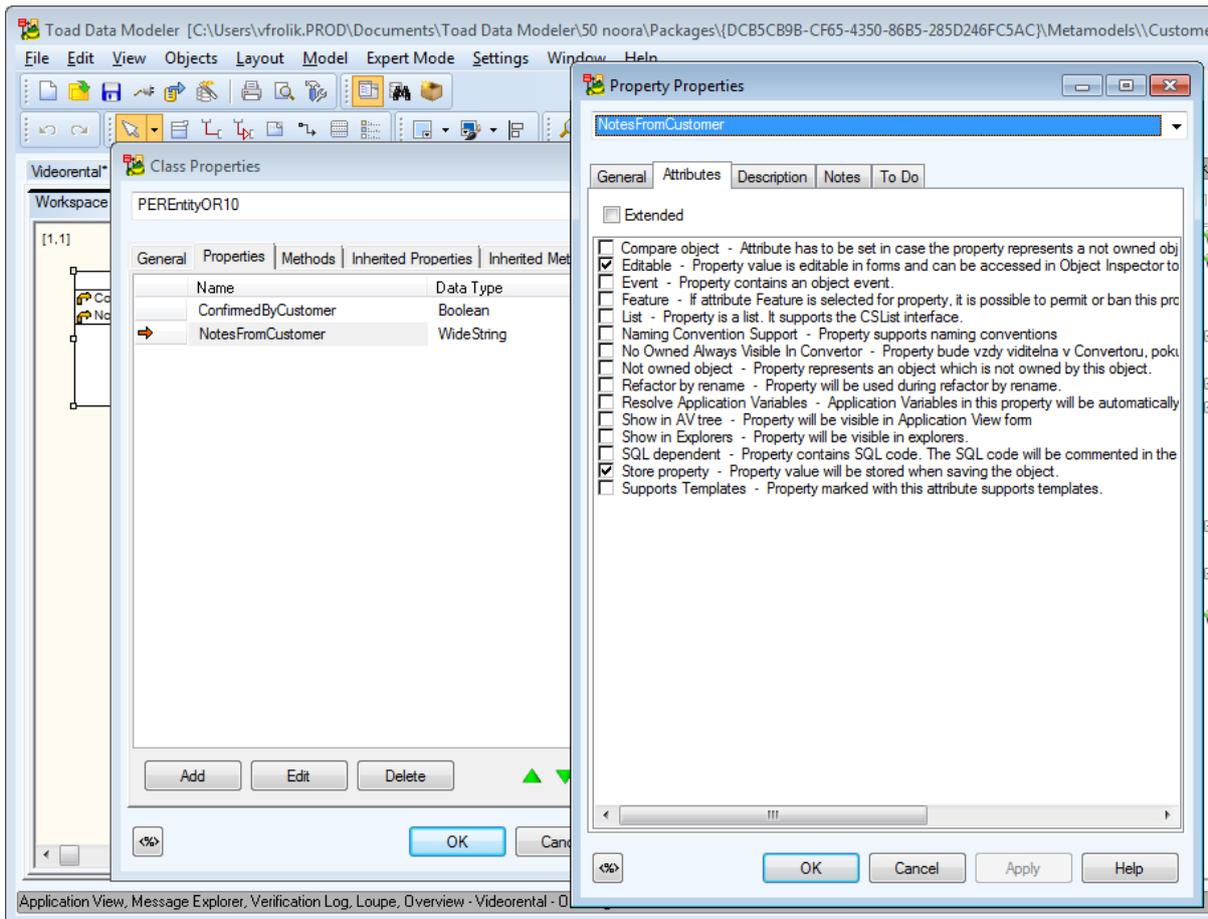
Select **Model | Model Items | Classes**. In the **Classes** dialog, you can add, edit classes, edit their names (F2) and delete them.



## Edit Classes

Double-click the class graphics on the Workspace  
or

Edit the class in the **Classes** dialog (**Model** menu | **Model Items** | **Classes** | **Edit**).



## General Tab

## Description

Name	<p>Defined object class. Its value consists of CSAOClassName + Abbreviation.</p> <p><b>Example:</b></p> <p>PERSequenceOR</p> <p>PER = Physical Entity Relationship model</p> <p>Sequence = Sequence</p> <p>OR = Oracle (all Oracle databases. OR10 stands for Oracle 10g, OR9 is for Oracle 9i etc.)</p>
Object Type	Numeric representation of object class without dependence on database system. A list of Object types is available in the TDM Reference document.
External	This checkbox has just an informative value. External class represents a class created in metamodel.
Default Name	Default name that will be assigned to the object after its creation. If you add "%d" to the default name, a numeric value will be added to the end of the default name.

Category	Category selection box. To see a list of Categories of your model, click the small button on the right.
<b>Properties Tab</b>	List of class properties.
<b>Methods Tab</b>	List of methods that belong to the class.
<b>Inherited Properties Tab</b>	Properties defined in predecessors.
<b>Inherited Methods Tab</b>	Methods defined in predecessors.
<b>Description Tab</b>	You can enter the class description here.
<b>Notes Tab</b>	Write notes on this tab.
<b>To Do Tab</b>	You can enter To Do tasks related to the class here.

## Create Properties

1. In the **Class Properties** form, click the **Properties** tab.
2. Click **Add**.
3. Confirm **Apply**.
4. **Edit** the new item and define properties of the new property.

## Edit Properties

Double-click a property or press **Edit** in the **Class Properties** dialog | **Properties** tab.

<b>General Tab</b>	<b>Description</b>
Name	Name of property
Data Type	Property data type
Default value	Property default value
External	Only properties that are marked as External are taken to particular class. Every new property defined by users must be set as External, otherwise the property will not be accessible in the class. If you need to add a property for just an informative purpose to your metamodel, uncheck the checkbox <b>External</b> .
<b>Implementation Area</b>	<b>Description</b>
Get Script Method and Set Script Method	Access methods for property. Name consists of ScriptName.MethodName - without brackets. Click the button next to the Get Script Method or Set Script Method

fields to assign default values there.

**Example:** PERSequenceOR.GetIncrementBy

Package	Name of package where the access methods are stored
Overridable	Select to set the property as Overrideable.
Dynamic	Select to set the property as Dynamic.
Read Only	Select to set the property as Read Only.
Write Only	Select to set the property as Write Only.
<b>Attributes Tab</b>	<p>On tab <b>Attributes</b>, you can assign attributes to properties. Every attribute may change class behaviour.</p> <p><b>Example:</b> You create a new property MyDescription. This property will be used for storing data entered into a new Text Field in the Entity Properties form. For this purpose, the following property attributes must be enabled: Editable and Store.</p> <p><b>Tip:</b> A list of attributes is accessible via <b>Model   Attributes</b> menu.</p>

## Create Methods

In the **Class Properties** dialog | **Methods** tab | **Add**.

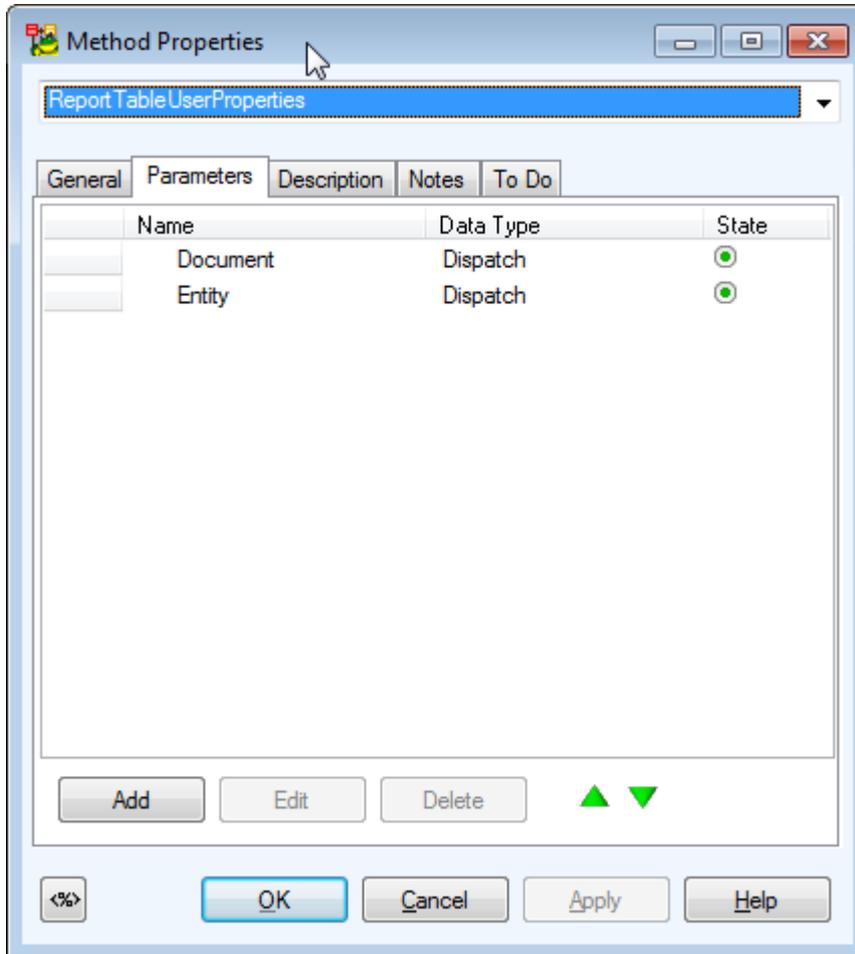
## Edit Methods

Double-click a method or press **Edit** in the **Class Properties** dialog | **Methods** tab.

General Tab	Description
Name	Name of method
Result Data Type	Data type that the method returns
External	Only methods that are marked as External will be taken to particular class. Every new method defined by users must be set as External, otherwise the method will not be accessible in the class. If you need to add a method for just an informative purpose to your metamodel, uncheck the checkbox <b>External</b> .
Implementation Area	Description
Method	Location of the access methods of property in the form:  ScriptName.MethodName. After you click  , default value will be inserted to this box.

Package Name of package where the access methods are stored.

**Parameters Tab** Add parameters to methods on this tab.  
Click **Add** to create a new method parameter.



## Generalization

Generalization is a link that defines a relation between two classes. Using generalizations, you can model inheritance.

Child class has all properties and methods of parent class, plus it may add new behaviors. If you create a new class (class name doesn't exist), then it will be necessary to define an inheritance to more general classes from the core or its successors.

### To create Generalization

1. Click  on the toolbar.
2. Move your mouse cursor over the work area.
3. Click parent class and then the child class.

## Edit Generalizations

- Double-click the generalization line on the Workspace.

or

- Edit the generalization in the **Generalizations** dialog (**Model** menu | **Model Items** | **Generalizations**).

## Associations / Aggregations

Associations and aggregations represent a relationship between two classes. If one class owns another class, then it's an aggregation. (Attribute has a Domain, Entity is owned by Model and Model has a list or collection of Entities etc.)

### To create Associations / Aggregations

1. Click  on the toolbar.
2. Move your mouse cursor over the work area.
3. Click the first class and then the target class.

## Edit Associations

Double-click the association line on the Workspace.

or

Edit the association in the **Associations** dialog (**Model** menu | **Model Items** | **Associations**).

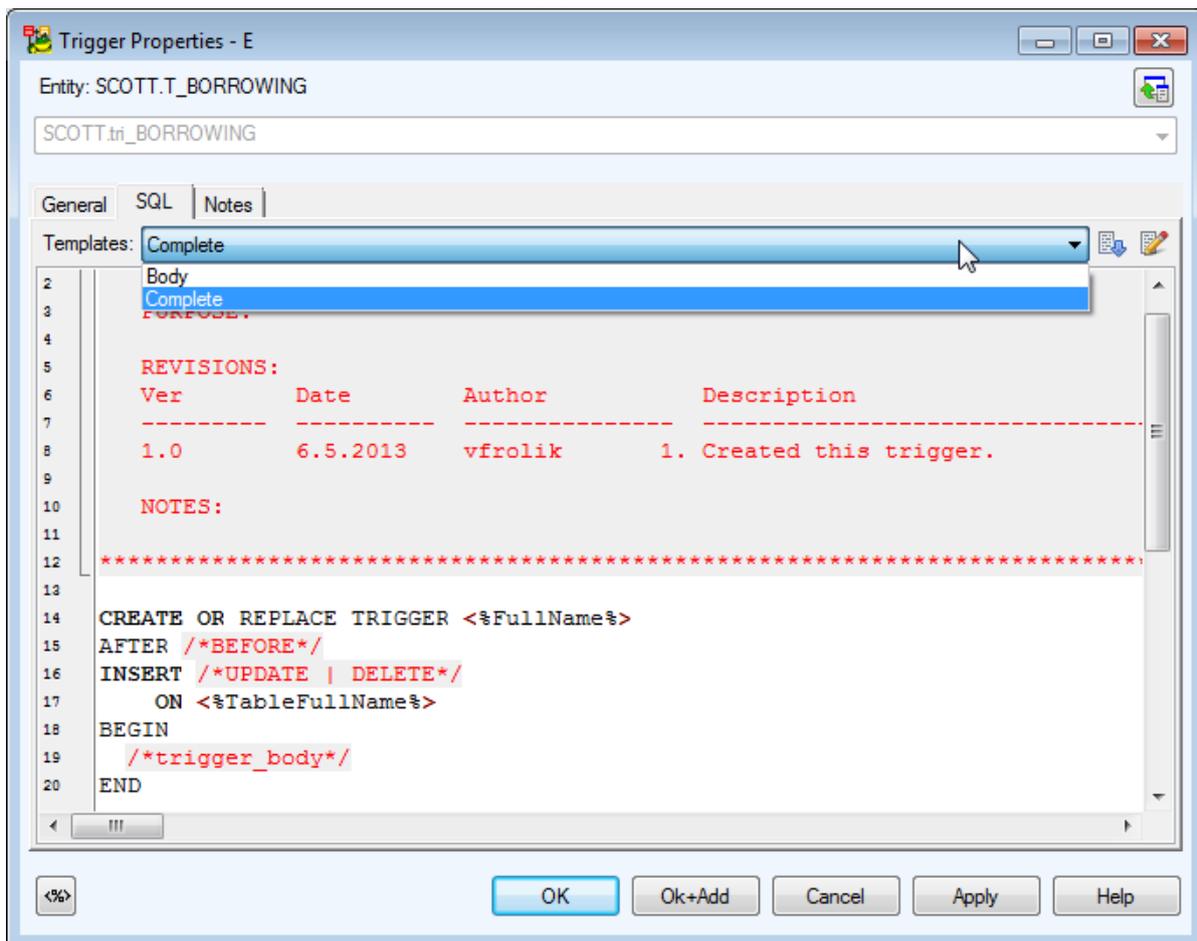
General Tab	Description
Name	Name of the association/aggregation
Advanced Tab	Description
Role	Name of property that will be added to class
Aggregation	Represents an owner of the second class.
Many	Property created in the second class will not be of the Dispatch type but the List type, and will work as a collection.
External	External associations create new properties in the selected class. Uncheck the <b>External</b> checkbox not to create the properties. In both cases, a relationship with information about linkages is created.
<b>Attributes1, Attributes2 Tab</b>	Represent attributes that can be assigned to classes on both sides of the association or aggregation. Names of properties are defined on tab <b>Advanced</b> in the <b>Role</b> box. Every attribute may change class behavior. A list of attributes is accessible via <b>Model</b> menu   <b>Attributes</b> .

# About Templates

Toad Data Modeler allows you to:

- Use pre-defined templates for properties that contain SQL code in your model (SQL, Before Script, After Script etc.).
- Create your own user templates for these properties.
- Set a default template for each property.
- Import Toad for Oracle templates.

**Example:** You have created an entity trigger. Edit it and see the **SQL** tab in the **Trigger Properties** dialog. The SQL code has been pre-defined = particular default template has been used (according to the database). You don't have to write the code manually for every new created trigger.



## To select another than default template in object Properties dialog

Press CTRL+A to highlight all text in the text box and select another template from the **Templates** box.

- i** Note: To replace one template with another, **all text should be selected**. Otherwise, the new template selected from the **Templates** box will be inserted to the position of cursor in the box. This behavior allows you to put together partial templates.

Option	Description
Templates	Shows all active templates for this object.
	Inserts the selected template.
	Opens the Template Editor.

All templates are available in [Template Editor](#).

**i** Note: Templates for Properties with SQL Code - To find out for which properties you can create the templates, see the Reference Guide and search the *Supports Templates* attribute.

## Template Editor

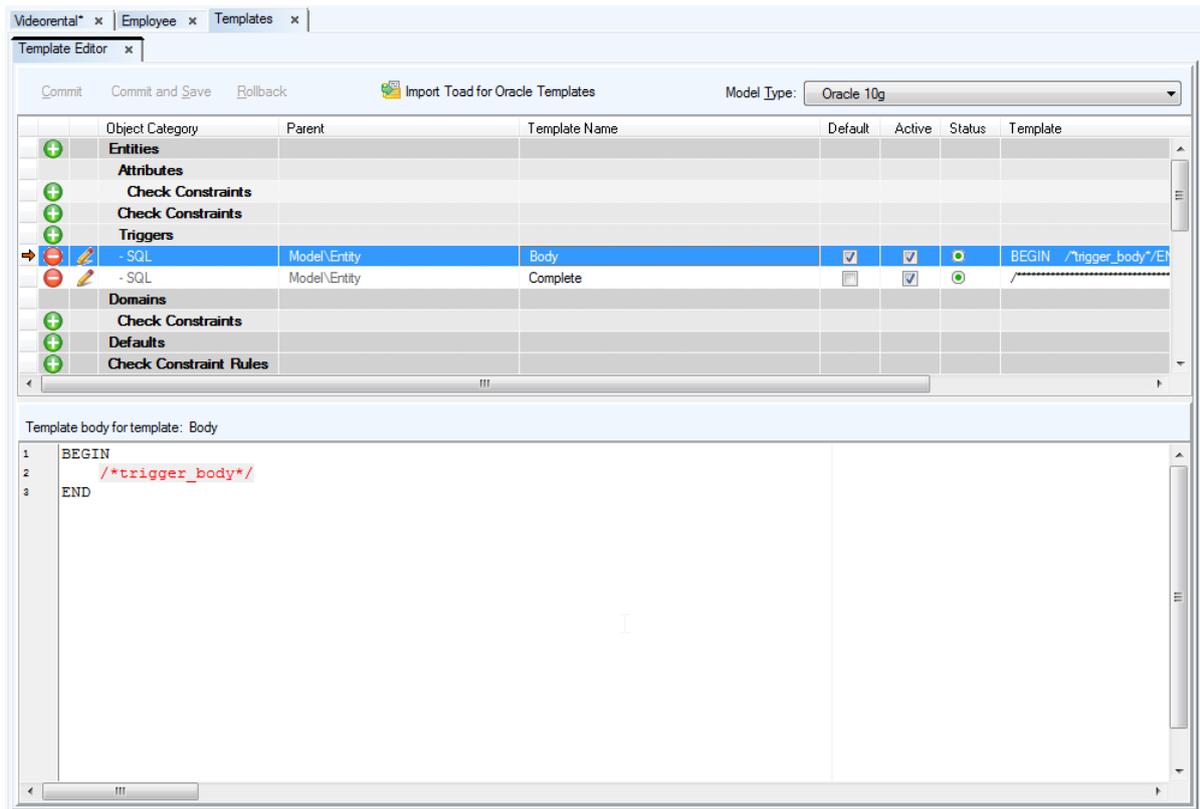
### **To open Template Editor**

Select **Settings | Templates**.

### **To display templates for particular database or database version**

1. From the **Model Type** box, select the database or particular database version.
2. Click **Load Templates**.

**i** Note: This option is available only if packages with the templates for selected database haven't been loaded so far.



Option	Description
Commit	<p>Confirms changes made in the Template Editor and saves them to particular packages.</p> <p><b>i</b>   Note: Packages are not saved to your hard disk.</p>
Commit and Save	<p>Confirms changes made in the Template Editor and saves them to particular packages (.txg files). The packages are saved to your hard disk.</p>
Rollback	<p>Cancels changes made in Template Editor.</p>
Import Toad for Oracle Templates	<p>Imports templates from Toad for Oracle to the Toad Data Modeler Template Editor.</p> <p><b>i</b>   Note: This option is available only if you have Toad for Oracle installed on your computer.</p>
Model Type	<p>Select a database or particular database version for which you want to display the templates.</p> <p><b>i</b>   TIP: Feel free to open Templates Editor for different database platforms or versions, dock the windows, compare the templates.</p>

Option	Description
	Creates a new template.
	Deletes the selected template. Click <b>Commit</b> to confirm the deletion.
	Opens the <b>Template Properties</b> dialog.
Object Category	List of objects (entities, attributes, triggers, check constraints, views, procedures etc.) with list of available SQL properties (SQL, Before Script , After Script etc.).
Parent	Information on parent object
Template Name	Specify any name for your template to distinguish it from others.
Default	Select this checkbox to set the template as default. The template will be pre-defined in the object <b>Properties</b> dialog.
Active	Select this checkbox to display the template in the <b>Templates</b> box in the object <b>Properties</b> dialog.
	
Template	Provides a quick view on the body of the SQL code.
Template body for template	Write the SQL code to this window. <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;"> <p><b>i</b> Note: Remember to save the changes <b>Commit</b> or <b>Commit and Save</b>.</p> </div>

## Available Pre-defined Templates

There are two pre-defined templates – *Body* and *Complete*. Both relate to the **Generate SQL Only** checkbox on tab **General** of object **Properties** dialog.

- **Body** – The **Generate SQL Only** checkbox is deselected.

**Example** of default code for trigger:

```
BEGIN

/*trigger_body*/

END
```

- **Complete** – The **Generate SQL Only** checkbox is selected. (It means that settings on tab **General** are ignored in final SQL code.)

**Example** of default code for trigger:

```
CREATE OR REPLACE TRIGGER <%%FullName%%>
```

```

AFTER /*BEFORE*/

INSERT /*UPDATE | DELETE*/

ON <%%TableFullName%%>

BEGIN

/*trigger_body*/

END

```

To set this property for your user templates, see the **Template Properties** dialog, **General** tab and select or clear the **Generate SQL Only** checkbox.

## Where Templates Are Stored

All pre-defined templates are saved in the *Templates.txg* file that is copied among user packages during first start-up of Toad Data Modeler.

All user templates are by default saved to *My Package.txg*.

Both files are stored among user packages at (default location):

C:\Documents and Settings\user name\My Documents\Toad Data Modeler\Installation name\Packages\{GUI}

Why are pre-defined templates stored in another package? - Once our team brings new pre-defined templates, you will be able to replace the old pre-defined templates with the new ones (*Templates.txg*). Your user templates will stay preserved in your *My Package.txg*.

We do NOT recommend to modify the pre-defined templates. If you do so, please remember to save the template to *My Package.txg* (see the **Template Properties** dialog, **General** tab, **Package** area.)

# Manage Templates

### To create a user template

1. In Template Editor, select **Object Category** (e.g. Triggers).
2. Click  and write the code in the **Template body for template** window.
3. Confirm the changes **Commit** or **Commit and Save**.

### To change properties of a user template (e.g. location, visibility etc.)

1. Select a template.
2. Click  to open the **Template Properties** dialog.
3. Check properties of the template.

Option	Description
--------	-------------

<b>General Tab</b>	
--------------------	--

Option	Description
Name	Write a name of your template.
Property Selection	Select for which database, object type and property the template should be used.
Active Template	Select this checkbox to display the template in the <b>Templates</b> box in the object <b>Properties</b> dialog.
Default Template	Select this checkbox to set this template as default in particular object <b>Properties</b> dialog.
Generate SQL Only	Select this checkbox to set the property <b>Generate SQL Only</b> enabled in the object Properties dialog. <a href="#">Available Pre-defined Templates</a>
Package	Select a package where you want to save the template. By default, user packages are saved to <i>My Package.txg</i> . <a href="#">Where Templates Are Stored</a>
Lock Package	Sets the ReadOnly property of the .txg file on the disk.
<b>Visibility Tab</b>	Select databases and database versions for which the template should be valid and available.

4. Confirm **OK**.
5. Confirm **Commit and Save**.

### **To set a template as Default**

1. Select a template in Template Editor.
2. Select the **Default** checkbox for the template.
3. Confirm **Commit and Save**.

### **To disable a template**

(not to display it in the combo-box in the object **Properties** dialog)

1. In Template Editor, find the template that you want to disable and uncheck the **Active** checkbox.
2. Confirm **Commit and Save**.

### **To delete a template**

1. In Template Editor, select the template that you want to delete.
2. Click  .
3. Confirm **Commit and Save**. Now it is not possible to edit the template.

**i** Note: After you reopen the Template Editor, the deleted item will be removed. The template has been deleted from the .txg package.

# Toad for Oracle Templates

Toad Data Modeler allows you to import the templates you created in Toad for Oracle and refresh them at any time you need.

We do NOT recommend to modify these templates in Toad Data Modeler as export of the templates to Toad for Oracle is not possible.

Toad Data Modeler imports Toad for Oracle templates of the following objects:

- Stored Procedures
- Functions
- Entity Triggers
- Packages (object in Oracle)

## To import Toad for Oracle template to Toad Data Modeler

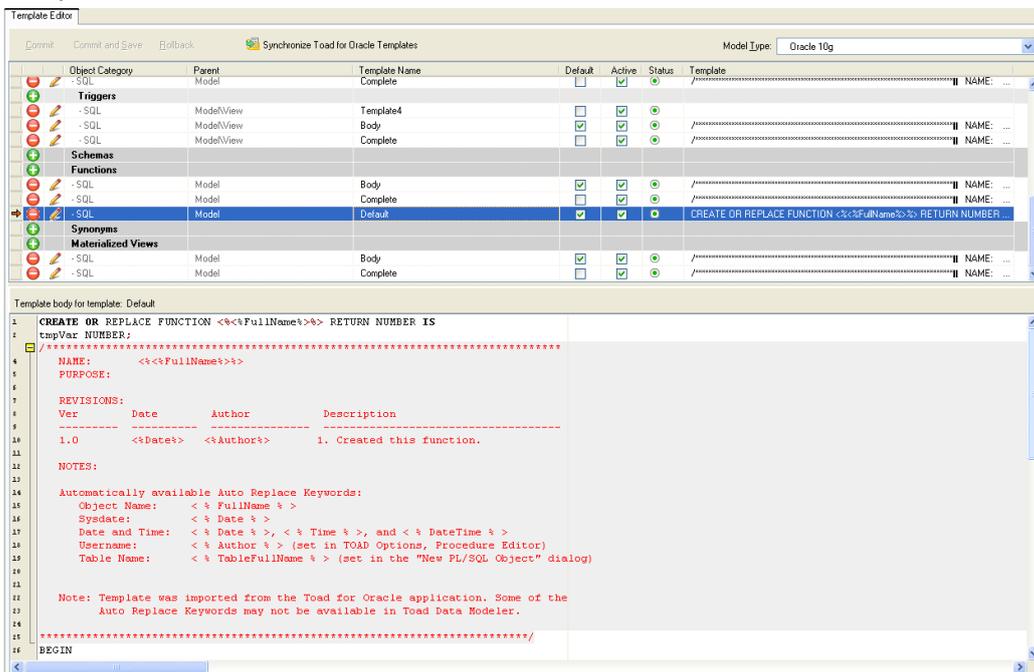
1. Open Template Editor - Select **Settings | Template Editor**.

2. Click .

 Note: This option is available only if you have Toad for Oracle installed on your computer.

3. Confirm **OK** and take notice of new templates in Template Editor (e.g. "Default" templates).

 Note: The Toad for Oracle templates are stored in TOAD.tgx file at: C:\Documents and Settings\user name\My Documents\Toad Data Modeler\Installation name\Packages\{GUI}



**To refresh your existing Toad for Oracle templates in Toad Data Modeler**



## Toad for Oracle Auto Replacement Words

Toad Data Modeler allows you to use Application Variables in your templates.

Toad for Oracle templates can have Auto Replacement Words.

During import of Toad for Oracle templates, the Toad for Oracle Auto Replacement Words are replaced by Toad Data Modeler Application Variables. See how:

Toad for Oracle Auto Replacement Word	Toad Data Modeler Application Variable
%YourObjectName%	<%FullName%>
%Date%	<%Date%>
%SysDate%	<%Date%>
%DateTime%	<%DateTime%>
%Time%	<%Time%>
%TableName%	<%TableFullName%>
%UserName%	<%Author%>

**i** Note: Not all Toad for Oracle Auto Replacement Words are supported.

## Rename

### **To rename your model**

Right-click the model name in **Model Explorer** (or **Application View**) and select **Rename**.

### **To rename objects on Workspace**

1. Select the object (entity, view etc.) and press F2.
2. Enter new name.
3. Click anywhere else on Workspace to apply changes.

### **To rename object in Model Explorer**

Select an object and press F2 or right-click an object and select **Rename**.

# Tips and Hints for Large Models

- [How to divide your large model?](#)
- [How to work concurrently in two or more places of the same Workspace?](#)
- [How to avoid creating the same or similar objects?](#)
- [How to display related entities, which are far from each other, close to each other?](#)
- [How to transparently display an entity that has a lot of relationships in your ER diagram?](#)
- [How to colorfully distinguish entities in your model?](#)
- [How to select objects in ERD by schema or category?](#)
- [Print large model](#)
- [Generating HTML reports](#)

## How to divide your large model?

Create new Workspaces!

Workspaces (WS) are similar to submodels. A model can have as many WS as you need. Each WS can display different (or same) parts (objects) of your large model. Each WS can have a different format or display view. On the Workspaces, you can manage your model objects quickly, easily and comfortably.

To create a new WS, simply click  on the toolbar.

For more information, see [Designer and Workspace](#) on page 36.

## How to avoid creating the same or similar objects?

Use the Gallery feature. There, you can store frequently used parts of your models - entities, single attributes, stored procedures and other objects. Then simply use the drag-and-drop technique to insert gallery items into your models. For more information, see [Gallery](#) on page 614.

## How to work concurrently in two or more places on the same Workspace?

Open another Designer for your Workspace (WS)!

Designer displays a Workspace (WS). You can open as many Designers (tabs) for the selected WS as you need. This allows you to display different parts of the same WS and also the same objects of the WS in a different zoom.

For more information, see [Designer and Workspace](#) on page 36.

---

### Scenario

Your model has one Workspace WS1. WS 1 represents main model and is large. You need to work with two parts of the model concurrently.

---

1. Right-click *Workspace1* item in the **Application View** or **Model Explorer** and select **Open Designer**. -  
> Another tab of the same name - *Workspace1* will open in the Application Window. It displays the same objects.

**Scenario:** You're working in the second Designer and need to move fast to find the required part/objects that you need to edit.

2. Click  on the toolbar.

In the **Overview** dialog, use drag&drop techniques to move the small frame in the **Overview** dialog. -> You will move on the WS at the same time. Change size of the frame at your convenience to zoom in or out the objects on the WS.

3. Find the required objects. (You can close the **Overview** dialog then.)
4. Now you can comfortably work in two places of the same WS at the same time.
  - You can switch between the two Designers (tabs *Workspace1*).
  - You can undock one of the Designers and see them in one screen, or move it to your second monitor if you have any.

## How to display related entities, which are far from each other, close to each other?

Create an entity shortcut!

Entity shortcut is another graphical representative of an entity. It's not a copy but the same object with the same properties. You can create as many shortcuts of an entity as you need (and of course, not only entities...).

For more information, see [About Shortcuts of Objects](#) on page 274.

---

### Scenario

*Entity1* and *Entity42* are related, however each is on a different page. You want to see them closer to each other.

---

**Scenario:** Create a shortcut of the *Entity 1* and place it next to the *Entity 42*. (You can also create a shortcut of the *Entity 42* and place it to the *Entity 1*.):

1. Find *Entity 1* on the Workspace.
2. Find *Entity 42* in **Model Explorer**.
3. Click the *Entity 42* item and drag it to the WS where you want to add the shortcut, next to the *Entity1*.

## How to transparently display entity that has a lot of relationships?

Create an entity shortcut and move shortcuts of some relationships to it!

---

### Scenario

Entity *Film* has five relationships leading from or to it. You want to create a shortcut of this entity, place it next to it, and move two relationships to it.

---

**Scenario:** Create an entity shortcut on the Workspace.

1. Click the *Film* entity on the WS and hold the mouse key down.
2. Press CTRL+SHIFT keys and hold them down.
3. Drag the *Film* entity on the WS.
4. Release the mouse button and then the keys.

**Result:** There are two shortcuts of the *Film* entity on the WS - *Film : 1* and *Film : 2*.

**Scenario:** Create shortcuts of the two relationships of the *Film* entity.

5. Right-click the WS and select **Add Selected Objects to Workspace**.
6. From the dialog, select the relationships that you want to 'redirect' to the *Film : 2* shortcut. (See the object details next to the relationship name in the dialog).
7. Confirm your selection.

**Result:** There are two shortcuts of the two selected relationships connected to the *Film : 2* entity shortcut.

**Scenario:** Remove redundant relationship shortcuts of the *Film : 1* shortcut.

8. Select the two needless relationship shortcuts on the WS (use SHIFT for multiple selection).
9. Press **Delete**.

**Result:** The selected shortcuts will be removed from the WS.

## How to colorfully distinguish entities in your model?

Assign selected entities to a Category!

Categories allow you to colorfully distinguish entities that logically go together. Each entity can be assigned only to one category.

For more information, see [Categories](#) on page 315.

---

### Scenario

You want to distinguish all entities that contain information on your customers on the Workspace.

---

1. Right-click the **Categories** item in Model Explorer and click **Add Category**.
2. Double-click the new category.
3. Define properties of the category - its name (*Customer Info*) and select a color (*Money Green*).
4. On tab **Objects**, select entities that you want to assign to the category.
5. Confirm **OK**.

**Result:** All the selected entities will have Money Green background color on all Workspaces of the model.

**Scenario:** As you have many categories in your model, you want to display a caption of categories on the Workspace.

6. Click  on the toolbar.
7. Click the work area where you want to place the caption.

**Scenario:** You want to display the entities of the category in Model Explorer (they will be Money Green in Model Explorer).

8. Right-click the **Model Explorer** and select **Settings**.
9. Select the **Use Colors of Category to Draw** checkbox.

For more information, see [Caption of Categories](#) on page 281.

## How to select objects in ERD by schema or category?

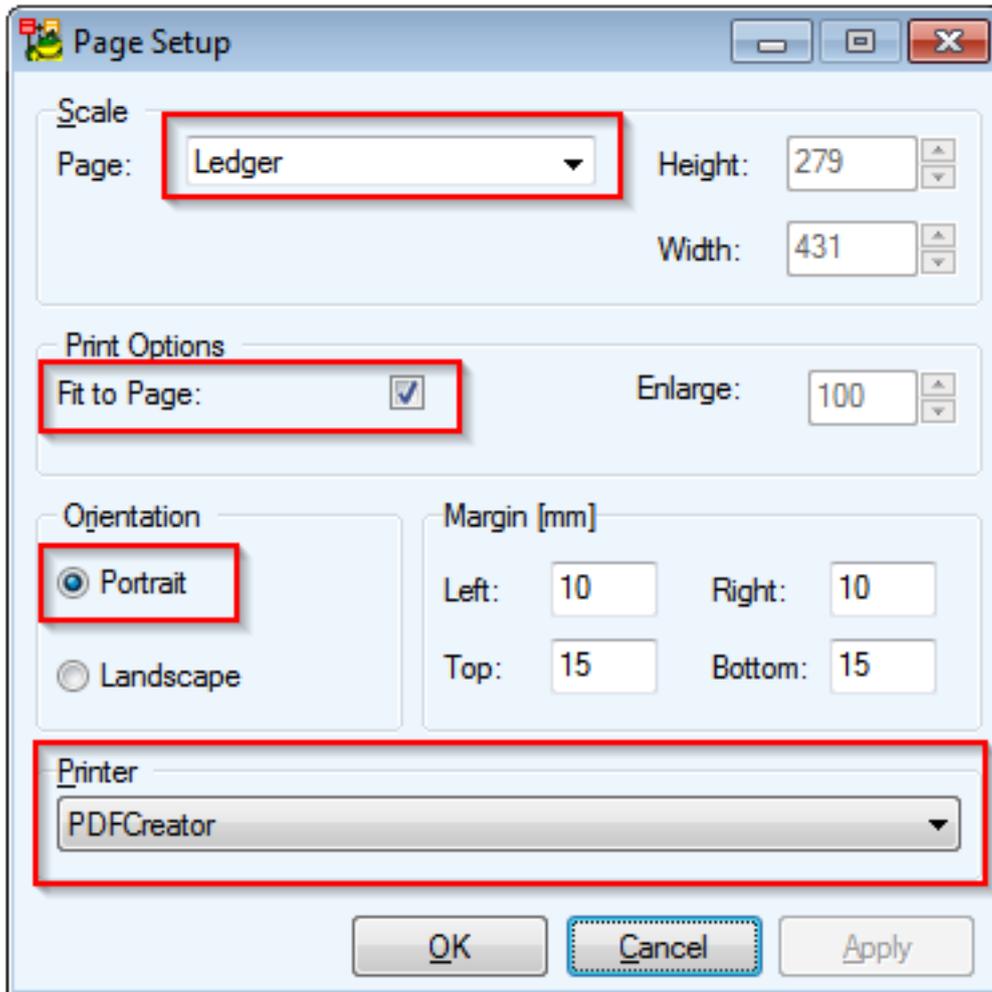
Right-click the Workspace and select **Select Objects**

## Print Large Model

### Is Print to PDF Possible?

Yes, you can print your ER diagram to PDF using "PDF printer" (Note: not all PDF printers are supported.) If you print your model on plotter, we recommend to print your ER diagram to PDF format first.

1. Install PDF Printer.
2. In Toad Data Modeler select **File | Page Setup**.
3. Select the PDF printer in the **Printer** box at the bottom of the dialog.
4. Define the following options:
  - **Page:** Ledger
  - **Print Options:** Fit to Page
  - **Orientation:** Portrait



5. Confirm **OK**.
6. Select **File | Print**.
7. On page **Settings | uncheck** checkboxes **Print Page Number** and **Print Frame**.
8. Print the output.

### Can I print whole ER diagram on one page?

Yes. In the **Page Format** dialog, select the **Fit to Page** checkbox.

### Is there a way to speed up the printing process?

Yes. Disable graphical options. Either select **Settings | Options | General**, uncheck the **Print Gradients** checkbox (it is disabled by default) or disable the graphical options **Gradient Effect**, **Graphical Display of Keys** etc. in the **Workspace Format** dialog.

## Generating HTML reports

With especially large models, it is possible to run out of memory while generating HTML report. In such situation, try the following suggestions:

- **Change Report Layout**

In **Select Layout** section of **Report Wizard**, try choosing **Frames - top/left menu** instead of **Frameless** layout. Frameless layout is not recommended for large models as it uses Javascript, which means more RAM is used. Additionally, navigating a Frameless HTML report for a large model might feel sluggish on lower-spec machines.

- **Close other models and restart Toad Data Modeler**

This will free some more RAM that can be used for report generation, so you're less likely to run into an Out of Memory error.

- **Generate report one workspace at a time**

In **Select Workspaces** section of **Report Wizard**, you can choose to generate report for specific workspaces only. The fewer workspaces, the less RAM will be needed for the generation process.

## Command Line Parameters

You can start Toad Data Modeler from command line and add additional parameters to make it perform various tasks on startup:

- Open files
- Create a new model
- Script execution
- Start logging
- Import Toad for Oracle Diagram or Model

### Open files

#### Definition:

Open-File -File:"Name=\*Path to file\*"[,Name=<string>]\*

Open-File -File:Name="\*Path to file\*"

**i** **TIP:** You can include more than one parameters. Use comma (,) to separate multiple parameters. See examples for more information.

#### Example:

The following command opens a model called Videorental.txp:

```
TDM.exe Open-File -File:Name="C:\Models\Videorental.txp"
```

The following command opens two models at once:

```
TDM.exe Open-File -File:Name="C:\Models\Videorental.txp", Name="C:\Program Files (x86)\Quest Software\Toad Data Modeler - Beta\Samples\Employee.txl"
```

## Create a new model

### Definition:

New-Model -ModelType:"\*\*model type\*\*" [-ModelName:"\*\*model name\*\*"]

### Examples:

Creates a new logical model with default name:

*TDM.exe New-Model -ModelType:"Logical Model"*

Creates Oracle 10g model with the specified name:

*TDM.exe New-Model -ModelType:"Oracle 10g" -ModelName:"My Physical Model"*

## Script execution

Executes script stored in the application. First parameter is **ScriptName** (name of the script), second is **MethodName**(name of the method in script) and the following parameters (**Par1Name**, **Par2Name**...) will be passed as parameters to the script method.

### Definition:

Execute-Script -Parameter:ScriptName="\*\*script name\*\*",MethodName="\*\*method name in script\*\*"  
[,Par1Name=**Par1Value**, Par2Name=**Par2Value**...] -TDM [-Silent]

### Examples:

*TDM.exe Execute-Script -*

*Parameter:ScriptName=MyScript,MethodName=MyMethod,Par1Name="ABC",Par2Name="DEF" -TDM -Silent*

### Switches:

-TDM: The executed script is stored in Toad Data Modeler.

-Silent: Script will be executed in Silent mode and you will be able to work with the application regardless the state of the executed script.

## Start logging

Writes messages to log. May be useful for sending reports to TDM developers.

### Definition:

Start-Log

## Import Toad for Oracle ER Diagram or Toad for Oracle Project

*TDM.exe Import-ToadForOracleERD -file:"NopathMovERD.erd"*

*TDM.exe Import-ToadForOracleERD -file:NopatMovERD.erd -Connection:"SERVER=OstDbServer:1521/ORCL, USER=movies,PROTOCOL=TNS,CONNECTAS=NORMAL,SAVEPASSWORD=1,ORACLEHOME=c:\oracle\product\10.2.0\db\_1,HOST=OstDbServer,SEVICENAME=ORCL, PORT=1521,LDAP=,METHOD=1"*

## Other Quick Tips

### Objects on the Workspace and Keyboard Arrows

- Move entities on the Workspace via keyboard arrows.

**i** | Note: To set the size of a step to move, select **Settings** | **Options** | **Graphics** | **Move**

**Objects by (mm/10)** (in tenths of millimeters).

- Select an entity, press SHIFT, hold it down and use the keyboard arrows to change size of the entity box.

## Navigation on Workspace

- CTRL + scroll mouse to zoom in/zoom out
- CTRL +, CTRL + Page Up to zoom in
- CTRL-, CTRL+ Page Down to zoom out
- Scroll mouse to move up/down on the Workspace
- SHIFT + scroll mouse to move to the right/left on the Workspace
- Holding down the middle button to move on entire page/Workspace
- Page Down, CTRL + down to move to next page
- Page Up, CTRL + up to move to previous page
- CTRL + left to move to the left page
- CTRL + right to move to the right page
- Click  to fit your entire ERD to screen.
- Press F11 to display the application in full screen mode.

## Make a Copy of Multiple Objects on the Workspace (CTRL+A, CTRL+C, CTRL+V)

- Before you press CTRL + V to paste the objects, close the **Model Explorer** dialog to accomplish the operation much faster.

(The larger your model is, the more significant difference in speed you will notice.)

## Print Models

- In **Settings | Options | General**, clear the **Print Gradients** checkbox for much faster print performance. (It is disabled by default.)

## HTML Report Layout

- For large models, **Frames - top menu** or **Frames - left menu** options are recommended.

(Frameless report layout is not recommended as it uses Java script that goes through all objects, which takes too much time if your model is large.)

# About Integration Options

Toad Data Modeler and Toad for Oracle products have started the integration process.

# Toad for Oracle - Basic Information on Product

Toad for Oracle® is a powerful application development tool built around an advanced SQL - PL/SQL editor. Using Toad, you can build and test scripts, PL/SQL packages, procedures, triggers, and functions. You can create and edit database tables, views, indexes, constraints, and users. The Schema Browser and Project Manager provide quick access to database objects.

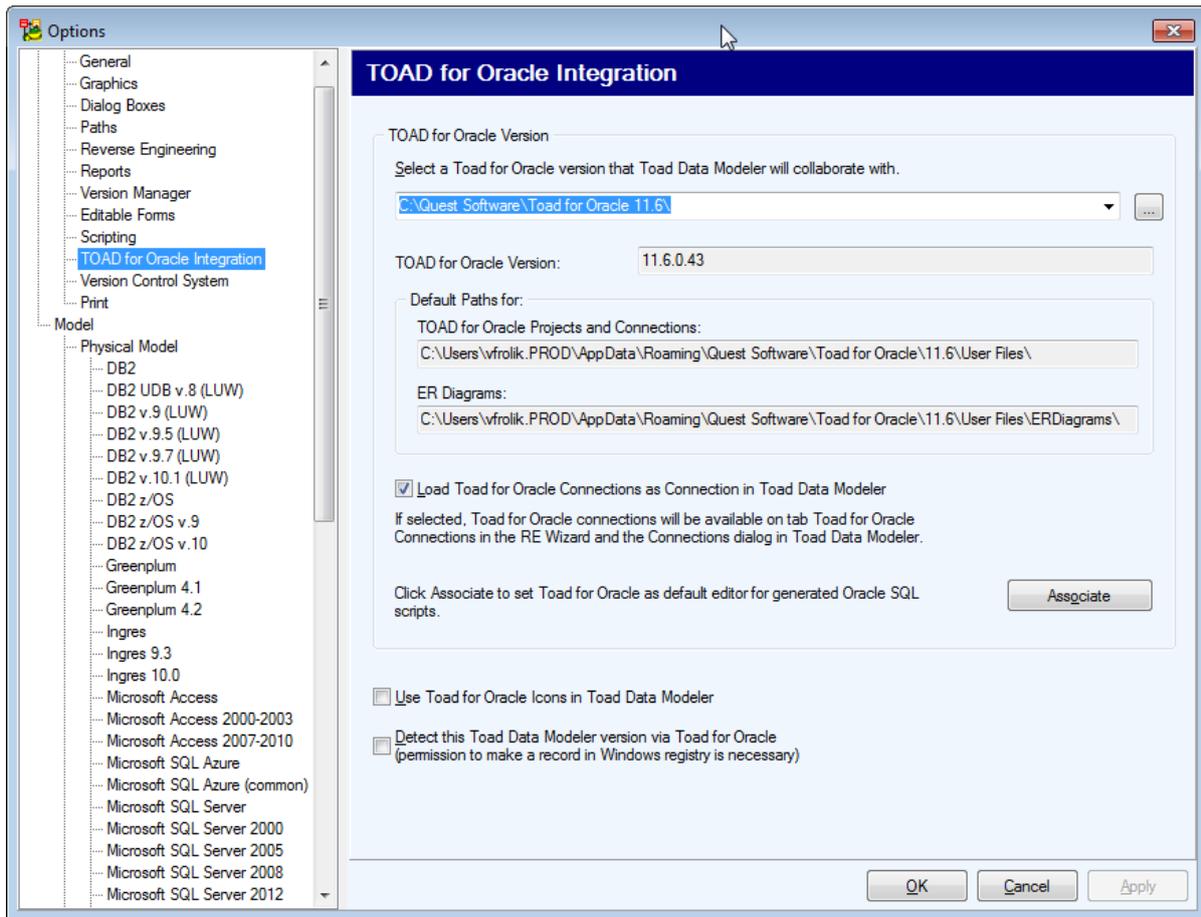
Toad's Editor provides an easy and efficient way to write and test scripts and queries, and its powerful data grids provide an easy way to view and edit Oracle data. With the optional DB Admin module you can manage space, compare schemas, monitor database performance, create new databases, maintain redo logs, perform health checks, and much more.

Toad for Oracle Integration in Toad Data Modeler

- Loading of Toad for Oracle® aliases
- Opening Toad for Oracle® projects
- Importing Toad for Oracle® ER diagrams
- Possibility to define Toad for Oracle® as a default editor for generated SQL scripts
- Possibility to use Toad for Oracle® icons in Toad Data Modeler
- Import of Toad for Oracle® templates. [Toad for Oracle Templates](#)

## Default Settings for Toad for Oracle Integration

Select **Settings** | **Options** | **Toad for Oracle Integration**.



**Note:**

1. Since Toad for Oracle® 10.0, the list of installed programs is not saved to registry but to a special file SettingsLocations.ini. What it means in reality: During start-up Toad Data Modeler searches for all Toad for Oracle® versions that you have on your computer. By default Toad Data Modeler pairs with the last installed Toad for Oracle® version. Once the Toad for Oracle® version is found, it is written in Config file of Toad Data Modeler.  
If you want Toad Data Modeler to pair with another Toad for Oracle® version, you have to define it manually in Toad Data Modeler: **Settings | Options | Toad for Oracle Integration**.
2. Toad for Oracle® cannot auto-detect Toad Data Modeler. To detect Toad Data Modeler via Toad for Oracle®, check the **Detect this Toad Data Modeler...** checkbox at the bottom in this window.

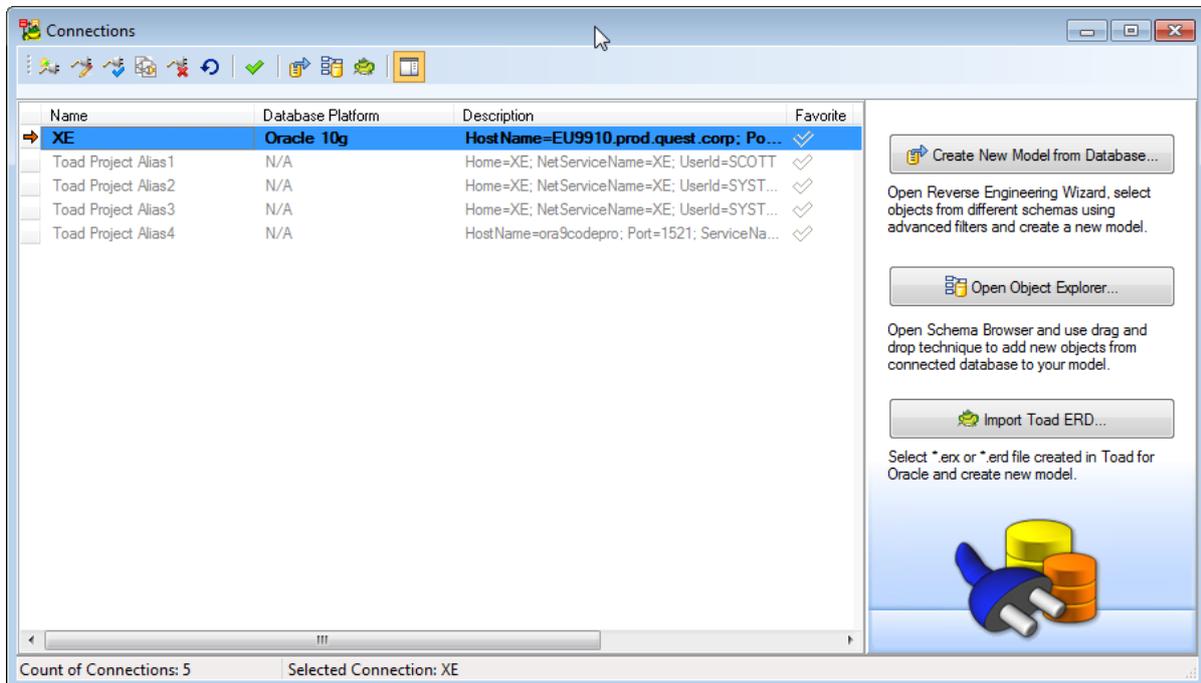
## Toad for Oracle® Connections

Toad Data Modeler recognizes Toad for Oracle connections and displays them in the:

- **Connections** dialog - click  on **Main Toolbar**.
- **Reverse Engineering Wizard** - click  on **Main Toolbar**.

Note:

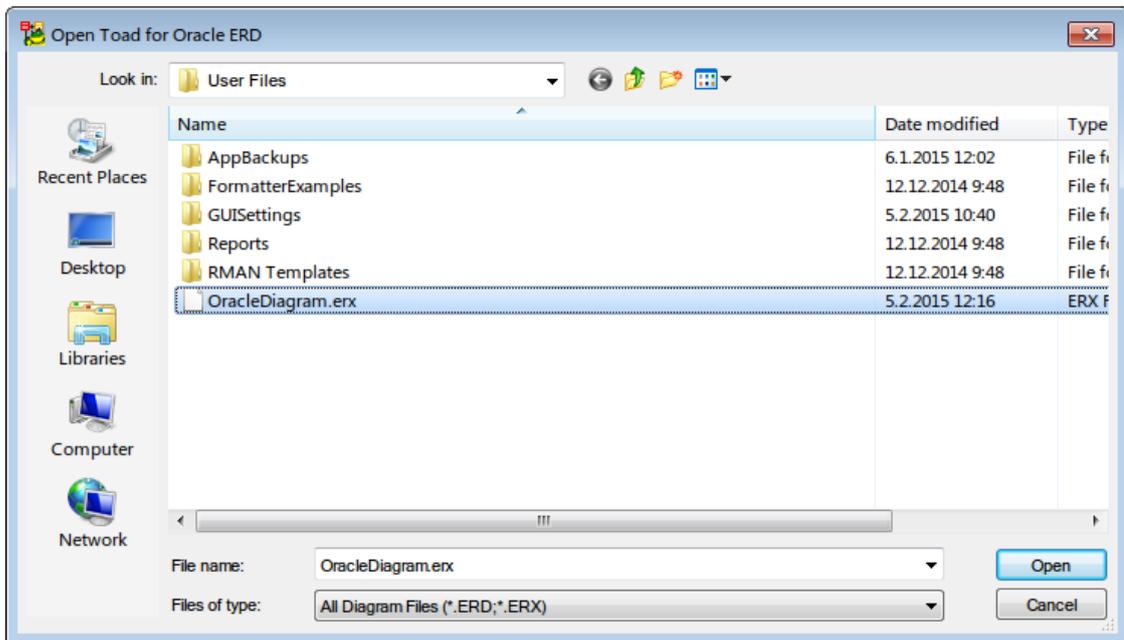
- Both options are available from the **File Menu | Reverse Engineering**.
- Toad Data Modeler allows you to use Toad for Oracle aliases but it doesn't allow you to save changes to these aliases.



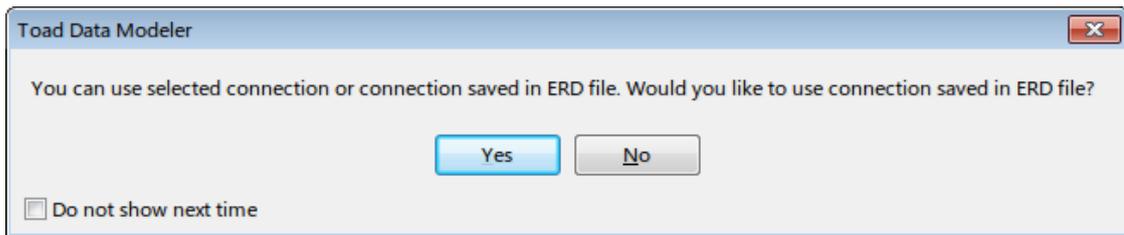
## Import Toad for Oracle® ER Diagrams

1. Select **File | Import | Toad for Oracle ERD** or **File | Reverse Engineering | Connections | Import Toad ERD**.

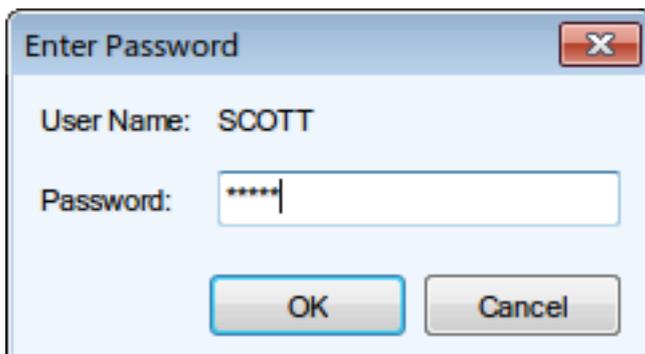
2. In the **Open** dialog, select a \*.erd or \*.erx file and click **Open**.



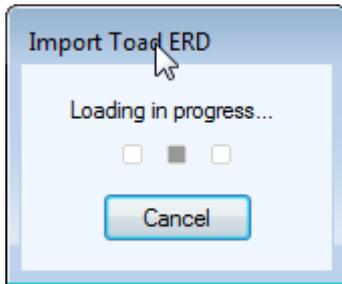
3. If you want to use connection string stored in ERD file, click **Yes** in the following dialog window:



4. Toad Data Modeler needs to reverse engineer database which contains the objects in your diagram. You will be asked for the database password.

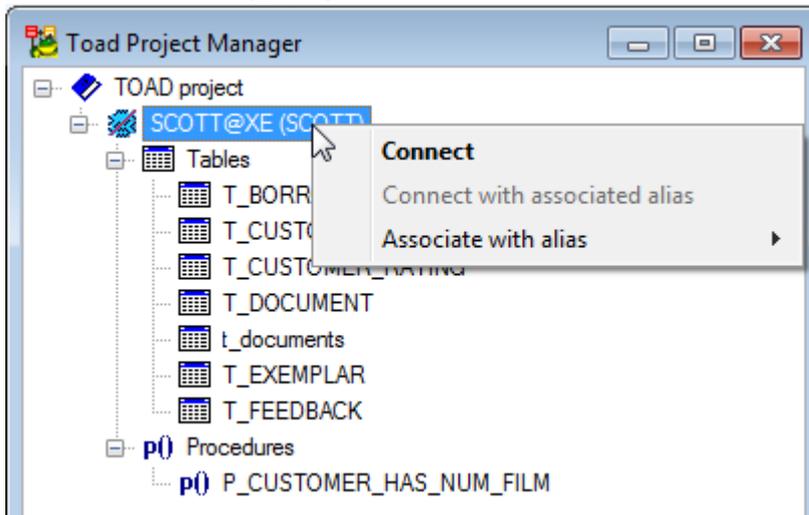


5. The import starts and ERD is created.



## Open Toad for Oracle® Projects

1. Select **File Menu | Import | Open Toad for Oracle Project**.
2. In the **Open** dialog, select a \*.tpr file. **Toad Project Manager** opens.
3. Select a connection type, right-click it and select **Connect**.



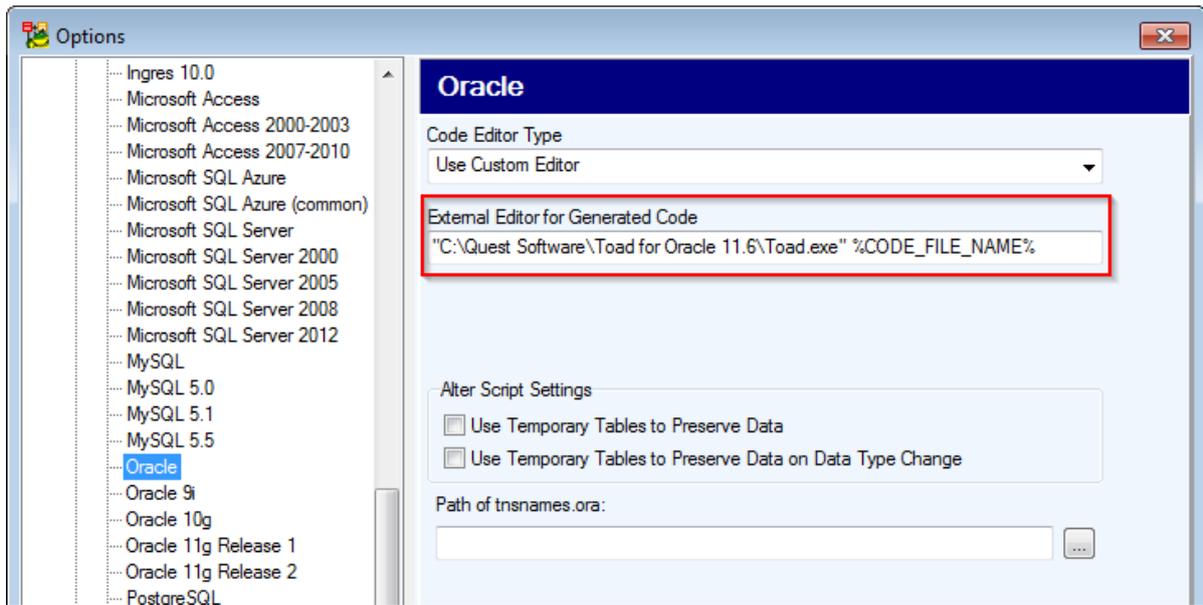
4. After you are connected, you can select items from the Toad Project Manager and drag them to a Workspace or Model Explorer of your model in Toad Data Modeler.
5. Right-click the connection and select **Disconnect** or simply close **Toad Project Manager**.

**i** Note: Tables that already exist in the model cannot be added to the model.

## Toad for Oracle® as Default Editor

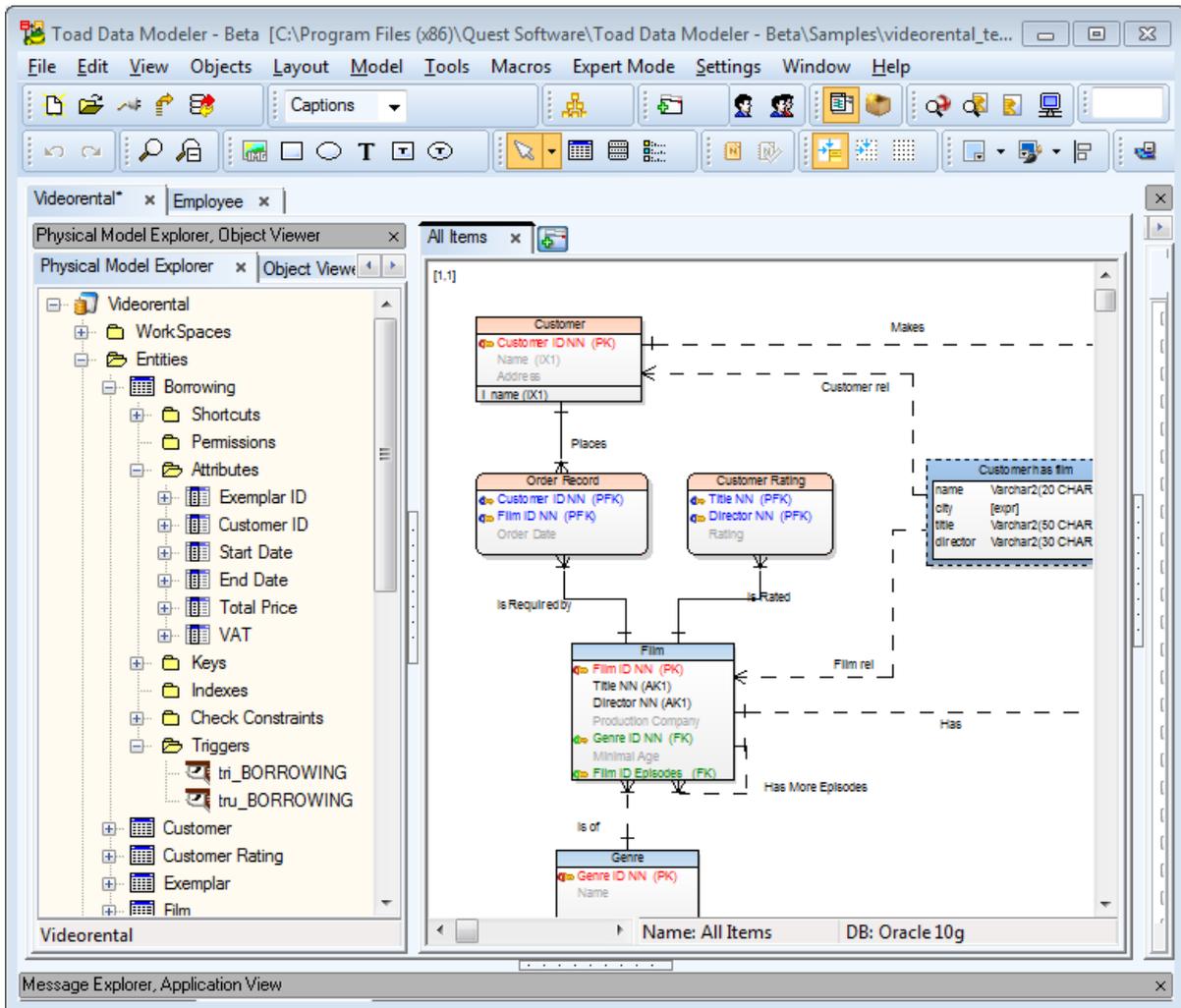
Select Toad for Oracle® as a default editor to open every generated SQL script in Toad for Oracle® (Oracle models only).

You can define any other third party application as default editor.



## Toad® for Oracle® Icons

If you are used to Toad for Oracle icons, Toad Data Modeler gives you the option to change the icon theme. Go to **View Menu | Icons Theme** and select **Toad for Oracle Icons**. Notice the changed icons on toolbars and in Model Explorer.



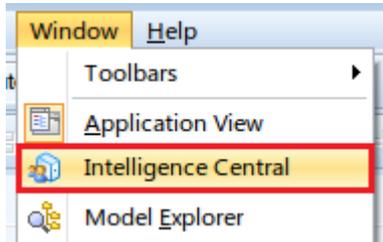
## Basic Actions

With Toad Intelligence Central, you can store and access all of your important files created in various Toad tools in one place. Learn how to:

- [Connect to your TIC server](#)
- [Save files from TIC server](#)
- [Publish your models/projects to TIC server](#)

## Connecting to TIC server

1. Open Intelligence Central using one of the two ways:
  - a. Go to **Window Menu** and choose **Intelligence Central**.



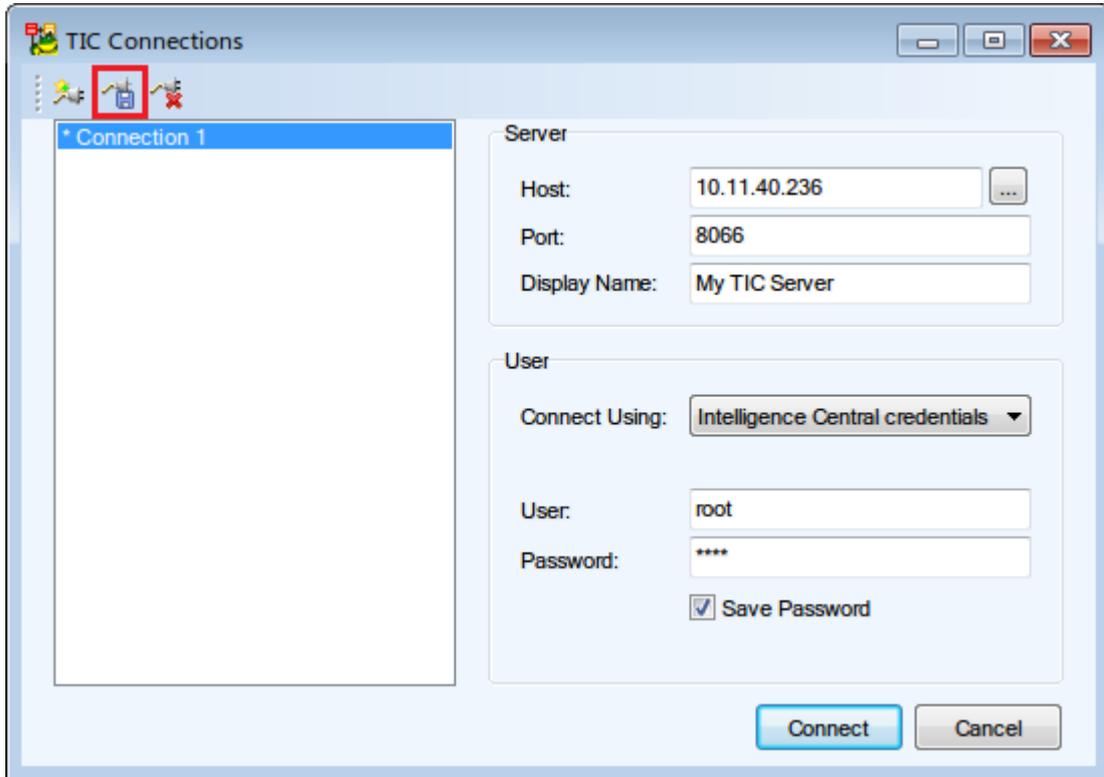
- b. Click the **Intelligence Central** button on **Intelligence Central Toolbar**.



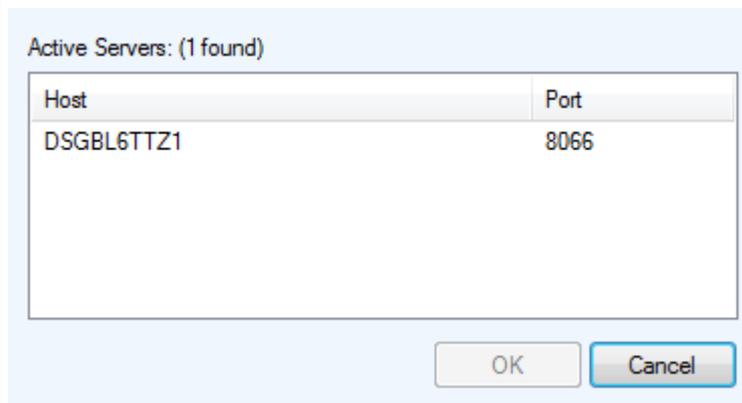
2. Click the **Connect** button in Intelligence Central.



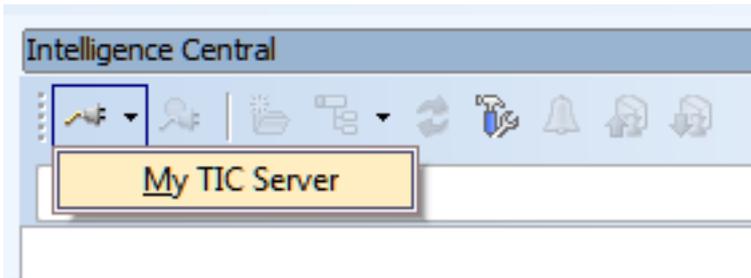
3. In the opened **TIC Connections** dialog, click the **New Connection** button and fill in connection details. Then, click the **Save Connection** button (or you can **Connect** right away, changes will be saved).



- i** Note: If your TIC server is located in the same domain as your computer, you can server discovery to detect its connection details (the ... button next to **Host**).

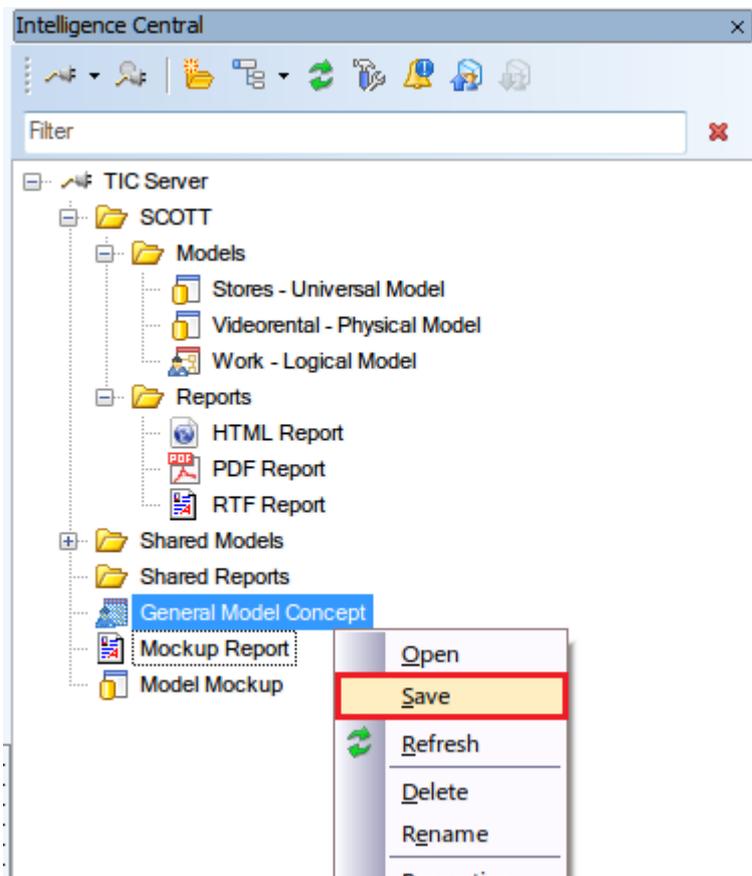


4. Now simply choose your saved Connection from the list in the Intelligence Central.



## Saving files (artifacts) from TIC server

In Intelligence Central, you can save any file you see to your computer. To do so, **right-click** a file, select **Save** and then choose a location on your drive.

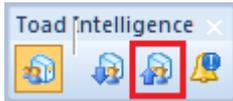


## Publishing files to TIC server

Intelligence Central displays all items that are shared with you. You can publish all types of models and reports to your TIC server:

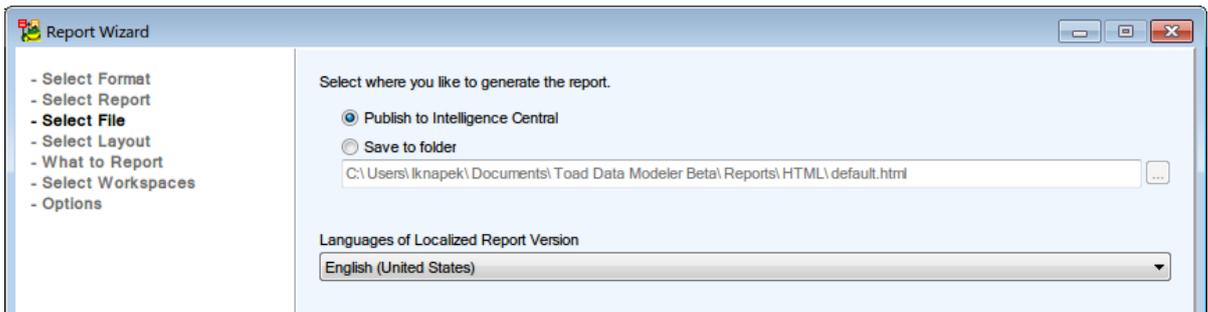
### Models

- Click the **Publish to Intelligence Central** button on **Intelligence Central Toolbar**. See [Publishing Models/Reports](#) for more information.



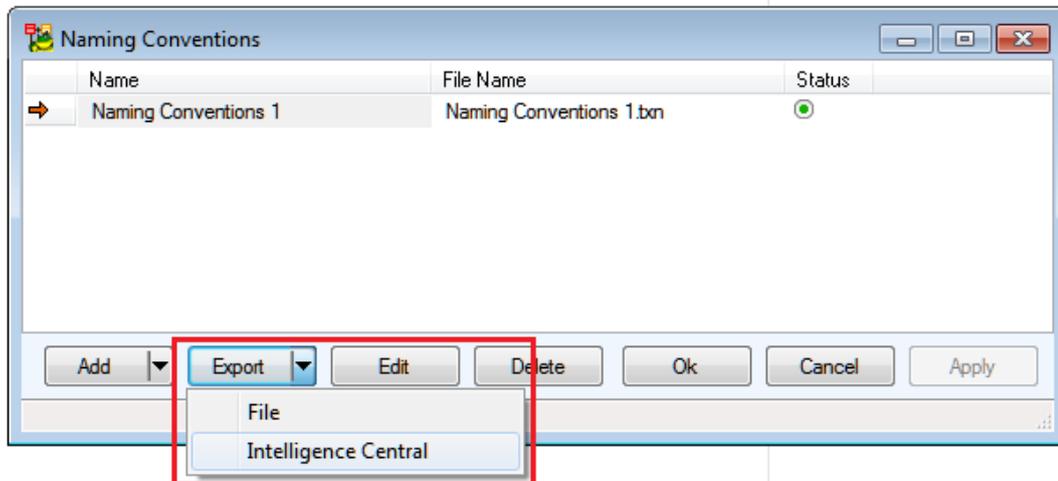
## Reports

- Reports can be published to TIC by checking the appropriate option in the **Select File** section of **Report Wizard**. See [Publishing Models/Reports](#) for more information.



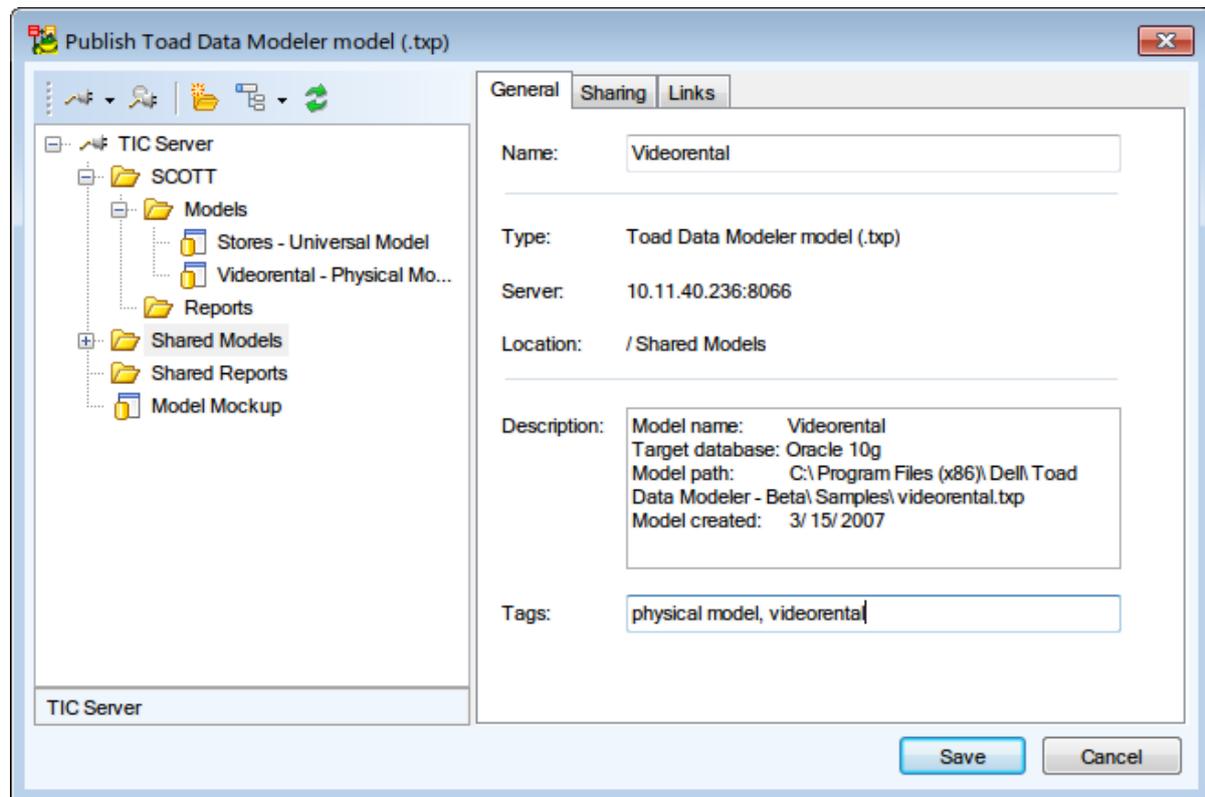
## Naming Conventions

- You can also publish your customized Naming Conventions to a TIC server. Go to **Tools Menu | Naming Convention | Manage**, select your Naming Conventions and choose **Export | Intelligence Central**.



# Publishing Models/Reports

When you choose to publish model/report to your TIC server, you can also describe your file, customize its sharing and link it to existing artifacts.



1. Choose the destination folder in the section on the left. There, you can also see all files of the same type as the file you are going to publish.
2. Look into the **General**, **Sharing** and **Links** tabs and customize the options.

## General tab

Option	Description
Name	The name under which the file will be stored on TIC server.
Type	The file type.
Server	Server host name/IP address.
Location	Folder on the server in which the published file will be stored.
Description	An editable file description
Tags	You can enter several comma separated, searchable tags.

## Sharing tab

You can choose to share your file in several ways:

**i** Note: 'Artifact' refers to a file stored on a TIC server.

- **Do not share this artifact with any other user** - Only you will be able to see and manage the file on TIC server.
- **Share this artifact with any other user** - All users will be able to see the file on TIC server. Additionally, when you check **Allow any user to manage this artifact**, all users will be able to manage the file.
- **Share this artifact with selected user or group** - You can share the file with a specific user/group by moving them to the **Shared with** field. You can also allow users/groups to manage the file by checking the appropriate checkbox.

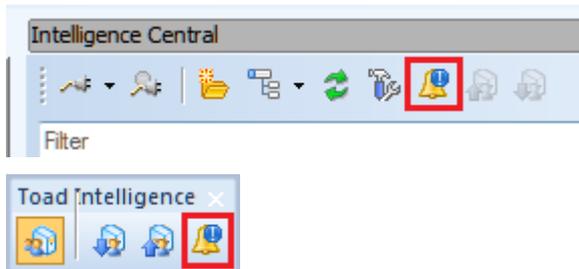
### Links tab

Links represent logical relationships between artifacts. You can create one between your published file and one or more existing artifacts. In **Link Description**, describe the nature of the relationship.

## Notifications

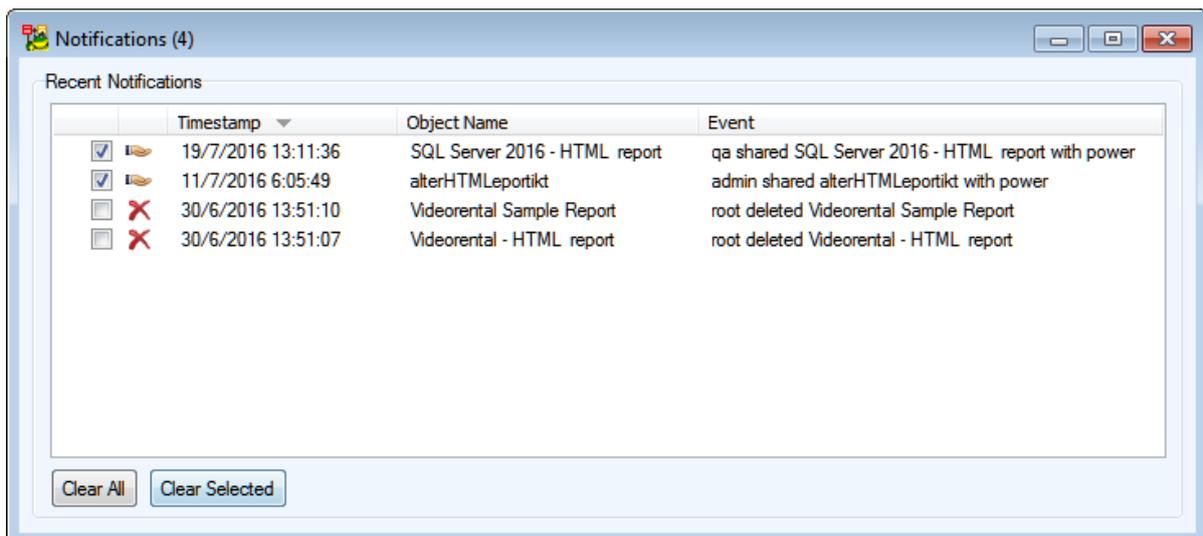
Toad Data Modeler notifies you about actions of other users done on your TIC server. These actions can be found in **Notifications** window.

To open the Notifications window, click the **Notifications** button either in **Intelligence Central window** or on **Intelligence Central toolbar**.



**i** Note: The Notifications icon will have a blue exclamation mark decorator whenever there is a notification that hasn't been reviewed and cleared yet.

Notifications windows displays the timestamp of recent actions, the object on which the action was applied and type of the action (event).



To acknowledge the changes, you can select any or all of the notifications and click **Clear Selected** or **Clear All** to remove them from the list.

## About Naming Conventions

Naming conventions can be defined for **physical models** and used to:

- Define custom rules for **physical names** of objects (e.g. adding prefix, suffix, use particular letter case...).
- Synchronize logical names (captions) and physical names (names) of your objects.
- Verify your model to see if it matches the naming convention rules. If not, there are quick fixes available to solve any issues.

**i** Note: Naming conventions can be only used to alter physical names (**names**) of model objects, NOT logical names (**captions**).

### Basic Information About Naming Conventions

- Naming conventions rules can be set for most physical model objects that have a defines name and a caption.
- Naming conventions are stored as **.txn** files in the Documents folder on local drive (**Documents | Toad Data Modeler | Installation Name | Naming Conventions**).
- Multiple sets of naming conventions can be created, but one model may use only one set at a time.

Naming conventions allow you to:

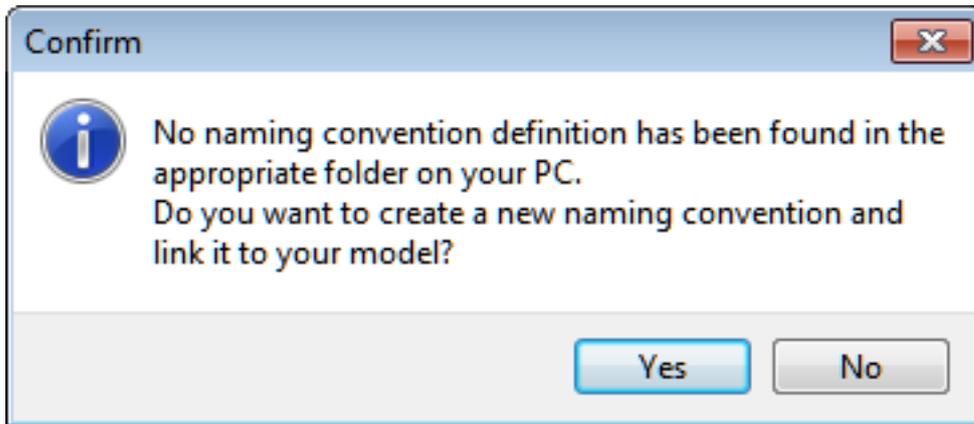
- Automatically create physical names based on captions and set naming conventions rules, e.g. Customer ID (caption) => T\_CUSTOMER\_ID (name).
- Define a set of valid/invalid characters used in physical names (and define how invalid characters should be replaced).
- Verify and automatically update names of multiple objects at once.
- Use glossaries to automatically replace language-specific characters or abbreviations. Glossaries can be exported/imported as CSV files.

**i** TIP: Toad Data Modeler comes with several CSV files to help you replace language-specific characters with English characters. The files are located in **Documents | Toad Data Modeler | Installation Name | Naming Conventions | CSV**.

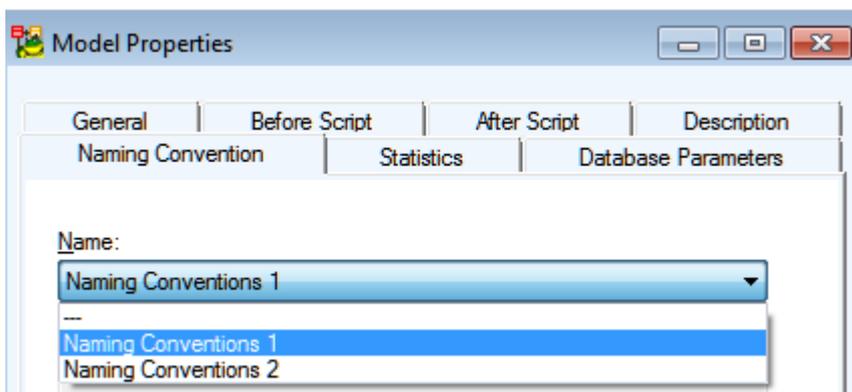
### Managing Naming Conventions

To start using naming conventions, you can go to **Tools Menu | Naming Conventions | Settings** or click  on **Naming Conventions Toolbar**.

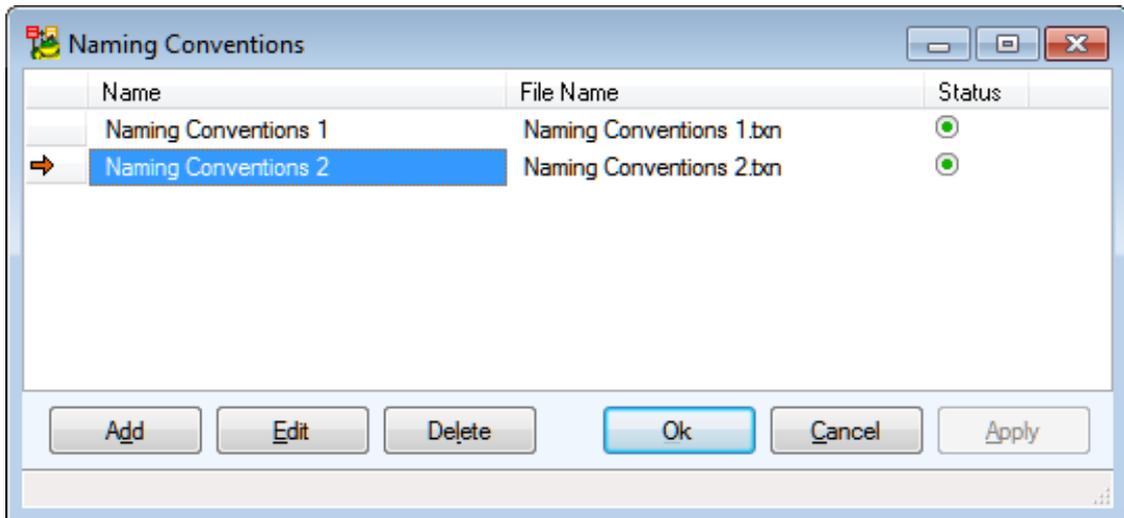
- If no naming conventions are found by Toad Data Modeler, you will be asked to create a new set of naming conventions which will become linked to the currently active model.



- You can link your model to another set of naming conventions in **Model Menu | Properties** (or **Model Properties** in context menu) | **Naming Convention** tab.



- Additional naming conventions can be created and managed in a dialog opened by clicking **Manage Naming Convention** in **Model Properties**.



## Verification and Synchronization

Naming Convention Verification and Synchronization can be accessed from:

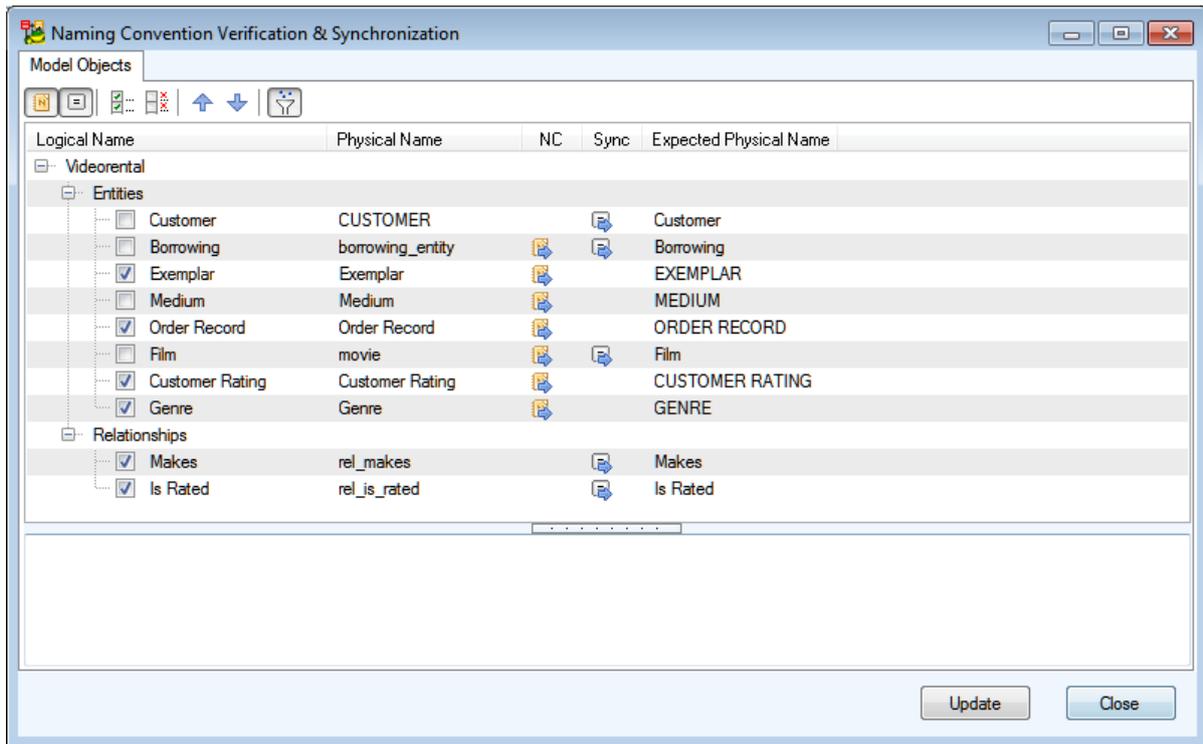
- **Naming Convention Properties | General tab | Verify and Synchronize** button
- **Naming Convention toolbar | Naming Convention Verification and Synchronization**  button

The verification and synchronization process allows you to do two things to multiple objects at once:

- **Verification** - verifies object **Names** to see if they satisfy **Naming Convention Rules**.
- **Synchronization** - synchronizes object **Name** with its **Caption**.

Names that do not pass verification or synchronization check can be automatically updated by applying the suggested Name modification. To update such Names, simply check the listed violations/suggestions and click on **Update**.

**i** Note: The behavior of verification and synchronization process is heavily influenced by **Naming Convention Properties**.



To show/hide **Naming Convention Violations** (  ) or **Synchronization violations** (  ), use the buttons on the toolbar.

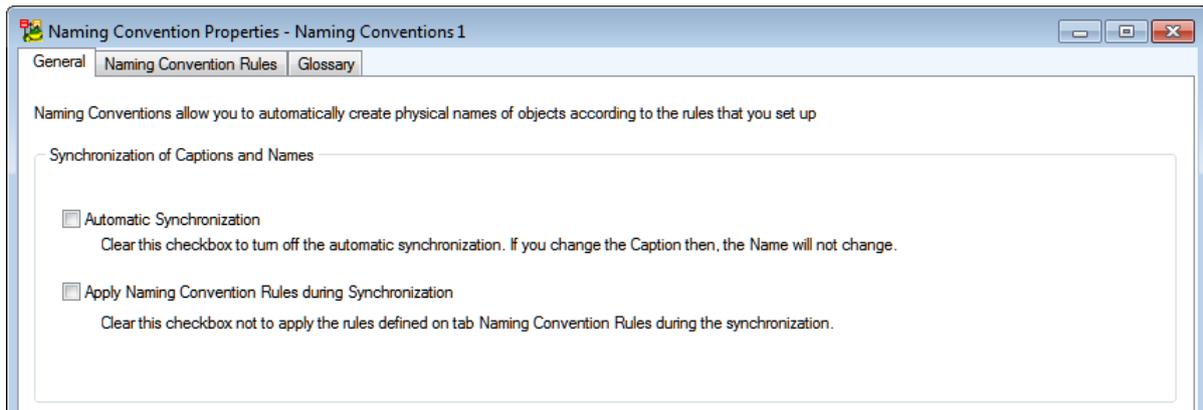
**i** Note: You can disable Verification/Synchronization checking for particular items by clicking the violation (  /  ) icon. This will influence the suggested Name modification.

		Film	movie			Film
		Film	movie			MOVIE
		Film	movie			movie

## On Form Synchronization

Toad Data Modeler offers you the possibility of synchronizing and verifying objects **Name/Caption** in various **Properties dialogs** as you type. This behavior is controlled by two options located in **Naming Convention Properties | General tab**.

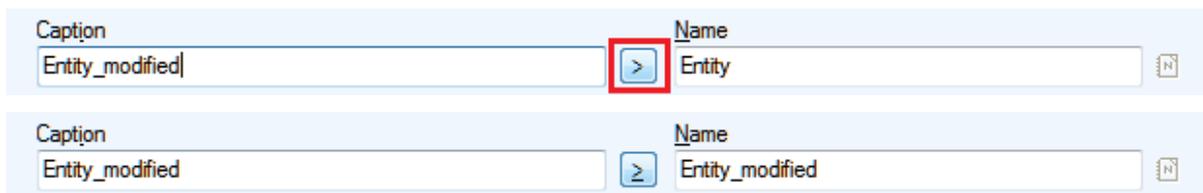
- **Automatic Synchronization**
- **Apply Naming Convention Rules during Synchronization**



### Example 1:

- Automatic Synchronization - **disabled**
- Apply Naming Convention Rules during Synchronization - **disabled**

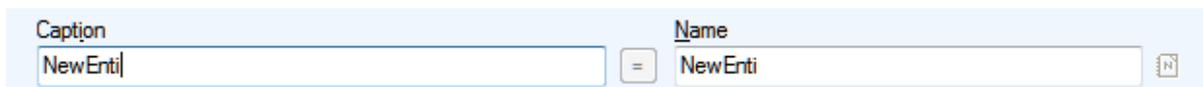
When neither option is enabled, you can change an object **Caption** but its **Name** won't change automatically. To synchronize Name/Caption manually, click the  button.



### Example 2:

- Automatic Synchronization - **enabled**
- Apply Naming Convention Rules during Synchronization - **disabled**

The object **Name** will be changed to match the **Caption** as you type.



### Example 3:

- Automatic Synchronization - **disabled**
- Apply Naming Convention Rules during Synchronization - **enabled**
- Naming convention rule enforces *upper case* and replaces *non-alphanumeric characters* with ? character.
- Glossary contains a defined entry which replaces '*Customer*' with '*CSTMR*'

Similar to Example 1, **Name** won't be automatically updated. Additionally, **naming convention rules** and **glossary replacements** will be applied when performing manual synchronization.

Caption	Customer_Entity	>	Name	Entity1
Caption	Customer_Entity	=	Name	CUSTOMER?ENTITY

### Example 4:

- Automatic Synchronization - **enabled**
- Apply Naming Convention Rules during Synchronization - **enabled**
- Naming convention rule enforces *upper case* and replaces *non-alphanumeric characters* with ? character.
- Glossary contains a defined entry which replaces 'Customer' with 'CSTMR'

The object **Name** will be changed to match the **Caption** as you type, **naming convention rules** and **glossary replacements** will be applied to the **Name** as you type as well.

Caption	Customer_Ent	=	Name	CUSTOMER?ENT
---------	--------------	---	------	--------------

## Naming Convention Properties

In the **Naming Convention Properties** dialog | **General** tab, you can see options for:

- Synchronization of Captions and Names
- Verification of Names

This topic describes how various options affect the **Naming Convention Verification and Synchronization** process.

**i** Note: Options mentioned in this topic also affect **On Form Synchronization** behavior.

## Automatic Synchronization

**Description:** When enabled, **Name/Caption** synchronization will be checked during process.

**Example:** An entity **Caption** is "Example Entity" and its **Name** is "ex\_entity".

A suggestion will be offered to rename the entity Name so it matches its current Caption - "Example Entity".

## Apply Naming Convention Rules During Synchronization

**Description:** When enabled, **Naming Convention Rules** will be applied and characters/words will be replaced according to **Glossary** during **Name/Caption** synchronization.

**Example:** An attribute **Caption** is "Customer main ID" and its **Name** is "Customer ID".

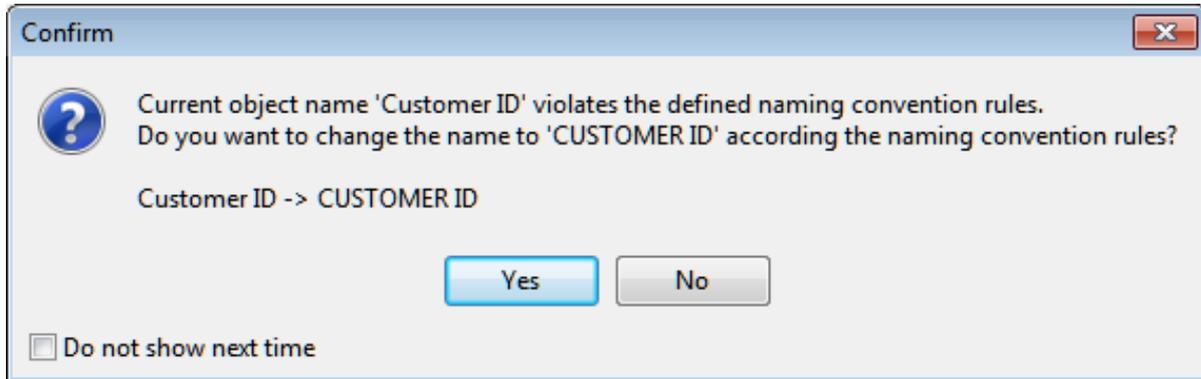
- A **Naming Convention Rule** defined for attributes enforces **upper case**.
- A **word replacement** in **Glossary** replaces "Customer" with "CSTMR".

**Automatic Synchronization enabled:** The suggested Name will be synchronized with the object Caption and then have the **NC Rule** and **Glossary word replacement** applied to it - *CSTMR MAIN ID*.

**Automatic Synchronization disabled:** The option only affects **manual synchronization** (see [On Form Synchronization](#) for more information).

## Automatic Verification

**Description:** When enabled, model objects will be automatically verified to see if they don't violate **Naming Convention Rules**. The verification will occur during **Naming Convention Verification and Synchronization** process and also when you open **Properties** dialog of any object.



**Example:** A relationship **Name** is "ASSIGNS\_orders".

- A **Naming Convention Rule** defined for relationships considers *underscore characters invalid* and **replaces** them with *space characters*.
- A **Naming Convention Rule** defined for relationships enforces **lower case**.

A suggestion will be offered to modify the Name so it matches the rules - "*assigns orders*".

### To Use Glossary for Naming Convention Verification

**Description:** When enabled, **Glossary** character and word replacements will be applied during both **Automatic Verification** and manual verification.

**Example:** An entity **Name** is "*Entrée kinds*".

- A **Naming Convention Rule** defined for entities enforces **upper case**.
- A word replacement defined in **Glossary** replaces "*kinds*" with "*types*".
- A character replacement defined in **Glossary** replaces "*é*" with "*e*".

A suggestion will be offered to modify the Name so it matches the rule and has the defined characters/words replaced - "*ENTREE TYPES*".

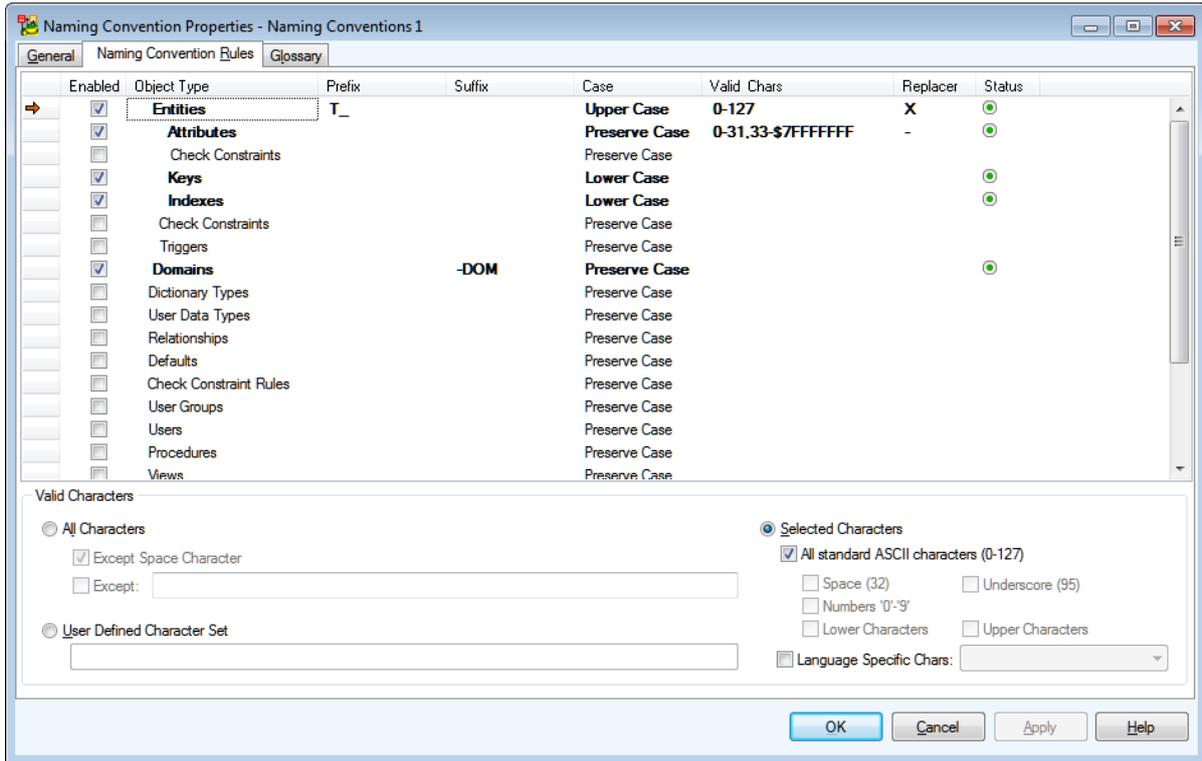
### To Turn Case-Sensitivity On

- Allow **Case Sensitivity** in **Tools | Naming Convention | Settings | Glossary | Word Replacement**

## Naming Convention Rules

Naming convention rules can be defines in **Naming Convention Properties (Tools Menu | Naming Conventions | Settings)** or click  on **Naming Conventions Toolbar**.)

**i** Note: Click the particular column and use F2 to turn on the edit mode. Press **Enter** to confirm every rule you define in the form.



Option	Description
Enabled	Enables set rule for particular Object Type.
Prefix	Defines a prefix for selected Object Type.
Suffix	Defines a suffix for selected Object Type.
Case	You can define Upper/Lower Case as a rule for object names.
Valid Chars	Lists characters that can be used in object name. The list can be configured in <b>Valid Characters</b> section.
Replacer	If set, replaces all invalid characters during verification and update.

## Valid Characters

You can define a set of characters that can be used in object names. Characters outside the defined set will be considered invalid and will be replaced during verification and update process (either by **Replacer**, if defined, otherwise by a blank space).

To define valid characters, choose one of the three options.

### All characters

All possible characters will be considered valid. You can choose to except space or any other characters. See **Character Set Syntax** for more information on how to specify a set of characters.

## User Defined Character Set

Only characters defined in this set will be considered valid. See **Character Set Syntax** for more information on how to specify a set of characters.

## Selected Characters

You can choose categories of characters that will be considered valid. **Language Specific Characters** can be also included, simply select one of the option from the menu. If you language is not present, you can create your own CSV file with characters specific to your language, see files in C:\Program Files (x86)\Quest Software\Toad Data Modeler\Naming Conventions\CSV as an example.

## Character Set Syntax

Example: Character set for letters a,b,c and numbers 0,1,2:

- Single character in single quote, divided by comma: 'a', 'b', 'c', '0', '1', '2'
- Single character's ordinal value divided by comma: 97,98,99,48,49,50
- Single character's ordinal value divided by comma, in hexadecimal format: \$61, \$62, \$63, \$30, \$31, \$32
- Set of characters in single quote: 'a-'c', '0-'2'
- Set of characters as ordinal value: 97-99, 48-50
- Set of characters as ordinal value in hexadecimal format: \$61-\$63, \$30-\$32
- Any combination: 'a'-\$63, \$30-49, '2'

## Naming Convention Valid Characters

Naming Conventions also allow you to define valid and invalid characters.

On **Naming Convention Rules** tab you can define valid characters for physical names and also set how invalid characters should be replaced.

**Example:** Let's say you have set a space as invalid character for entity names in your naming convention. See how Toad Data Modeler will behave in the following situation:

*Customer Data* logical name in the **Caption** box, will automatically change to *CustomerData* physical name in the **Name** box. -> The space will be ignored.

You edit the naming convention and define that every space should be replaced with '\_'.

*Customer Data* logical name in the **Caption** box, will change to *Customer\_Data* physical name in the **Name** box.

Once you manually edit the physical name, the automatic synchronization will turn off.

## Valid Characters and Character/Word Replacement

# Character Replacement

Use Character Replacement to replace diacritical characters. Do not use Character Replacement to replace a space with another character.

# Word Replacement

Use Word Replacement to replace **one** word with another word or more words. Word Replacement launches after characters are replaced.

## Valid Characters + Replacement

Use this combination to define valid characters. Also, it is possible to replace invalid characters with an alternative character. Check of valid characters launches as last. See the example below.

### **Example:**

Character Replacement:

ñ -> n

Word Replacement:

espanoles -> esp

residentes -> res

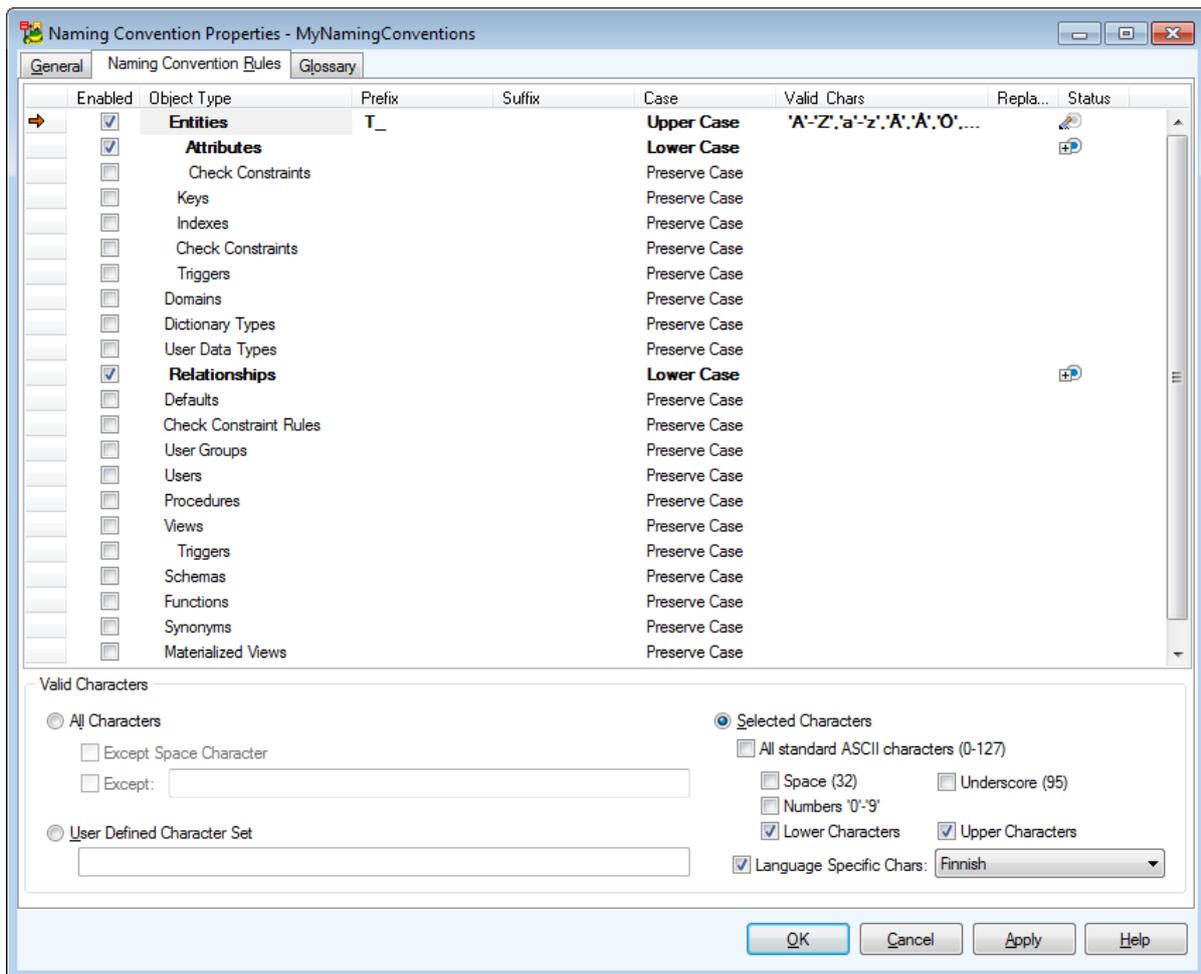
Valid Characters:

Space -> \_

Toad Data Modeler proceeds in the following order:

- 1) "residentes españoles" -> "residentes espanoles"
- 2) "residentes espanoles" -> "res esp"
- 3) "res esp" -> "res\_esp"

## Valid Characters Definition



Option	Description
<b>All Characters</b>	
Except Space Character	If it is selected, all characters are valid except for a space.
Except	Click this checkbox and manually write out character set that you want to forbid. For more information, see <a href="#">Character Set Syntax</a> on page 607.
<b>User Defined Character Set</b>	
	Select this option to manually define character set. Incorrect definition will be highlighted in red.
	<b>i</b> Note: Valid character set is automatically written out in the <b>Valid Chars</b> column.
<b>Selected Characters</b>	
	Select any of the following options to set:
All Standard ASCII	- all standard ASCII characters valid, others are forbidden.

Option	Description
Characters (0-127)	
Space (32)	- only Space (32) character as valid.  <div style="border-left: 1px solid black; padding-left: 5px;"> <p><b>i</b> Note: Number 32 is an ordinal value of the Space character.</p> </div>
Number '0'-'9'	- number 0 – 9 as valid.
Lower Characters	- only lower characters as valid.
Underscore (95)	- Underscore 95 as valid. (Number 95 is an ordinal value of the Underscore character.)
Upper Characters	- only upper characters as valid.
Language Specific Chars	- a diacritical character set of the particular language as valid. Click the box on the right to select the language.  <div style="border-left: 1px solid black; padding-left: 5px;"> <p><b>i</b> TIP: You can create your own table with diacritical character set for your language. See: C:\Program Files\Quest Software\Toad Data Modeler 3\Naming Conventions\CSV. Create your CSV file in this location.</p> </div>
<b>Valid Chars</b> Column	In this column, a complete character set is written out – according to your settings in the <b>Valid Characters</b> area.  <div style="border-left: 1px solid black; padding-left: 5px;"> <p><b>i</b> Note: You can also define the character set manually to this column – via F2 key. However, to make sure your character set is correct, please use the <b>User Defined Character Set</b> option.</p> </div>
<b>Replacer</b> Column	Define a character that will be used as a replacer. Example: underscore character.

## Character Set Syntax

Example: Character set for letters a,b,c and numbers 0,1,2:

- Single character in single quote, divided by comma: 'a', 'b', 'c', '0', '1', '2'
- Single character's ordinal value divided by comma: 97,98,99,48,49,50
- Single character's ordinal value divided by comma, in hexadecimal format: \$61, \$62, \$63, \$30, \$31, \$32
- Set of characters in single quote: 'a'-'c', '0'-'2'
- Set of characters as ordinal value: 97-99, 48-50
- Set of characters as ordinal value in hexadecimal format: \$61-\$63, \$30-\$32
- Any combination: 'a'-\$63, \$30-49, '2'

## Glossary

Glossary has two main functions:

- Character replacement

**i** Note: This function serves primarily to replace various language-specific characters with standard English ones. A replacement is invalid (red) if:

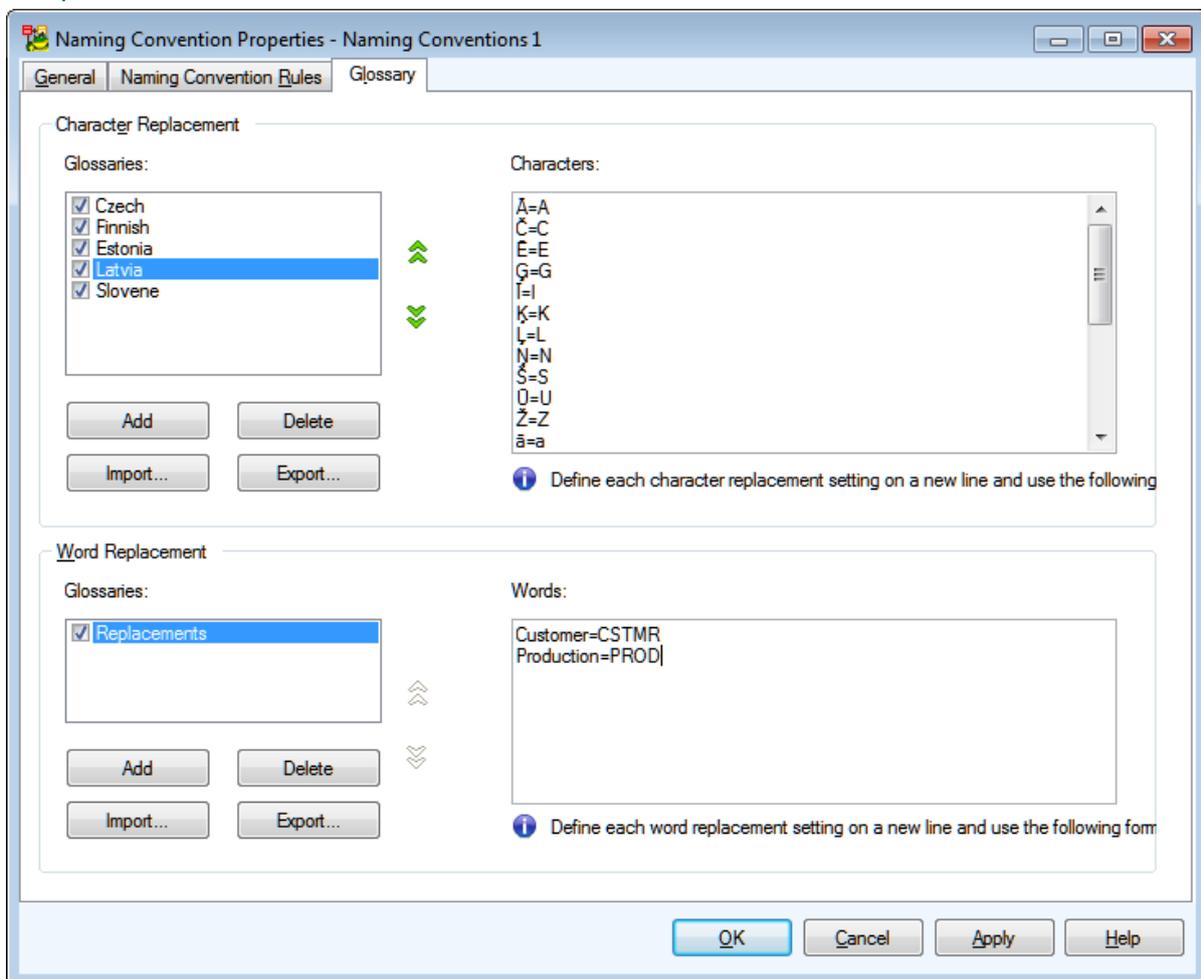
- The left or the right side contains more than one character
- The left side is equal to the right side
- The left side character is an ASCII character (has ASCII value  $\leq 128$ , except for space character)

- Word replacement

In order to replace characters using glossary, you have to define one first. Simply **Add** a new glossary in **Naming Convention Properties | Glossary tab** and start defining your custom replacements. Another option is to **Import** glossary from a CSV file.

**i** Note: Toad Data Modeler includes several existing glossaries used for character replacement. They can be found in:

**C:\Users\\*username\*\Documents\Toad Data Modeler\Standard Installation\Naming Conventions\CSV**



To use glossary during name verification, make sure to enable the options **Automatic Verification** and **Use Glossary for Naming Convention Verification** are enabled.

## To Turn Case-Sensitivity On

- Allow **Case Sensitivity** in **Tools | Naming Convention | Settings | Glossary | Word Replacement**

## To Set Delimiters

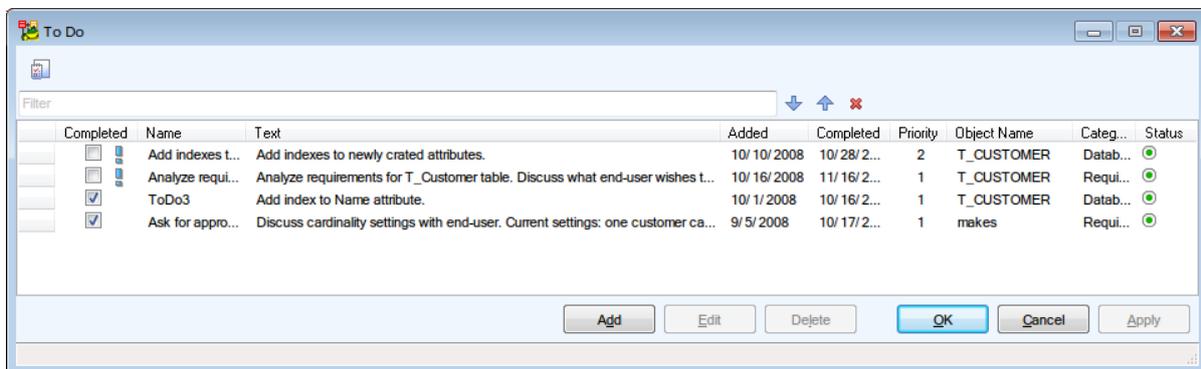
- Select from space, underscore, dot, colon or define your own delimiter

# To Do List

To-Do List allows you to keep records of tasks and make notes on unfinished actions.

You can assign tasks to:

- Particular object of your model in its Properties dialog (see the **Entity Properties** form | **To Do** tab)
- Main **To Do** dialog - see the **Model** menu | **To Do** (A complete list of all To Do items can be found here.)



- Hides/shows To Do items that were entered out of the **To Do** dialog.

In Toad Data Modeler, you can assign tasks to the following objects:

Model, Entity, Relationship, Attributes, Keys, Indexes, Check constraints, Triggers, Users, User groups, Dictionary Types, User Data Types, Domains, Defaults, Rules, Views, Procedures, Schemas, Categories, Metamodels.

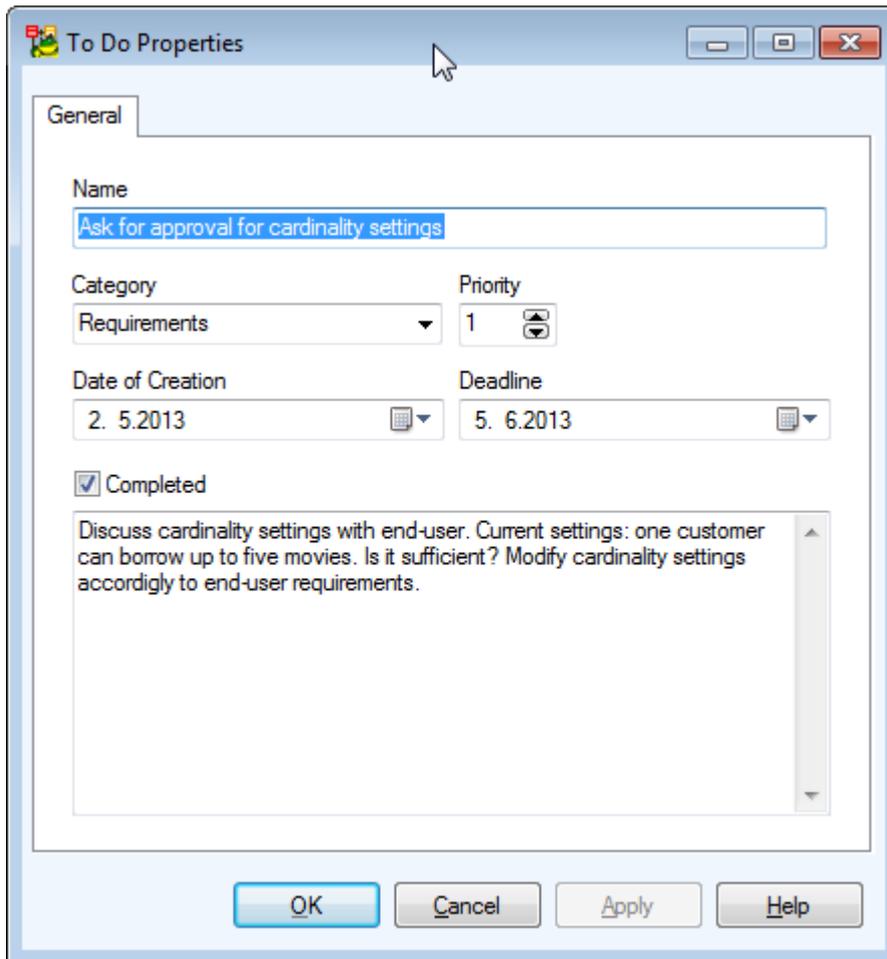
**i** | Note: Toad Data Modeler allows you to generate a To Do report. [XSL Transformation](#)

## To add a new To Do item

Select **Model | To Do | Add**.

## To edit a To Do item

Select **Model | To Do** | double-click the selected item.

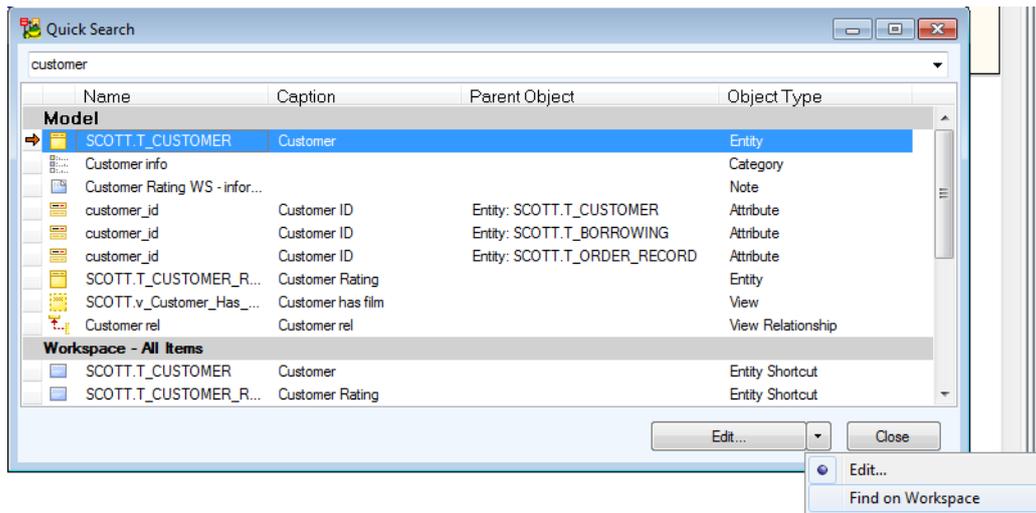


General Tab	Description
Name	Task name
Priority	Task priority
Category	To logically divide your To Do items, you can define categories for them. Simply write a category name to the <b>Category</b> box. The existing categories will be available here via a drop-down menu for other To Do items as well.
Date of Creation	Date when the task has been entered to To Do.
Deadline	Date when the task should be accomplished.
Completed	If selected, the task has already been accomplished.
Text	On this tab, you can write a text description on particular task.

# Quick Search

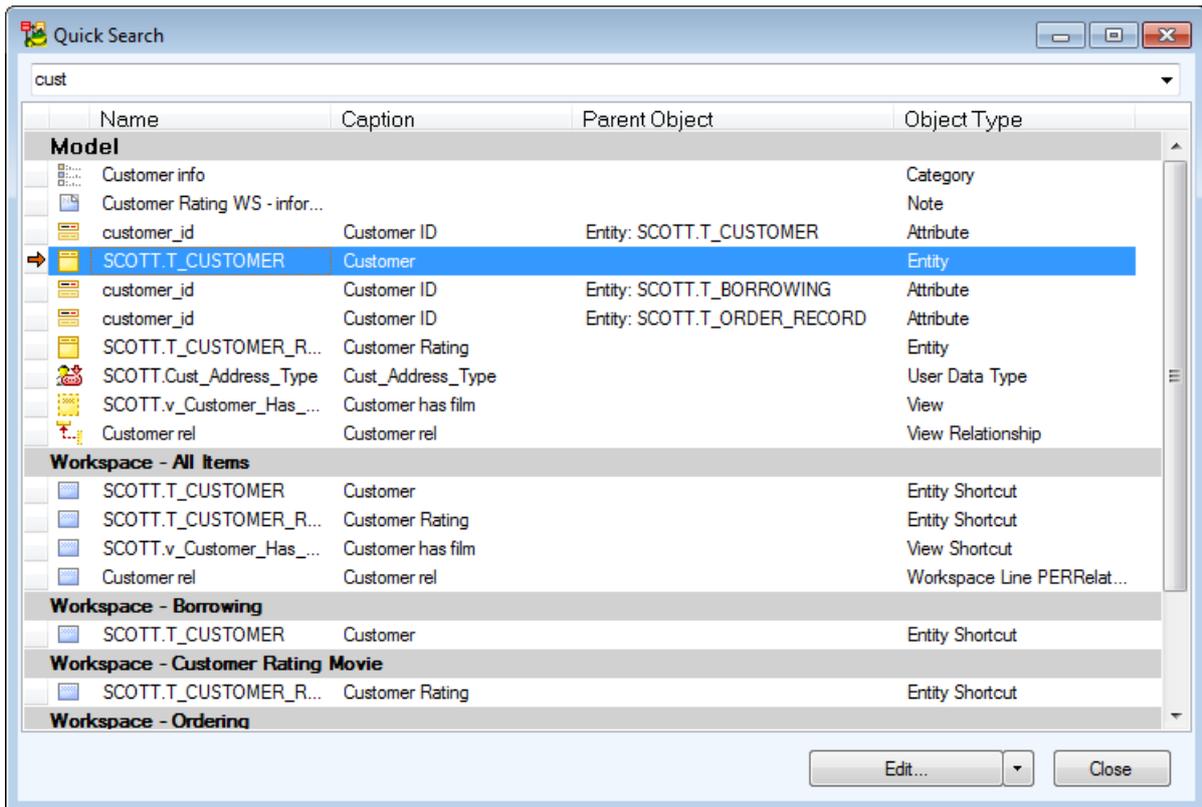
To quickly find an object in your model or on your workspace:

1. Press **CTRL + F**.
2. Type an object name or caption, or part of object name or caption.
3. Click the object or use the arrow down key on your keyboard to select it.
4. Press **Enter** to perform suggested action. Select the suggested action via the **Arrow** button at the bottom of the dialog. Possible actions (depend on the selected item): **Edit**, **Find on Workspace**, **Format**.



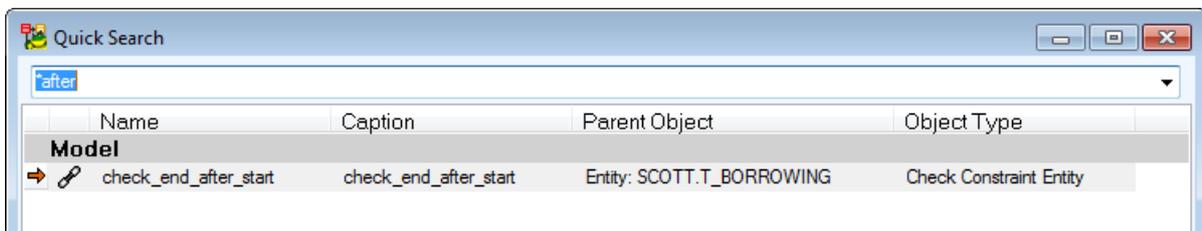
# Standard Search

When you search for an object, write part of the name or caption to the first field.



## Wildcards

Available wildcards are \* and ? characters. (The star wildcard at the end of search term is not required).

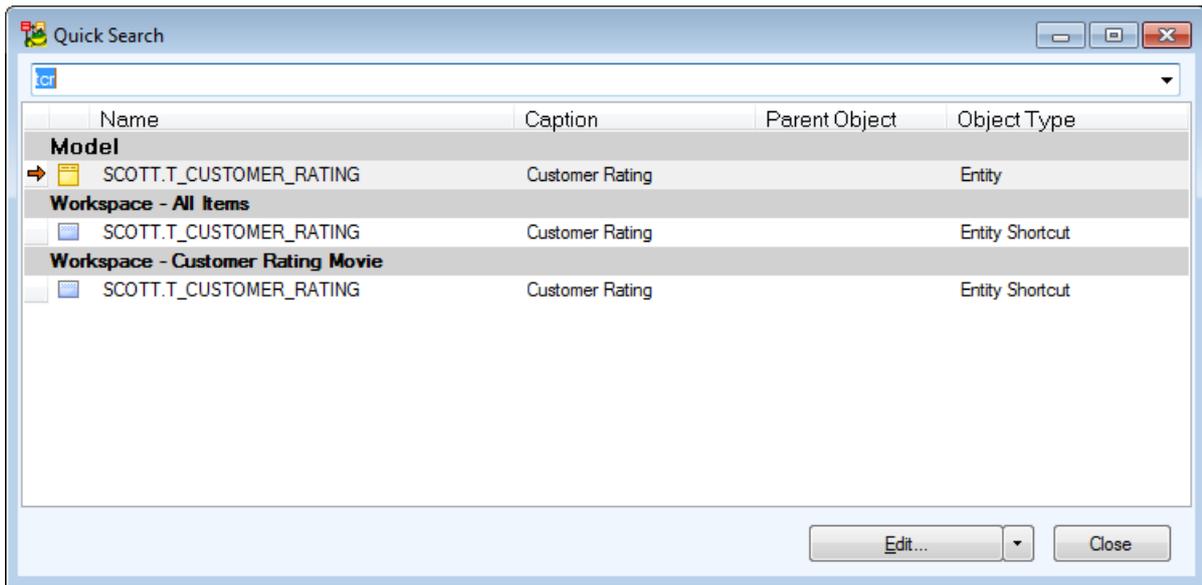


## Word Recognition

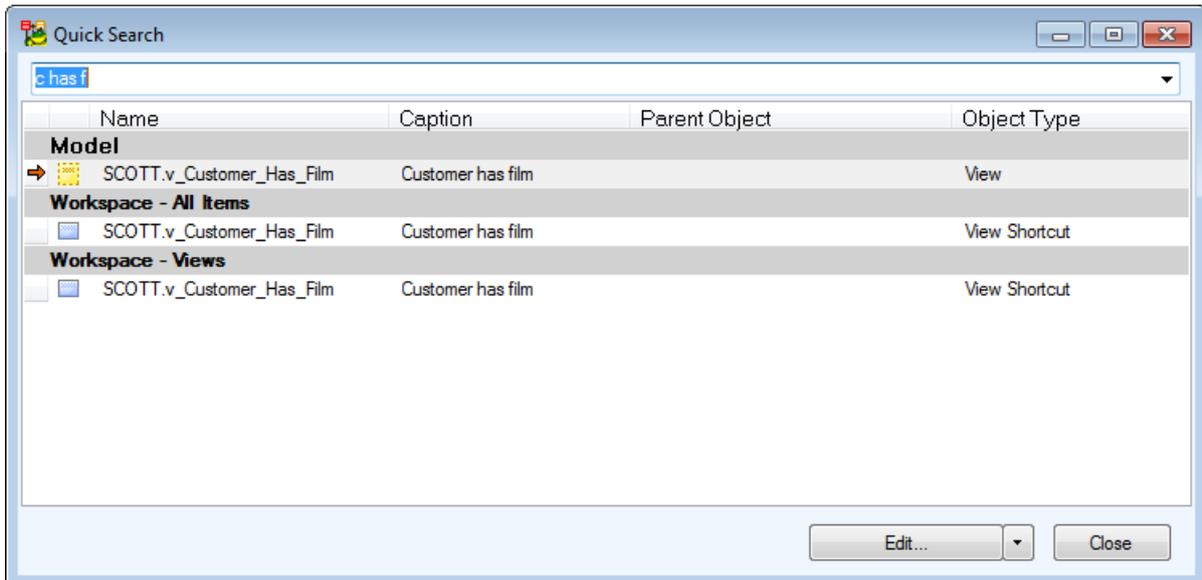
If you use underscores in names, you can type just first characters of the words to refine search.

The following example shows search results of string **tcr**. Results include **T\_CUSTOMER\_RATING**.

Underscore characters were used as word delimiter.

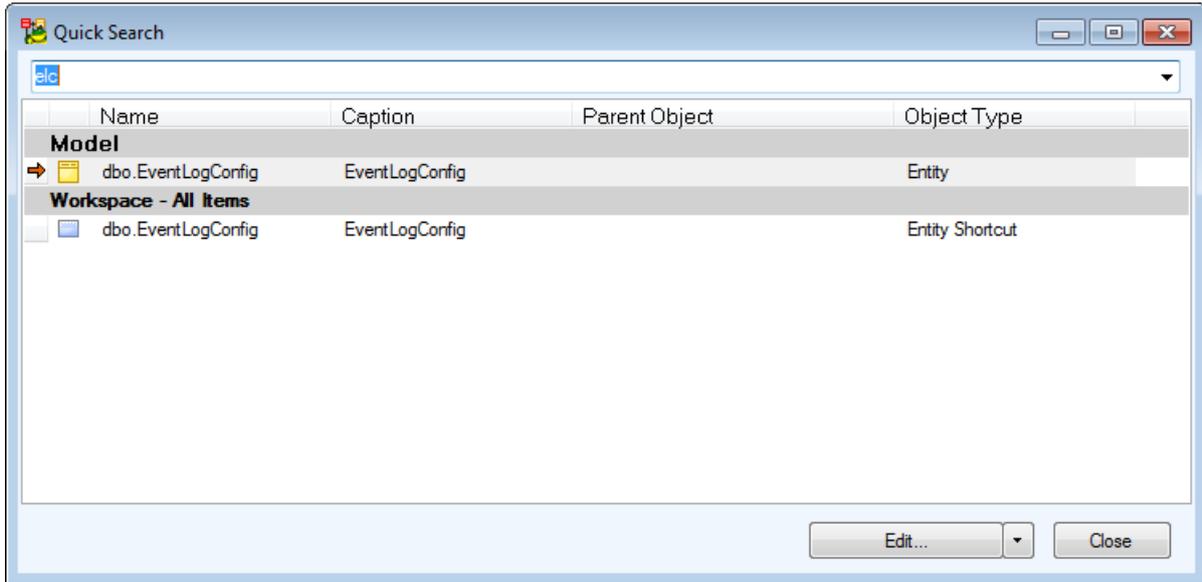


Similar functionality is available for search in captions where **space** is used as word delimiter. You can partially combine the standard search with the word recognition based search.



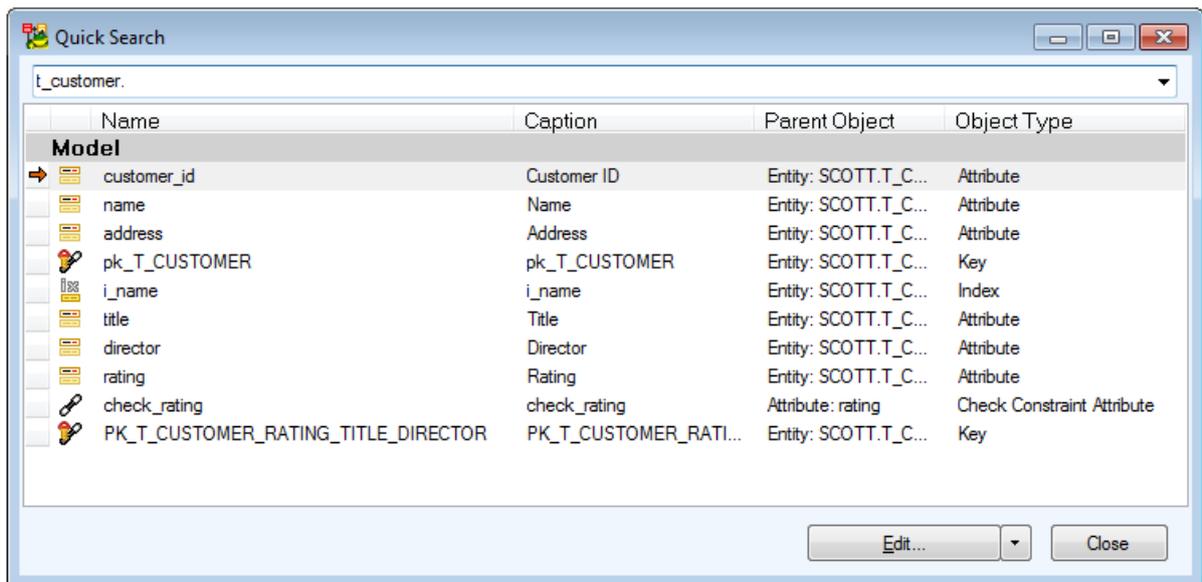
## Camel Case

In case your objects are defined using CamelCase, type just the characters you expect to be in uppercase. Example: To find **EventLogConfig**, search for **elc**.



## Dot Notation

Define object name and type . character to display child items of the object. The dot notation works for schemas as well.



## Gallery

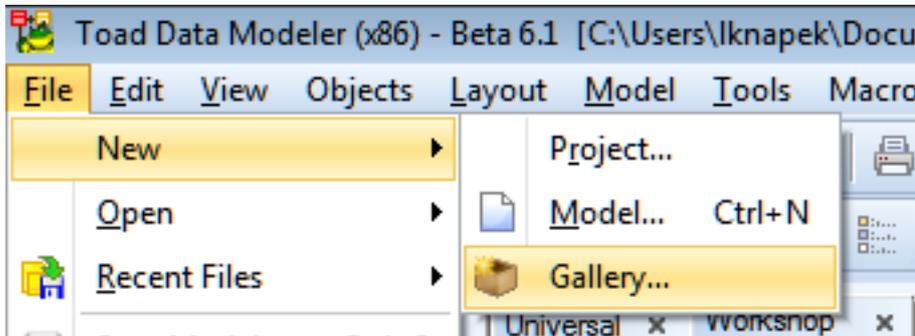
Toad Data Modeler allows you to reuse frequently used parts of your model, no matter if they are entities, single attributes, stored procedures or other objects. This can be done using Gallery where you can simply drag and drop items from your models and use them in your other models.

Some basic Gallery actions include:

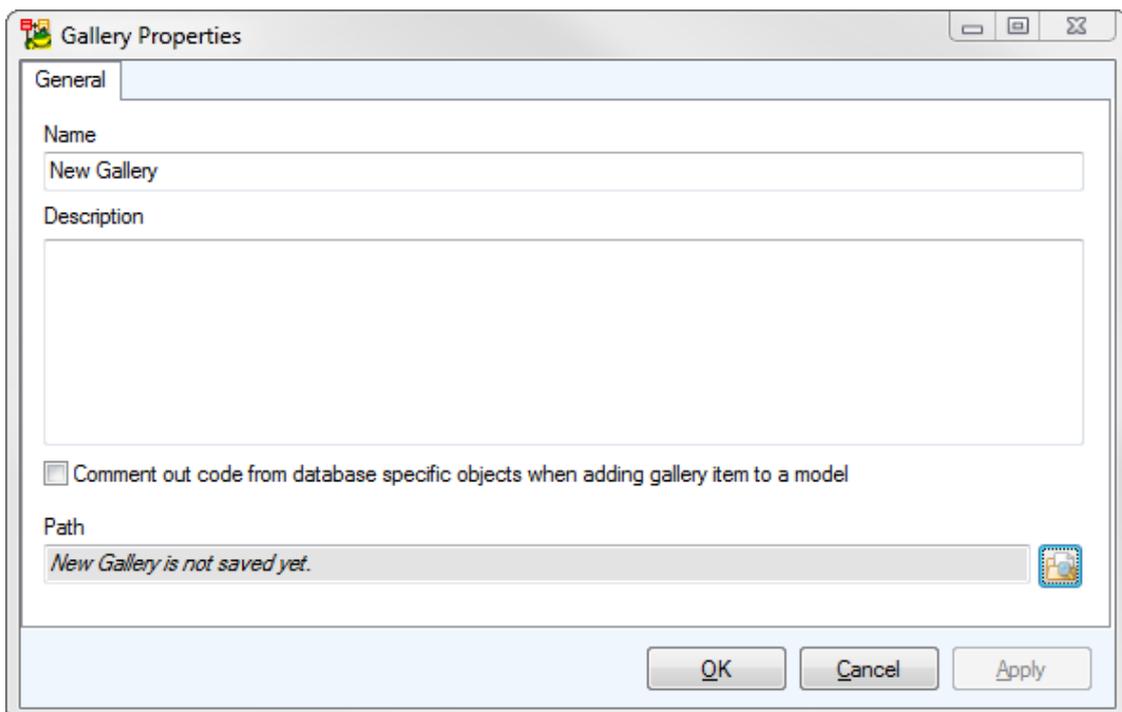
- Creating a new Gallery
- Adding objects to a Gallery
- Inserting objects to a Model

## Creating a new Gallery

1. Go to **File Menu** and select **New | Gallery**.



2. Choose the **Gallery Name**, enter its **Description** and choose whether you want to comment out code of database specific objects (this option is explained in the **Inserting objects...** section).

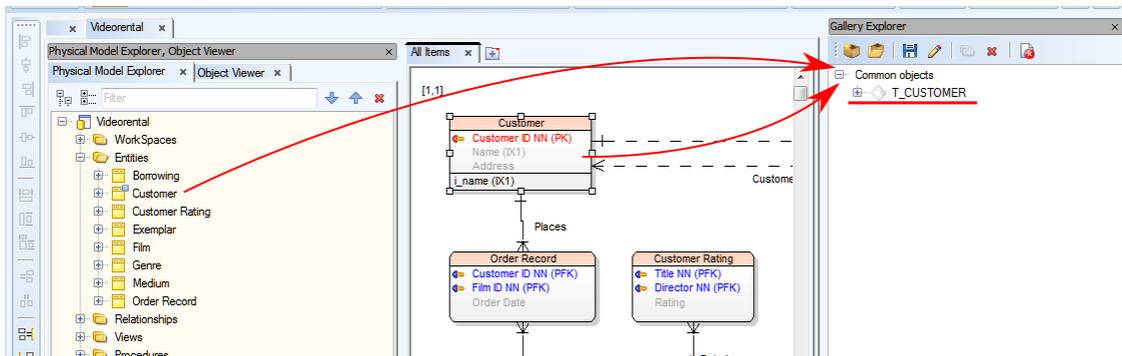


3. Your Gallery will be opened in the **Gallery Explorer**.

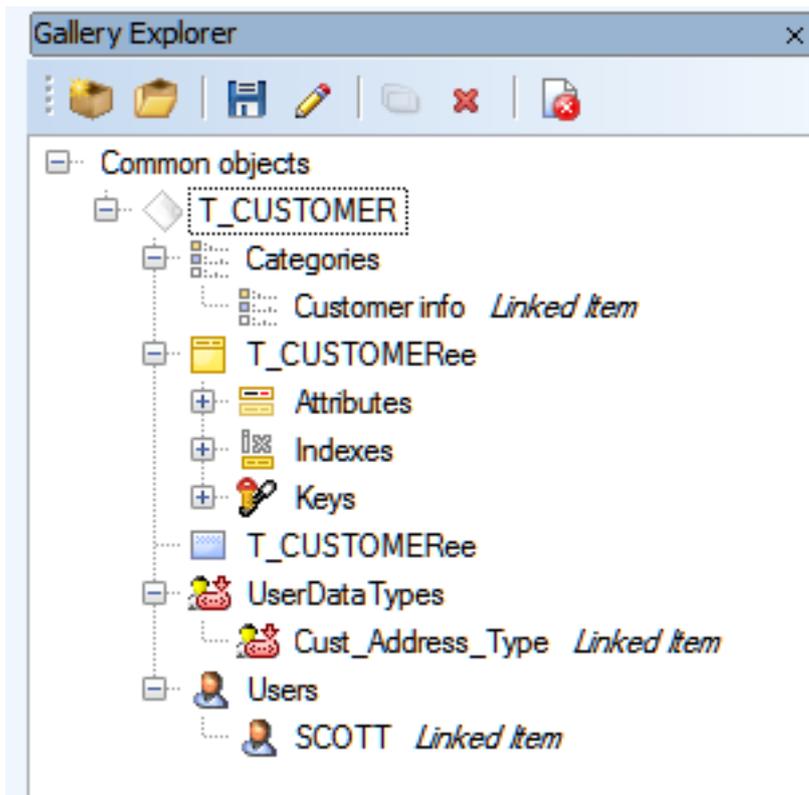


## Adding new objects to a Gallery

1. To add a new object to a Gallery, simply drag it from **Workspace** or **Model Explorer** to and drop it on the Gallery name in **Gallery Explorer**.

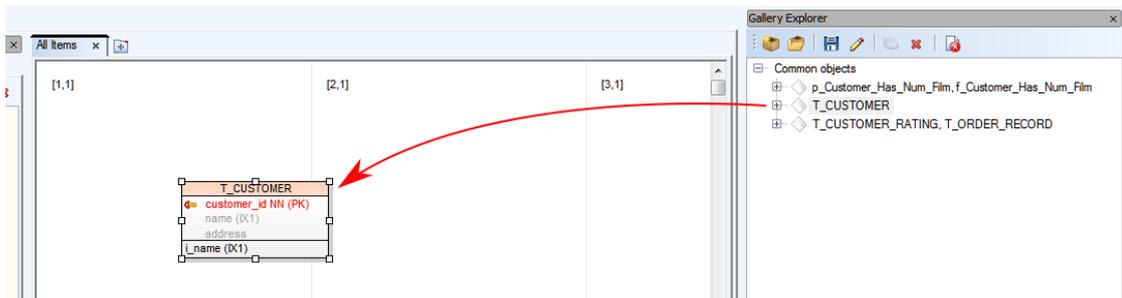


- Your object is now part of the Gallery and you can view its properties by expanding it.



## Inserting objects to a Model

- If you have one or more objects stored in Gallery, you can insert them into a model by simply dragging them to workspace.

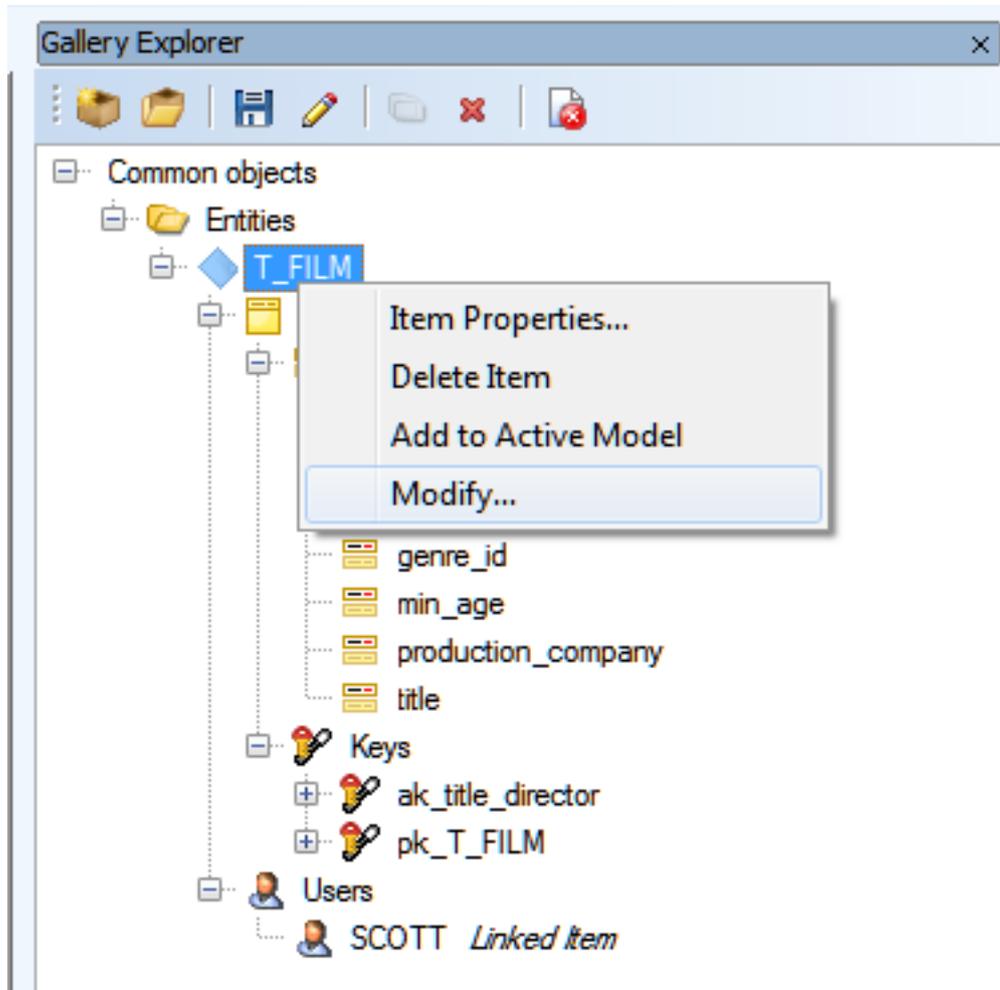


- i** Note: If the model database of the object is different from the target model database (e.g. Entity created in Oracle 10g model dragged to SQL Server 2010 model), it will be converted. Note that database-specific properties (mainly SQL code) may be syntactically invalid in the target database. If you have enabled the **Comment out code from database specific objects...** option, the properties will be commented out and you can fix their syntax later.
- i** Note: You have to drag and drop the main node of the object in the Gallery to actually move the object, not any of its child nodes.

## Modifying Items

Items in Gallery can be easily modified in a temporary model and the changes can be saved immediately.

1. In **Gallery Explorer**, open your **Gallery**, **right-click** the item(s) you wish to modify and select **Modify....**



2. The item(s) will be placed into a temporary model and you will be able to modify them. The exact way to modify the items varies depending on the number of objects and whether they are shortcuts or not:
  - **One object (not shortcut)** - the **Properties** dialog of the objects will be opened right away.  
**Example** - Entity added to a Gallery by dragging it from **Model Explorer**.
  - **One object (a shortcut)** - the object will be shown on workspace, you can view and change its **Properties**.  
**Example** - Entity added to a Gallery by dragging it from **Workspace**.
  - **Multiple objects (not shortcuts)** - no Properties dialog or Workspace will be shown, you will have to open Properties by locating the object in Model Explorer.  
**Example** - Two entities added to a Gallery at once by dragging them from **Model Explorer**.

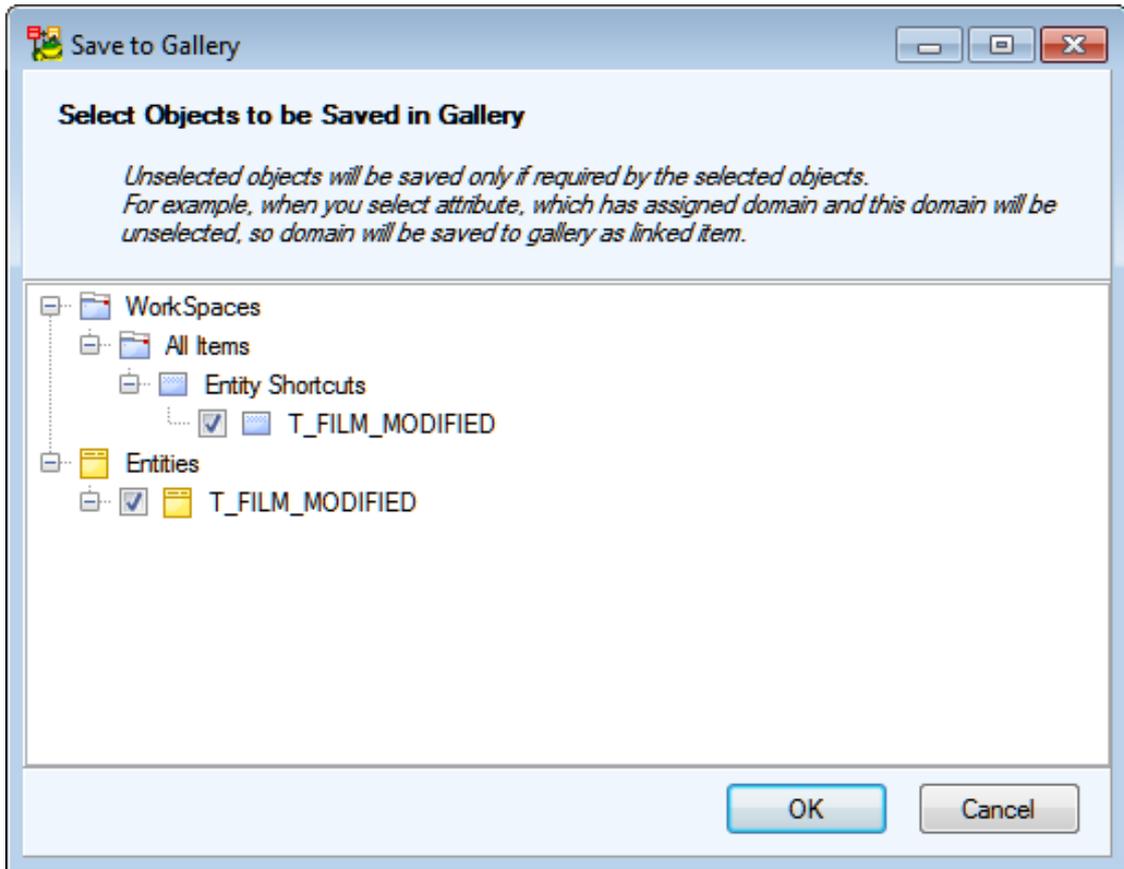
- **Multiple objects (shortcuts)** - the objects will be shown on Workspace, you can view and change their **Properties**.

**Example** - Two entities added to a Gallery at once by dragging them from **Model Explorer**.

3. One way or another, you should be able to modify the item(s) as you need. Once you're done, click the **Save to Gallery** button on **File Toolbar**.

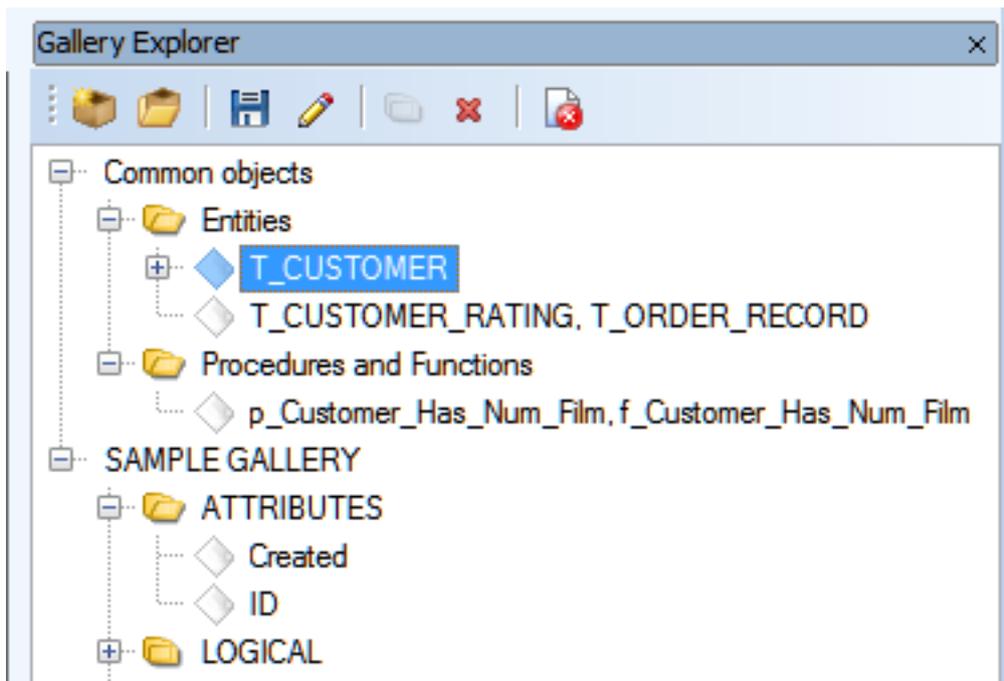


4. In the opened dialog, you can choose if you want to save the changes made to the modified item and its related items.



## Gallery Explorer

This windows is capable of displaying all of your Galleries and their objects and it also allows you to change their structure.



The Explorer toolbar contains several options:

Option	Description
New Gallery	Creates a new Gallery as described in <a href="#">Basic Actions</a> .
Open Gallery	Opens an existing Gallery file (.txgall file format)
Save Gallery	Saves recent changes made to Gallery.
Properties	Displays Gallery Properties - Name, Description, Path.
New Folder	Creates a new folder in the Gallery.
Delete Selected Item	Deleted the selected object.
Close Gallery	Closes (not deletes) the selected Gallery.

Items in the Explorer also have the following unique right-click options:

Option	Description
Save as (Gallery)	Allows you to save Gallery as a .txgall file to the selected location.
Item properties (Item)	Allows you to modify item properties (Name, Author, Description) and view its objects.
Modify (Item)	Allows you to modify the stored item in a separate model and synchronize the changes with the item in the Gallery. See <a href="#">Modifying Items</a> for more information.

# Model Verification

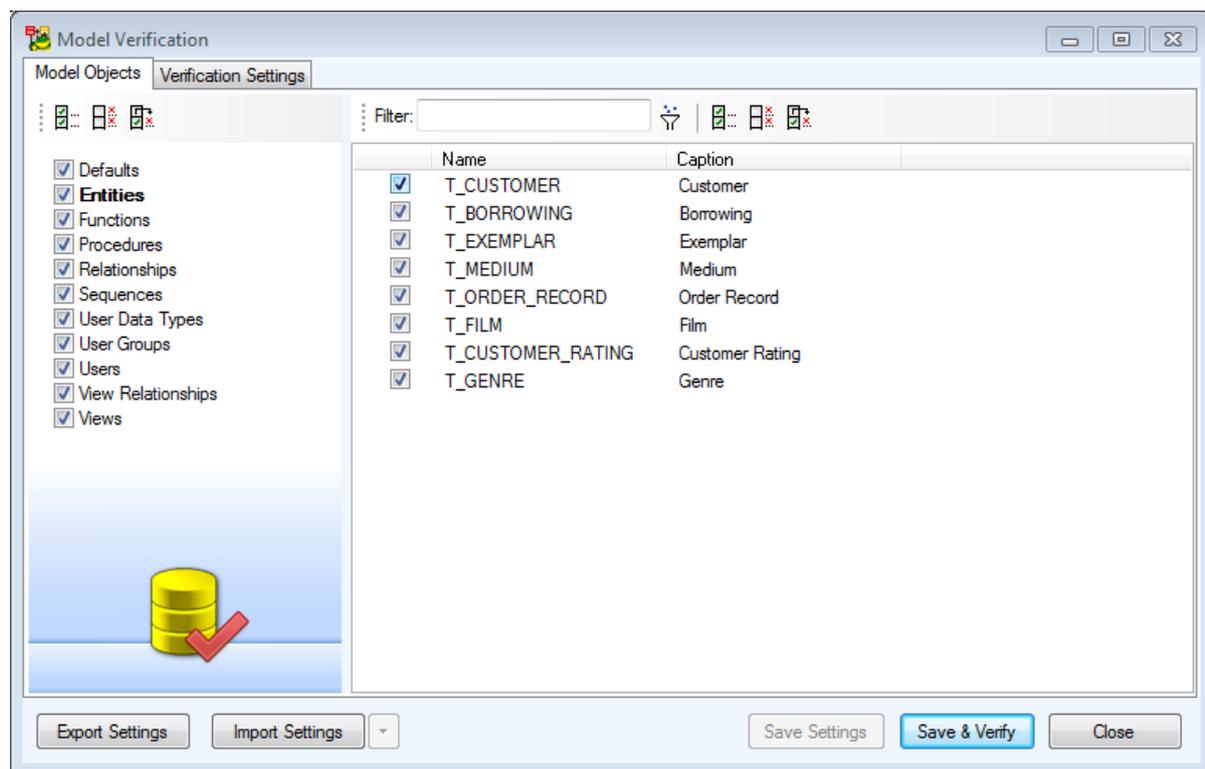
Toad Data Modeler allows you to verify your models. You can choose the items you want to verify, find out what errors and flaws your model contains and use quick fixes to resolve any issues.

## To verify your model

1. Select **Model Menu | Verify Model...** (also CTRL+F9).
2. On the **Model Objects** tab you can mark items for verification. Select either entire object group types or single objects.
3. On the **Verification Settings** tab select the rules that should be part of the verification. To save the settings, click **Save & Verify**.
4. The result of the Model Verification is shown in **Verification Log**, which is displayed automatically. To display the log manually, select **Windows Menu | Verification Log**. Objects that did not pass the verification are also marked by error/warning icons in **Model Explorer**.

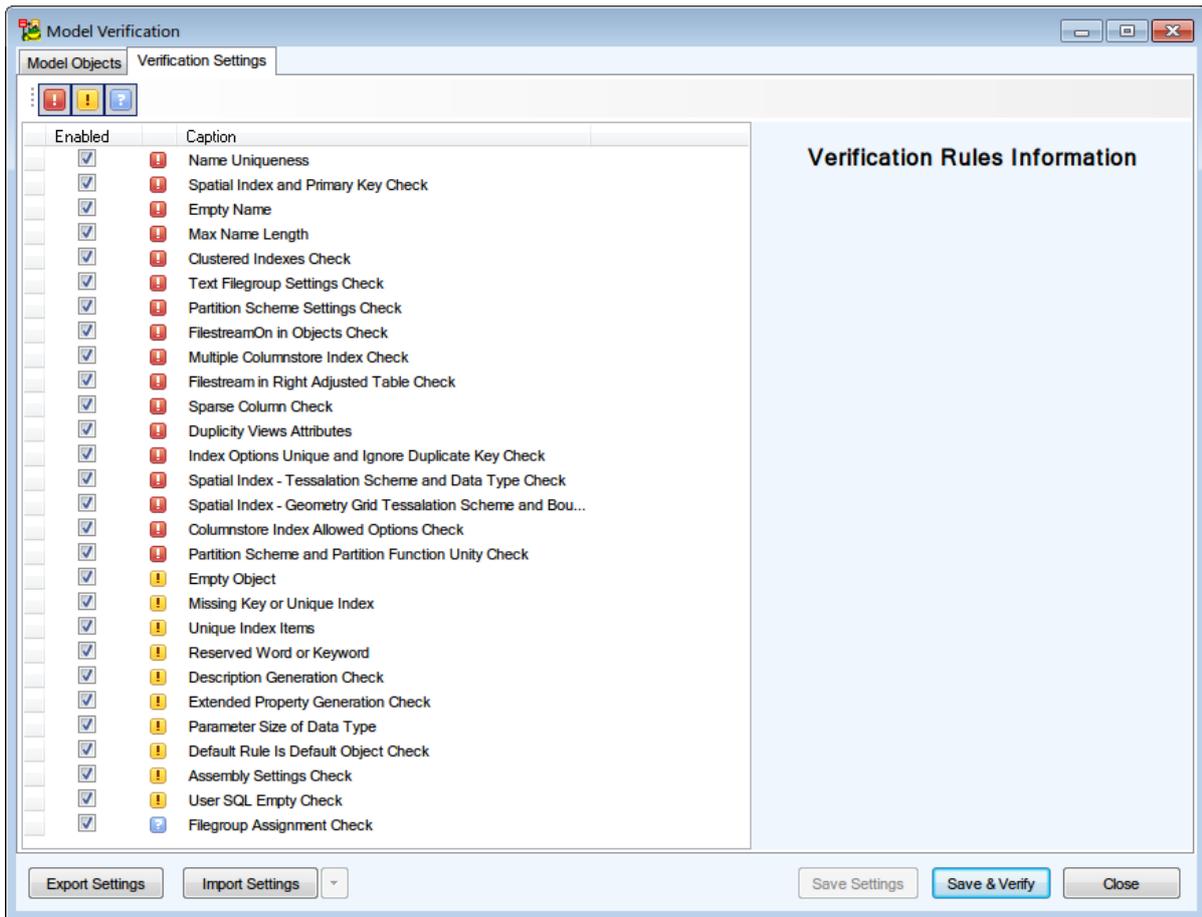
## Model Verification Form

**Model Objects tab** - lists all objects and object group types in your model. Checked object are those that will be verified.



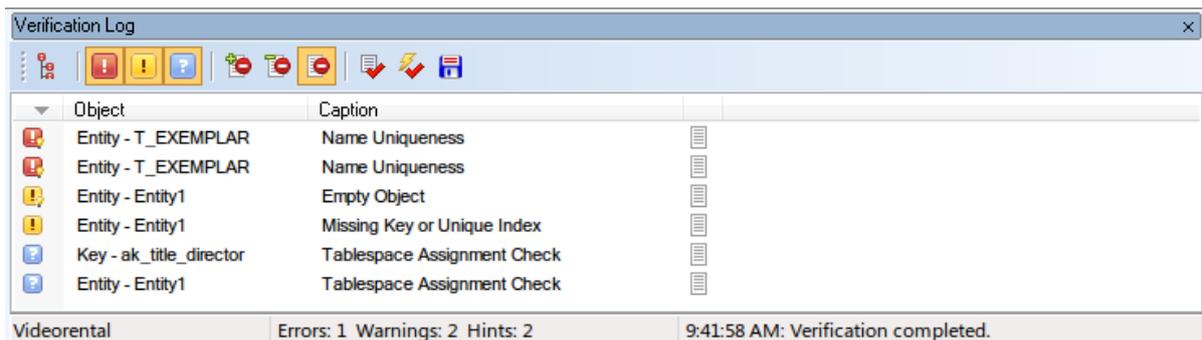
**Verification Settings tab** - Here you can enable/disable rules that should be considered during verification. Use the buttons on the top of the form to check/uncheck all errors/warnings/hints at once.

Rules are database dependent, the following screenshot shows rules for Microsoft SQL Server 2012.

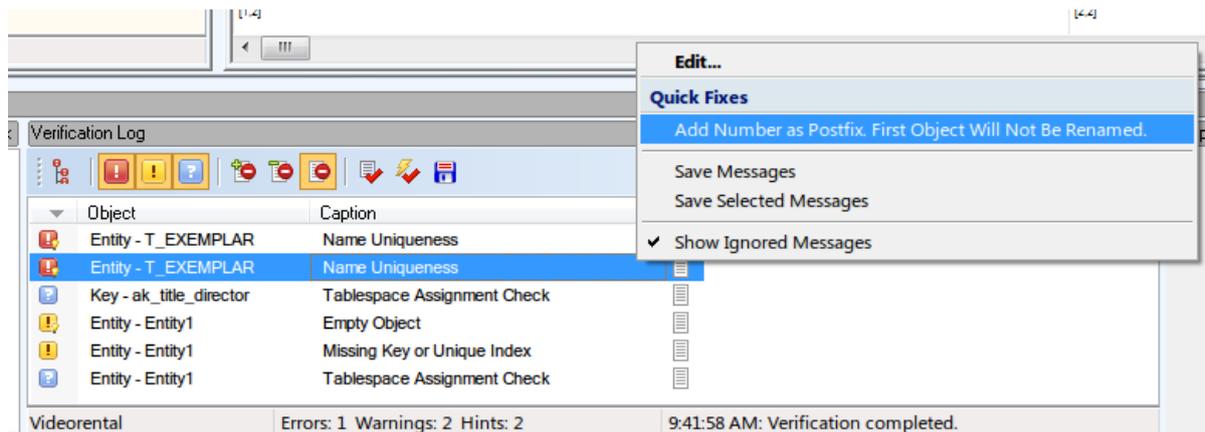


## Verification Log

Verify Model results are displayed in **Verification Log**.



Toad Data Modeler offers **Quick Fixes** - recommended solutions for problems and issues found by Model Verification. Quick Fixes are accessible for problematic objects both in **Verification Log** and **Physical Model Explorer**, in right-click menu.

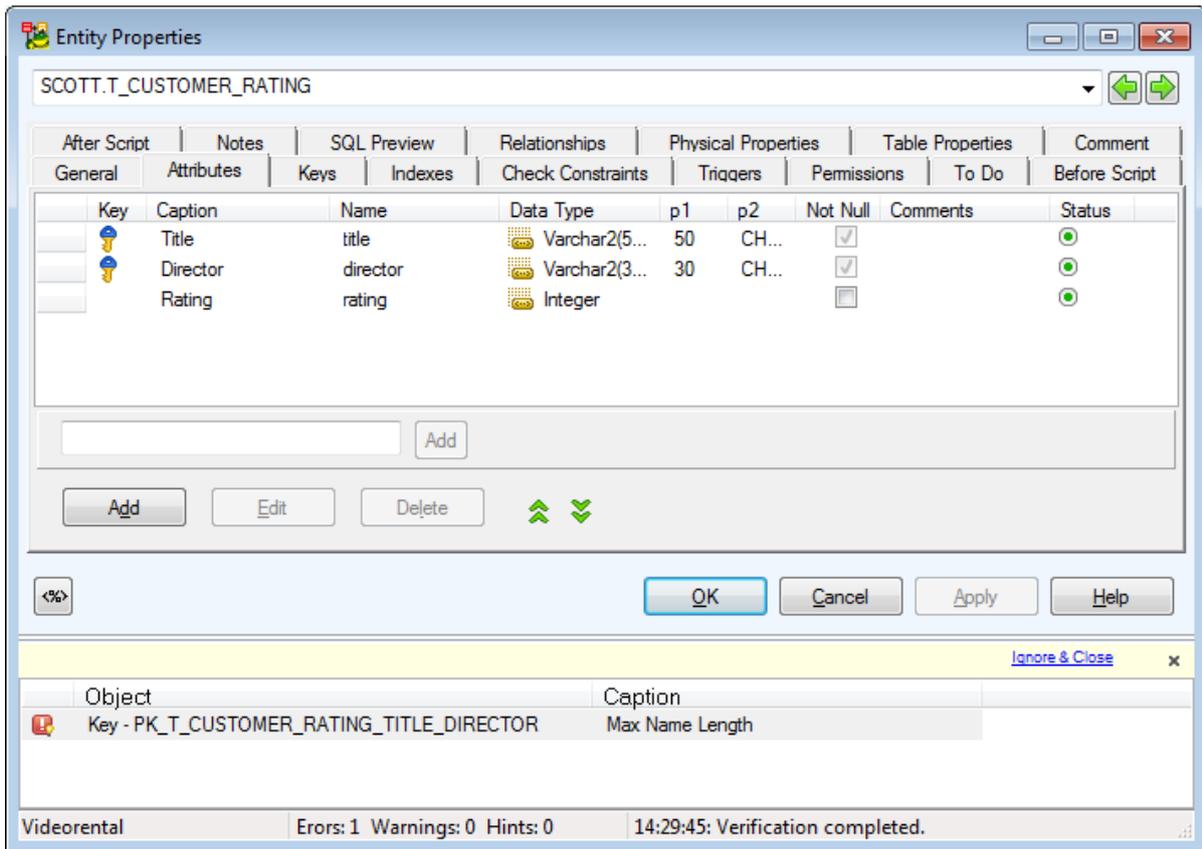


The Verification Log toolbar contains several buttons:

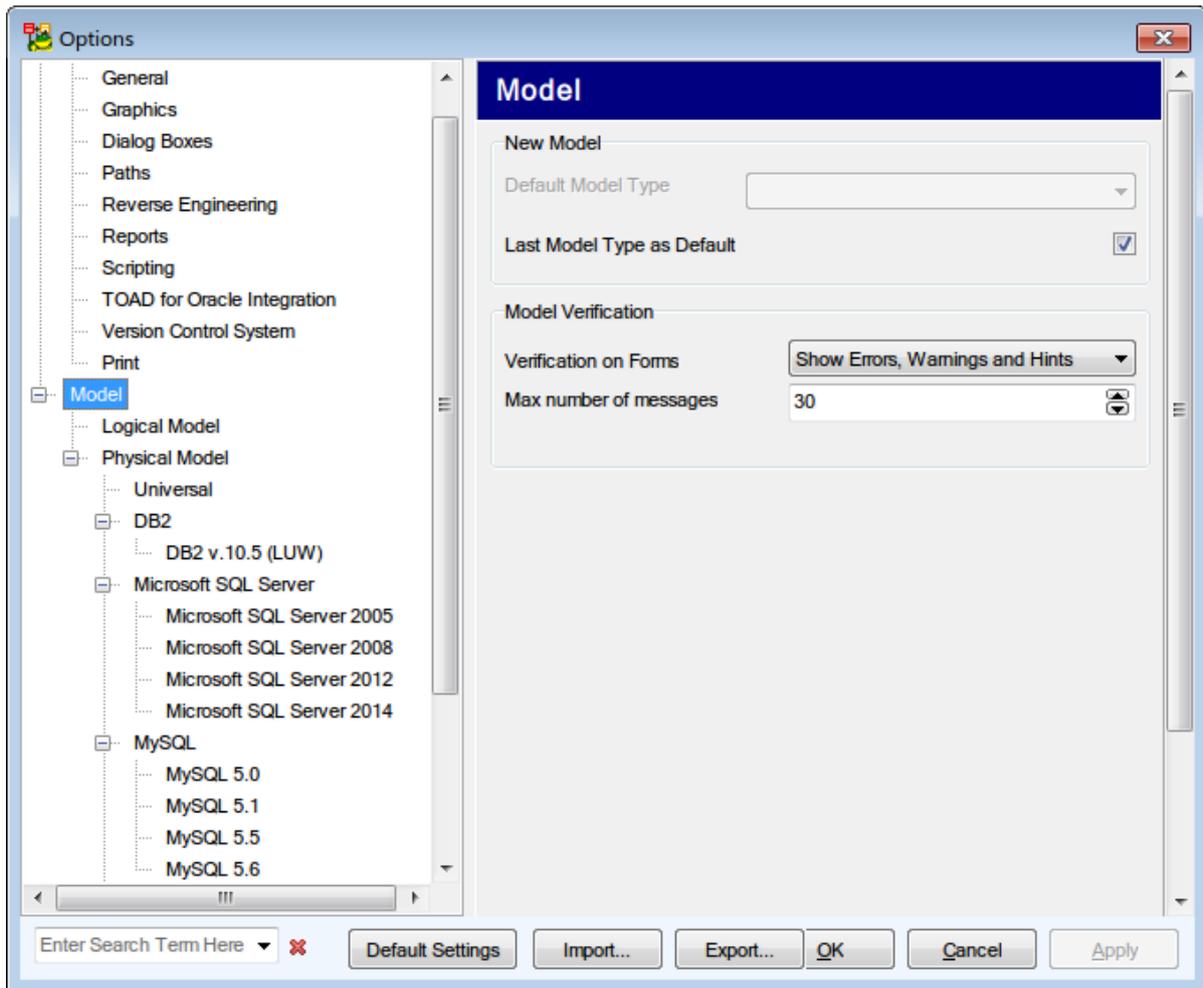
Button	Option	Description
	Tree View	Switches Verification Log to tree view where the problem description, affected objects and quick fix suggestion can be all seen. Note that you cannot ignore messages or remove them from ignored in this view.
	Show Errors/Warnings/Hints	When enabled, Errors/Warnings/Hints are shown in Verification Log.
	Ignore Message	Marks the selected message as ignored.
	Remove Message from Ignored	Removes the selected message from ignored list.
	Show Ignored Messages	Shows/hides ignored messages.
	Verify Form	Displays Model Verification dialog.
	Verify	Verifies model using the last saved settings.
	Save Verification Log	Saves all verification messages (including the ignored ones) to a CSV file.

## On Form Verification

Errors, warning and hints can be displayed directly in the Object Properties form, at the bottom. Every time you make changes to the object and confirm, the object is verified and all issues are displayed at the bottom, similar to the screenshot below. Quick Fixes are not available in this mode.



On Form Verification can be enabled/disabled in **Settings Menu | Options | Model** . You can also decide what kind of verification messages should be shown - errors/warning/hints.



## Syntax Validity

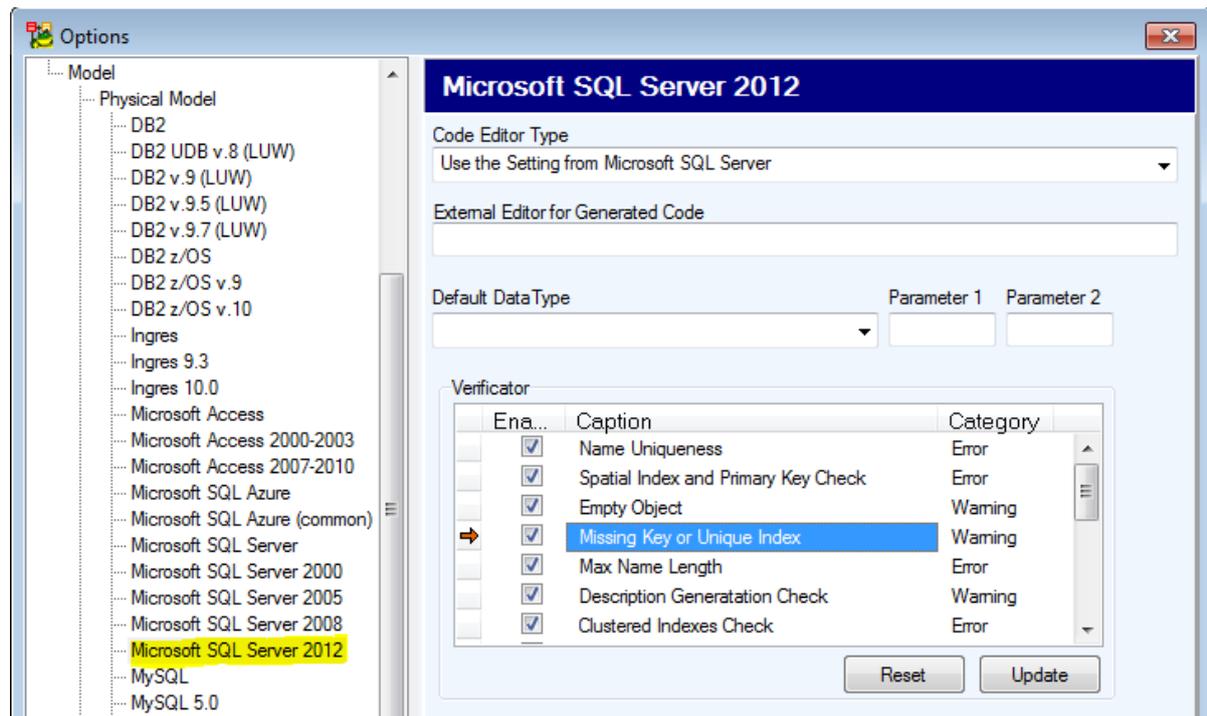
SQL Preview tab now highlights errors in SQL script syntax.

- Check Syntax Validity to perform syntax checks in SQL Preview for the following objects:
  - Procedure, Function, View, Materialized View
  - User Data Type, Package (Oracle only)
- Options are available in **Options | Model | Physical Model | Verification | SQL Syntax Options**
- Syntax Validity check is supported for the following databases:
  - IBM DB2 LUW, IBM DB2 z/OS, Microsoft Azure SQL Database, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, SAP Anywhere, SAP ASE, SAP IQ, Teradata

## Settings

Model Verification can be configured for each specific database. Go to **Settings Menu | Options | Model | Physical Model | \*Target Database\***.

In the Verificator frame you can enable/disable verification rules or change their category, e.g. Empty Object can be shown as an error instead of warning.



Syntax validity check may be disabled in **Settings | Options | Model | Physical Model | <database> | Syntax Validity** checkbox

## Data Generation for SQL Server

Use Toad Data Modeler random data generation to quickly and easily create data for your databases. Data generation is supported for SQL Server 2005 and onwards

### To generate data

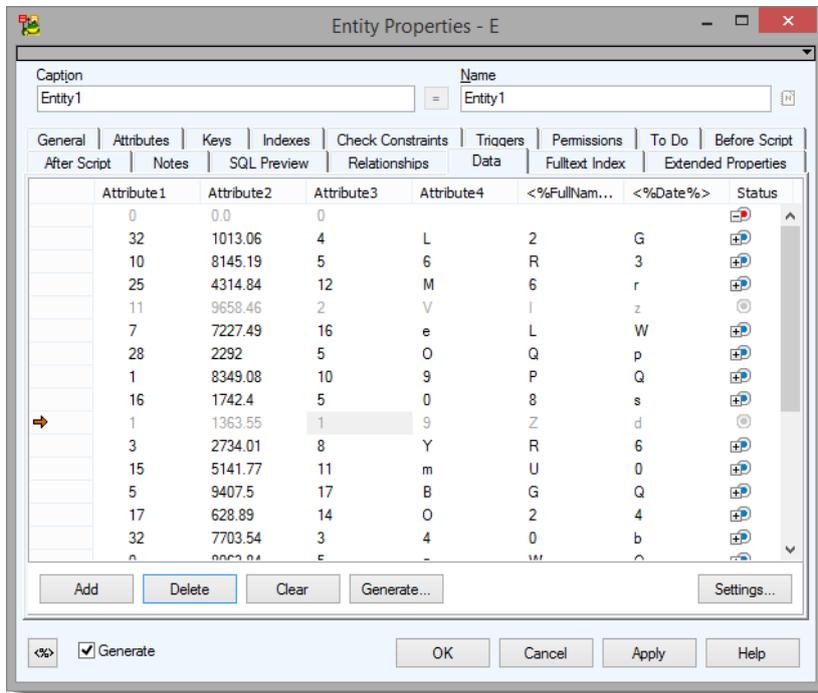
1. Double click any of your entities to open **Entity Properties**
2. Setup your **Attributes** and switch to **Data** tab to generate your data
3. Click **Add** to add one record, click **Generate** to add a specified number of new records

### To work with records

- Click **Delete** to remove the selected record and **Clear** to clear out all your generated data. Cleared data appears grayed out
- Click **Settings** to disable/enable attributes and adjust minimum and maximum values for generation of data for each column (attribute) for the current model

- Click **Settings | Options | Model | Physical Model | Microsoft SQL Server | Generating Data** to adjust default data generation settings for all MS SQL Server models. When you change any of these default values it will be carried on to all your SQL Server models

**i** **TIP:** Click `<%>` to add application variables in **Attributes** in order to use them in your generated data



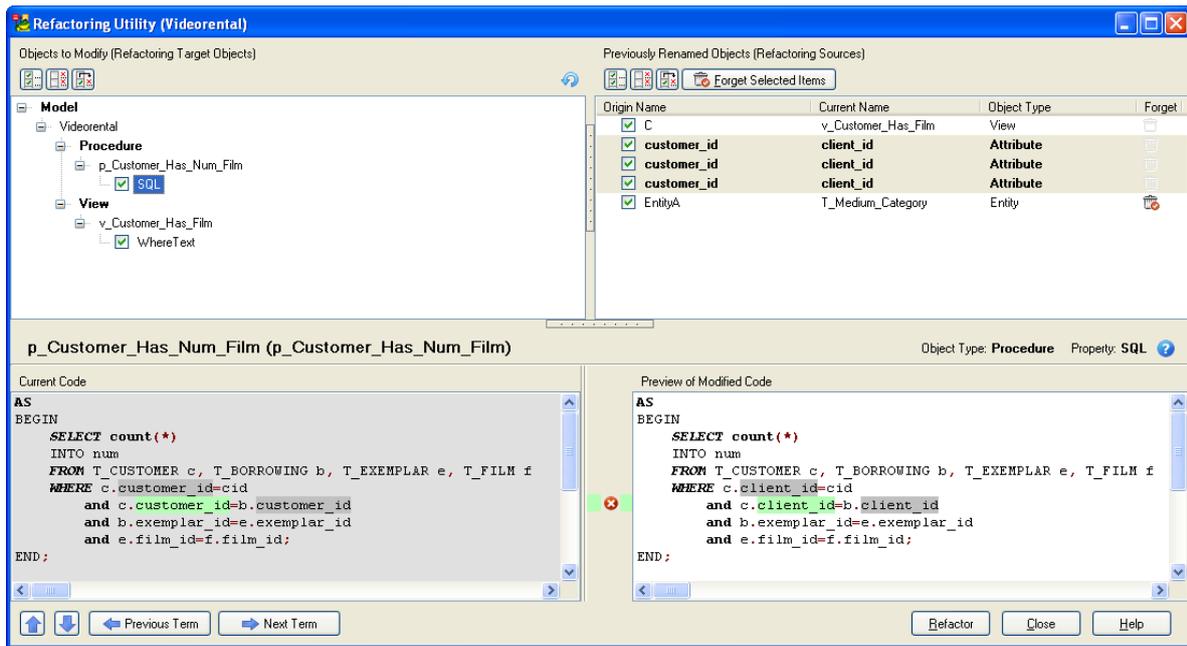
## Refactoring Utility

Toad Data Modeler provides you with a tool that is able to refactor all references to an object when you change its name. For example, let's say you've changed name of an entity. Usually, you would have to go through the rest of the model and change the name in all the places where it is referenced. Refactoring Utility is able to do this automatically, saving your time.

**i** **Note:** The utility refactors text properties only (e.g. AfterScript, BeforeScript, SQL tab, text Views...)

### To open Refactoring Utility

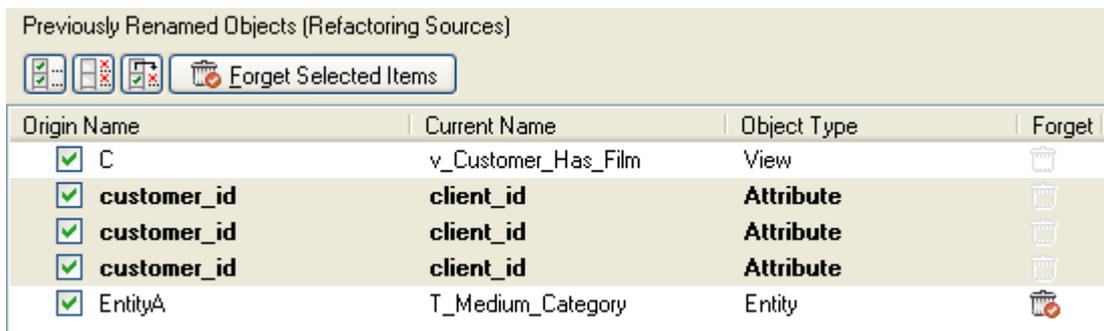
- Click **Tools | Refactoring Utility**



**Fast Parser** checkbox - When checked, the Refactoring Utility uses Fast Parser to find broken references. In some cases it might be able to find more inconsistencies than the default Advanced Parser.

## Renamed Objects

In section **Previously Renamed Objects** (top-right), you can find objects that were renamed.

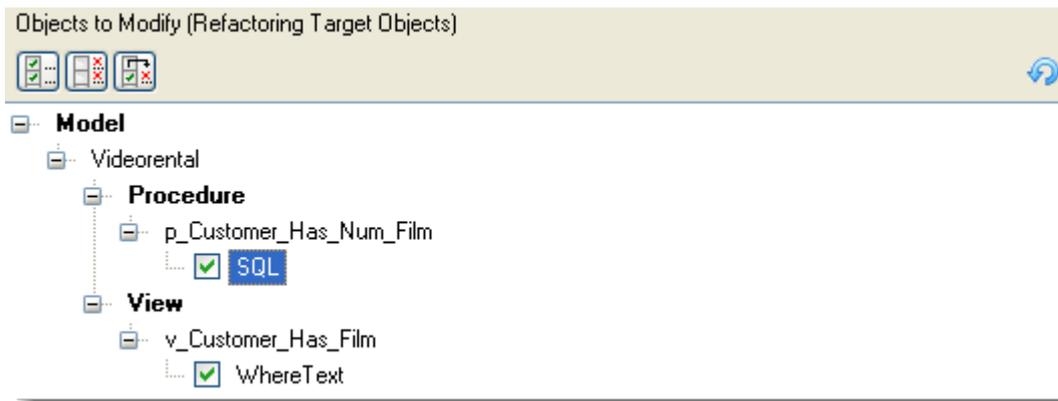


Use checkboxes to select what items you wish to use for current refactoring action.

Renamed objects will stay in section **Previously Renamed Objects** until you change names in other objects or until you enable icon in column **Forget** and click the **Forget Selected Items** button. (In this sample it should be used for EntityA).

## Objects to Modify

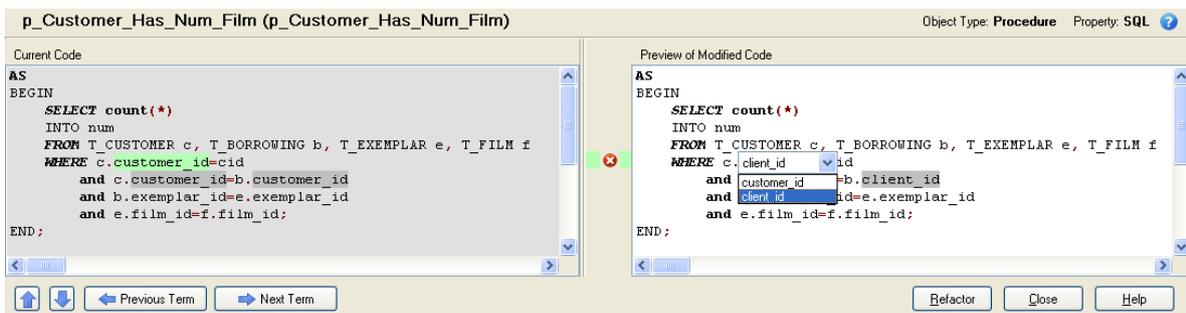
In section **Objects to Modify**, you can find objects and properties that contain old names.



## Current and Modified Code Previews

In section **Current Code**, the body of SQL code is shown. Use buttons **Previous Term** and **Next Term** to navigate among names in the same code.

Section **Preview of Modified Code** displays a preview of refactored code. Click any of the highlighted names and select old or new name or use the icons in middle column to reject suggested changes.



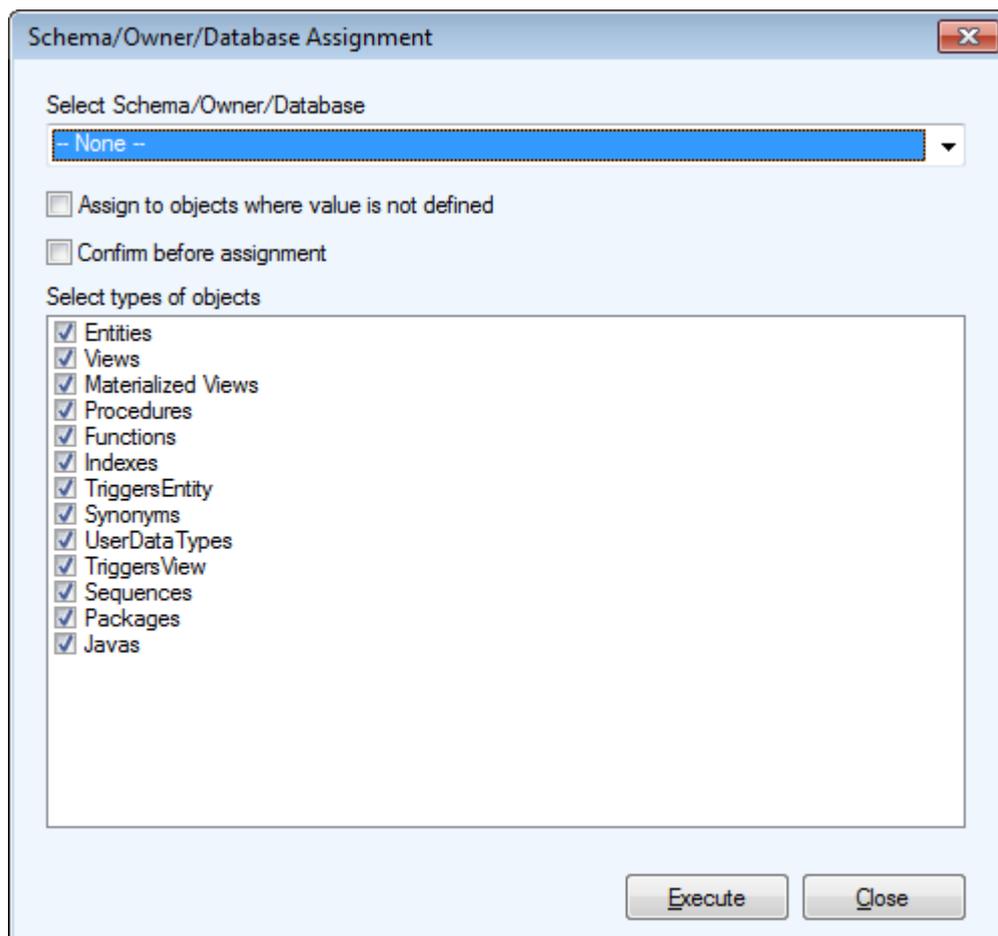
**Note:** Renaming from the default name (Entity1, Entity2...) is ignored deliberately.

## Schema/Owner Assignment

This tool allows you to assign or remove schema/owner/database to/from multiple object groups in your model at once.

### **To assign a schema/owner to your model**

Select **Tools | Schema/Owner Assignment**.



Option	Description
Select Schema/Owner/Database	Contains existing Schemas/Owners/Users in your model. Select <b>--None--</b> from the list to remove the existing schema from objects you mark in the <b>Select types of objects</b> section.
Assign to objects where value is not defined	Assigns schema/owner to objects, which have none assigned yet.
Confirm before assignment checkbox	Confirmation dialog for each object where schema is being assigned/removed will be displayed.
Select types of objects area	Mark objects which should be assigned a Schema/Owner.

## Infer Relationships

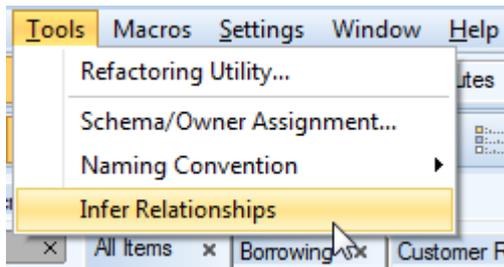
This feature tries to map **Primary Keys** or **Alternate Keys** to identically named attributes in other entities.

Conditions:

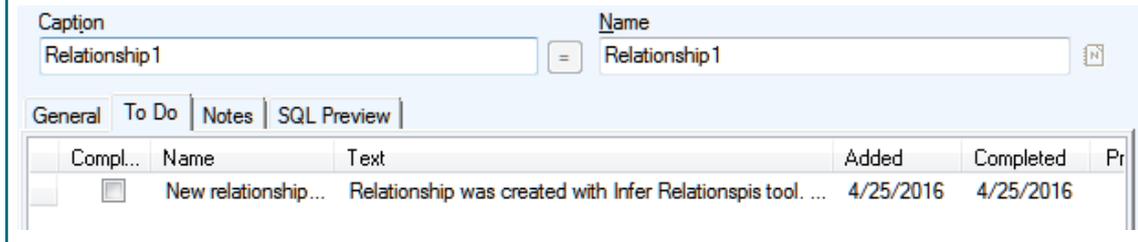
- Mapping is performed between key and non-key attributes. In other words, if identically named attributes are part of primary key in various tables, e.g. ID column in table Customer and ID column in table Order, the two ID columns will not be mapped and no relationship will be created.
- Names must be identical and data type must be the same, including parameters.
- Non-identifying relationships are created by default. In case you need an identifying relationship, edit the automatically added relationship and change its type to identifying manually.

### To run the Infer Relationship function

Select **Tools | Infer Relationship**.



**i** Note: Relationships created this way will be marked with a **To-Do task** prompting you to check whether the relationship has been created correctly.



### Example:

Model with no relationships. See the **Customer ID** column in tables **Customer** and **Order Record**.

In table **Film** there is an alternate key with two columns **Title** and **Director**. Identically named columns are in table **Customer Rating**.

Customer
<b>Customer ID NN (PK)</b> Name (IX1) Address <hr/> i_name (IX1)

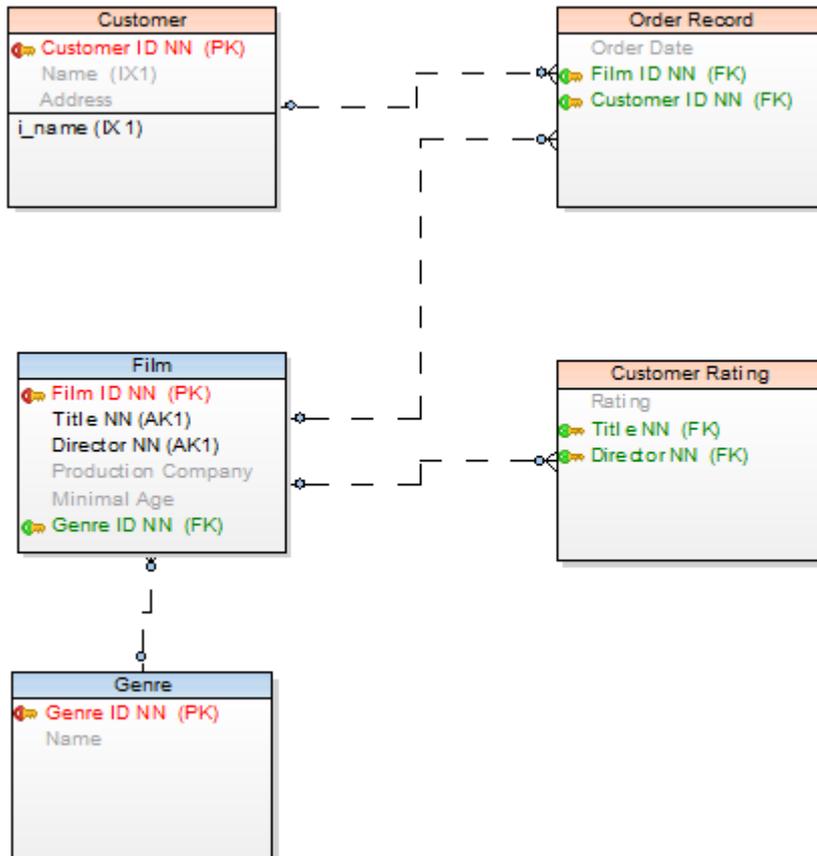
Order Record
Order Date Film ID NN Customer ID NN

Film
<b>Film ID NN (PK)</b> Title NN (AK1) Director NN (AK1) Production Company Minimal Age Genre ID NN

Customer Rating
Rating Title NN Director NN

Genre
<b>Genre ID NN (PK)</b> Name

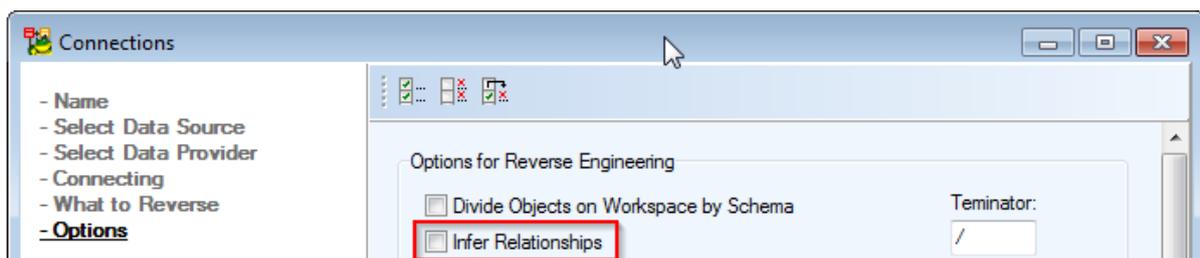
Run the Infer Relationship function and see the result:



## Infer Relationships and Reverse Engineering

The same tool can be used automatically during reverse engineering.

Create a new connection or edit an existing stored connection and check the **Infer Relationships** checkbox to activate this feature.



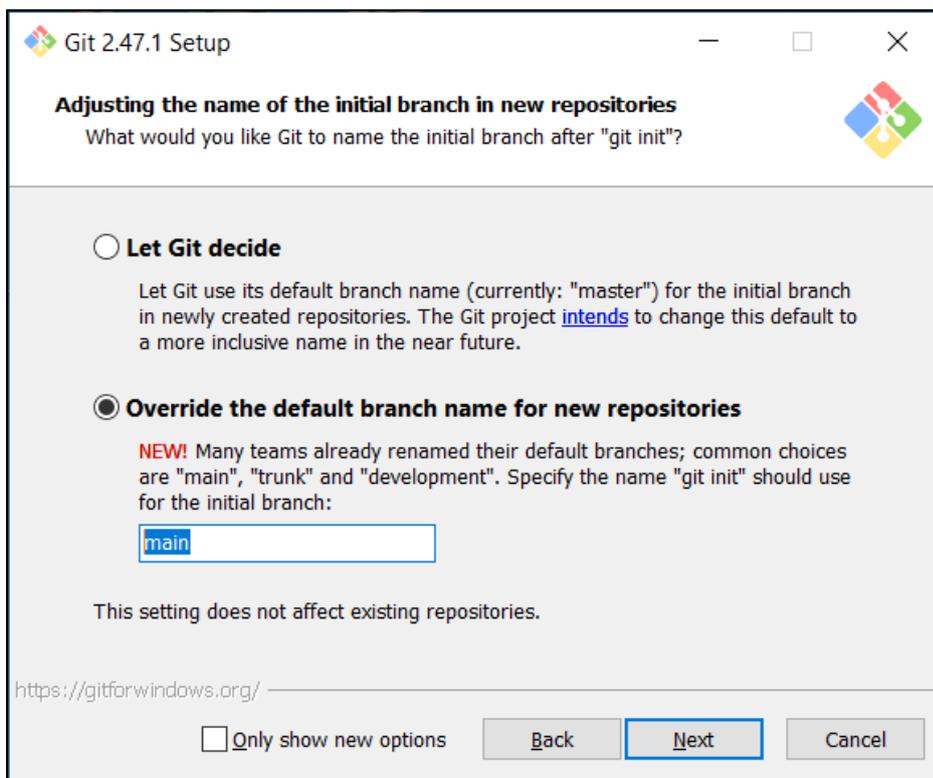
# Git Version Control

## To configure git

Configure Toad Data Modeler to work with Git in **Options | Application | Version Control System**. The values set here will be your default values for setting up new repositories in Toad Data Modeler.

1. Select **Git** in Type
2. Set paths to your Git client and working directory for your repository
3. Click **Apply** and **Initialize Local Repository**. The folder will be created (if it does not exist) and initialized
4. Enter a path to the remote repository, user name, and password
5. Click **Apply** and **Clone Repository**

**NOTE:** If users select the **Override the default branch name for new repositories** option during Git installation, the branch name (Master/Non-Master) specified for the Toad Data Modeler product must match the branch name in the GitHub or Bitbucket repository.



## Git application view

- The following Git commands are displayed in **Application View** toolbar when you work with a model that is tracked:



- [Add Existing Models to Project](#) to display the Git commands in **Application View**

Application View Button	Description
<b>Version Control System Settings</b>	Opens <b>Options</b> page where you can set the <b>Version Control System</b> preferences
<b>Refresh</b>	Refreshes file status in Toad Data Modeler
<b>Add to Index</b>	Adds the file to the list of tracked files
<b>Remove from Index</b>	Removes the file from the list of tracked files
<b>Commit</b>	Commits changes
<b>Push</b>	Pushes the committed changes into the remote repository
<b>Pull</b>	Pulls the current state from the remote repository
<b>Fetch</b>	Fetches information about the current state of the remote repository
<b>Information</b>	Displays information about the commit and its author
<b>History Browser</b>	Allows you to browse different versions of the file, merge and compare them and create change script

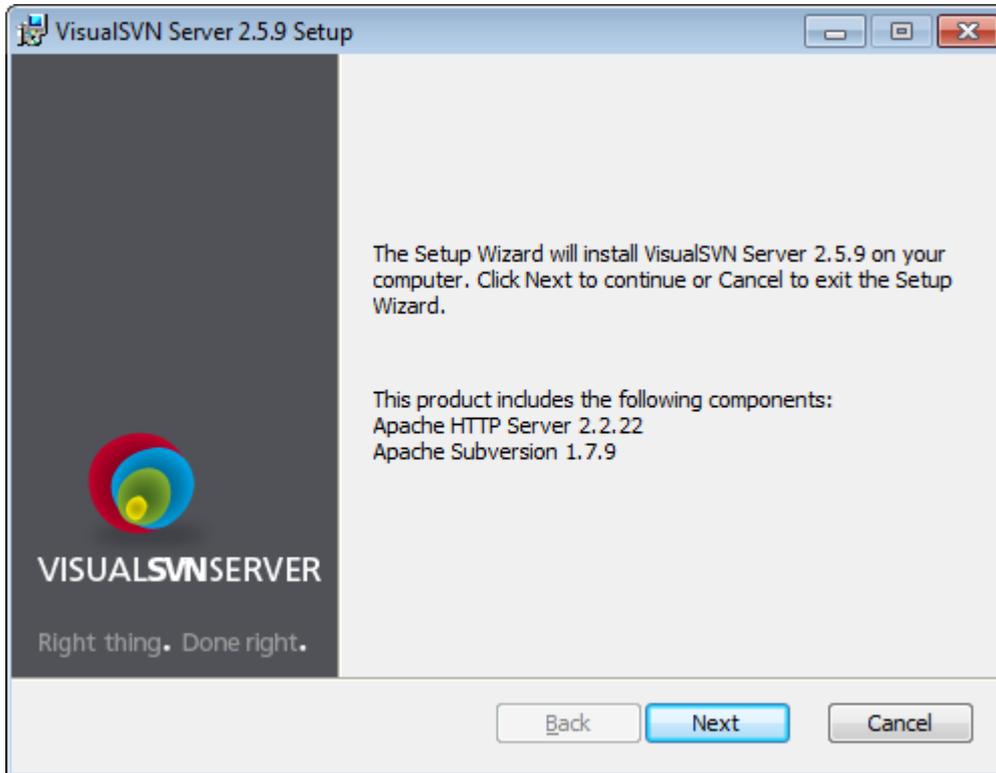
## Getting Started - Subversion

Toad Data Modeler offers you integration with a third party version control system - Apache™ Subversion®.

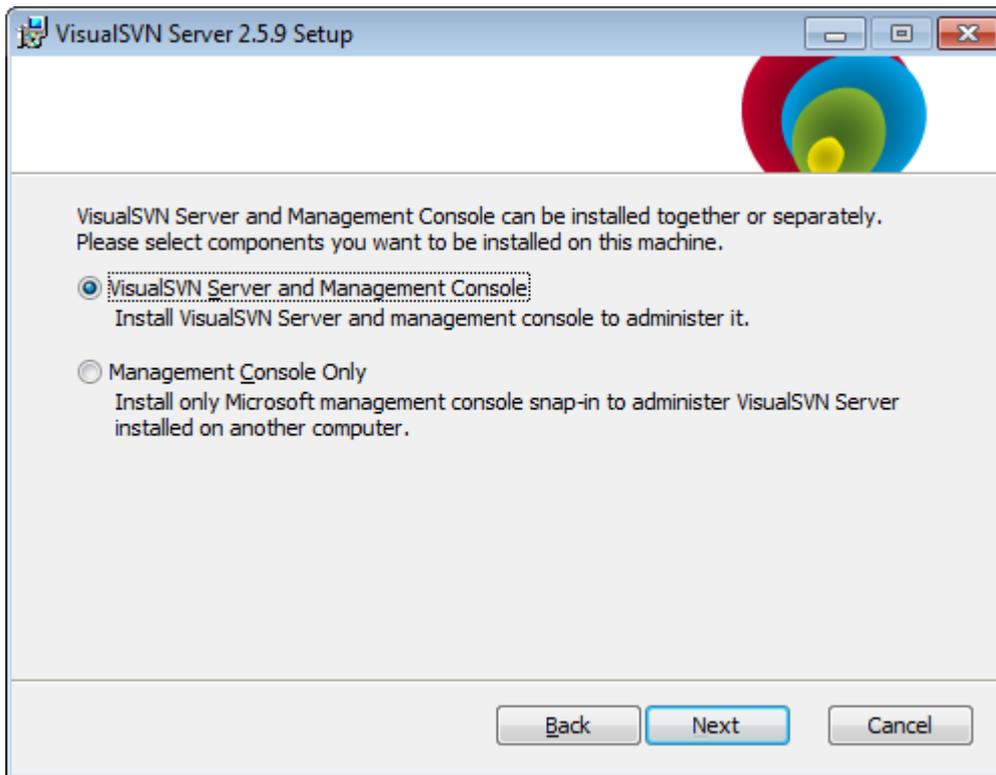
- i** Note: If you are familiar with Apache Subversion, you can skip this topic and navigate to the [Application Settings](#) topic.

### To configure Apache Subversion on your machine

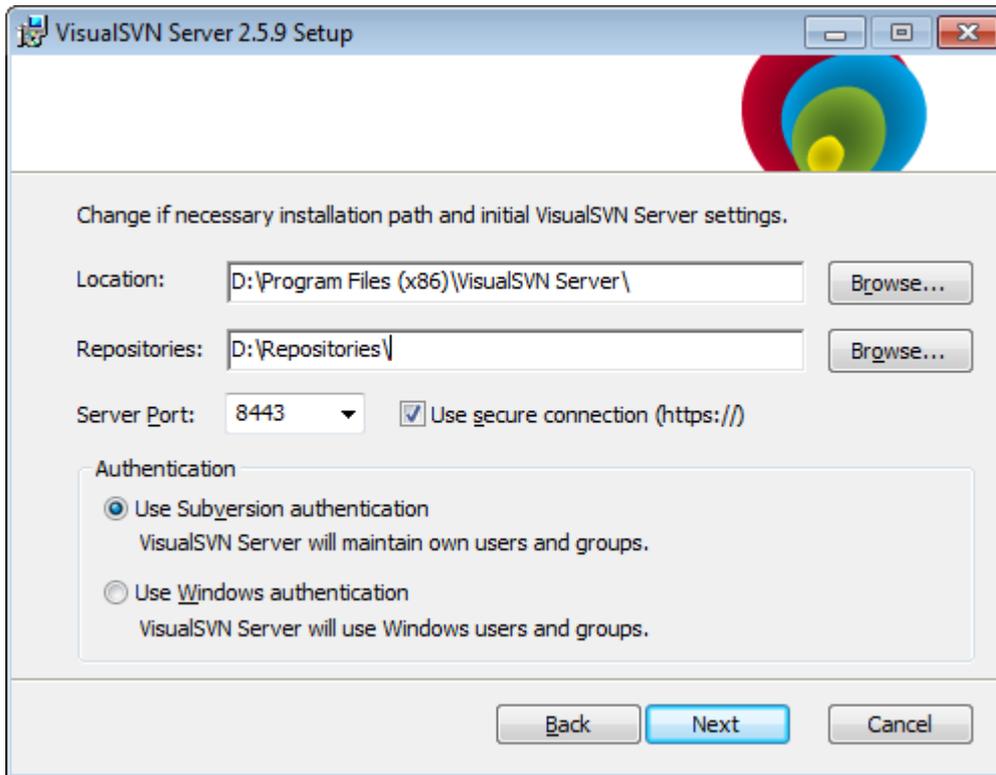
1. Download your preferred installation package. Here, the VisualSVN is the preferred package because it contains Server, Management Console and svn.exe.



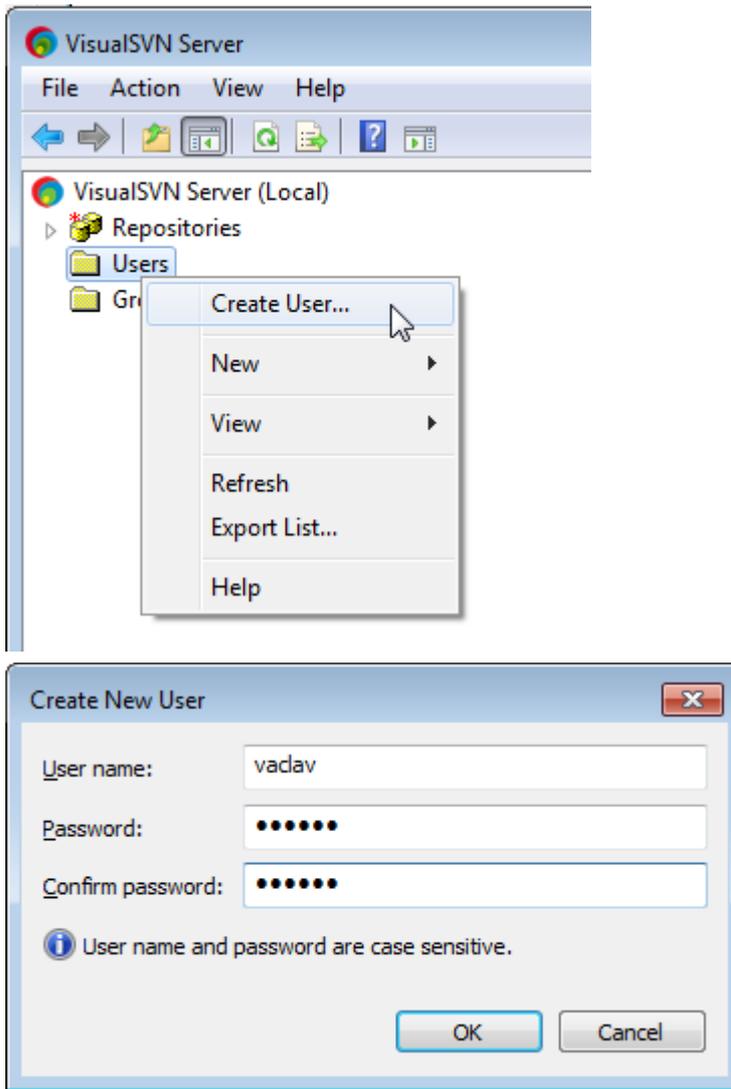
2. Install both VisualSVN Server and Management Console.



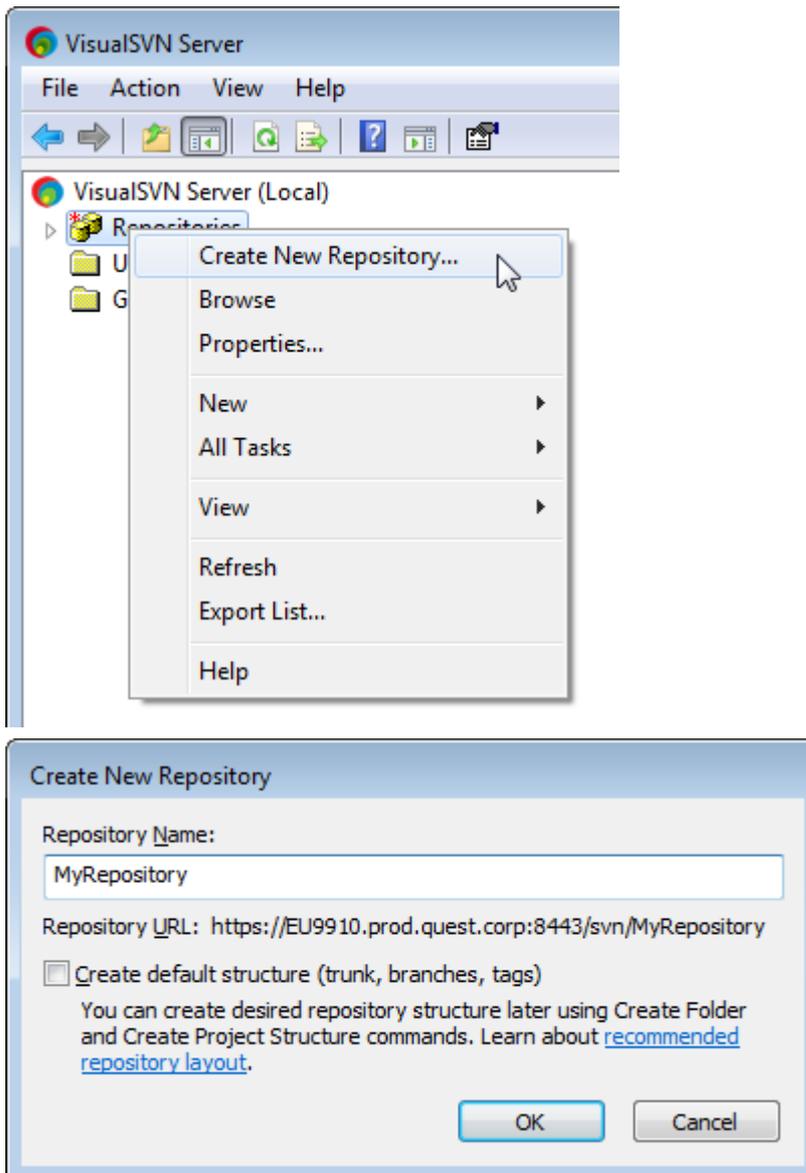
3. Define a path to your SVN Server location, specify folder for your repository and select an authentication method.



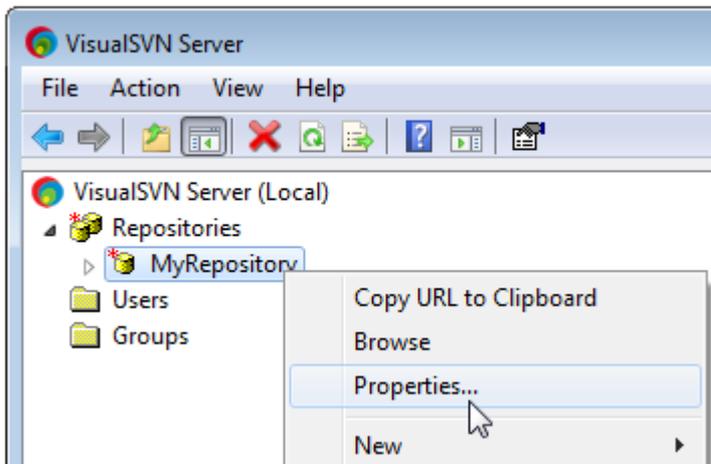
4. After the installation, run the Visual Server SVN Manager and create a new user.



5. Create a new repository.



6. Edit properties of your repositories and add the setup permissions for your newly created user.

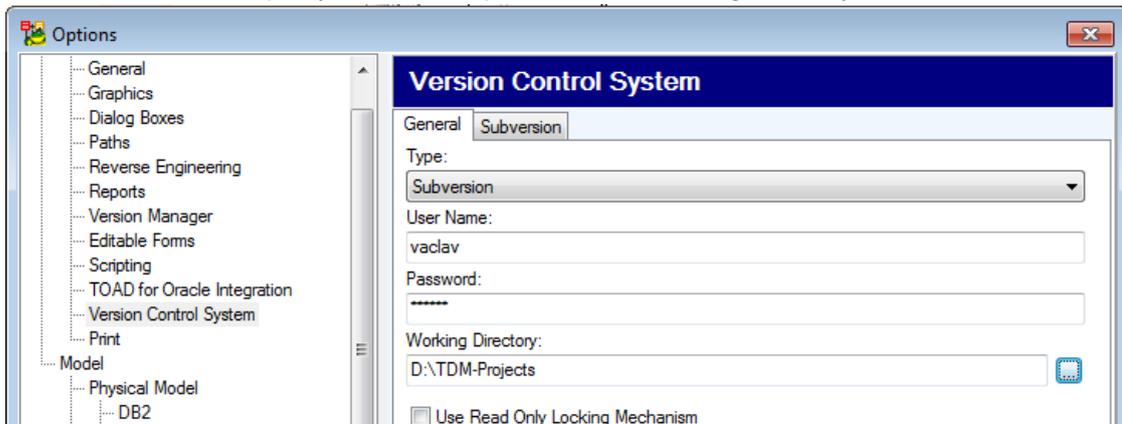


7. Next step: configure the Toad Data Modeler [Application Settings](#).

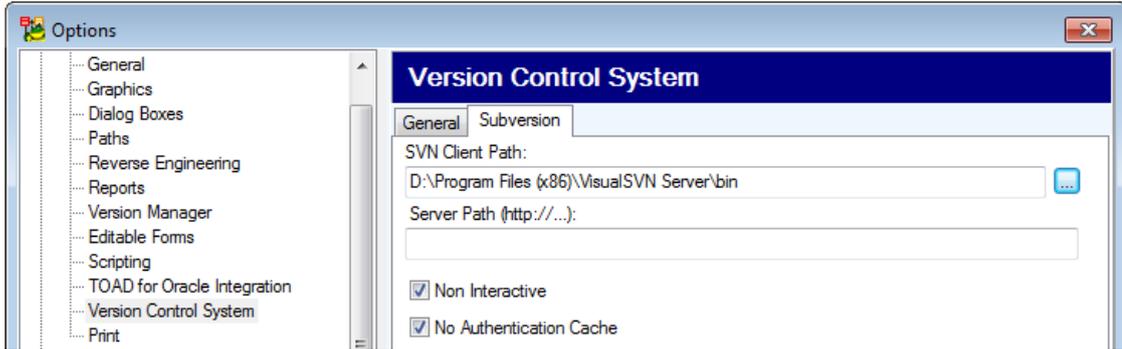
## Application Settings - Version Control System

### *To configure settings for Version Control System in Toad Data Modeler*

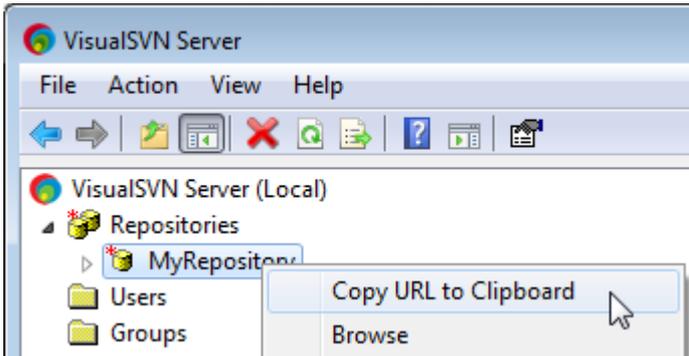
1. Select **Settings | Options | Version Control System**.
2. On the **General** tab, specify user name, password and Working Directory.



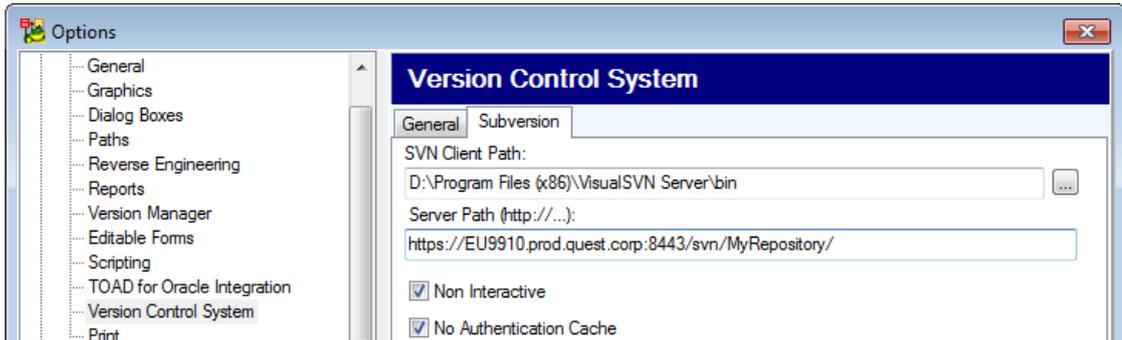
3. On tab **Subversion**, define a path to folder with **svn.exe** file.



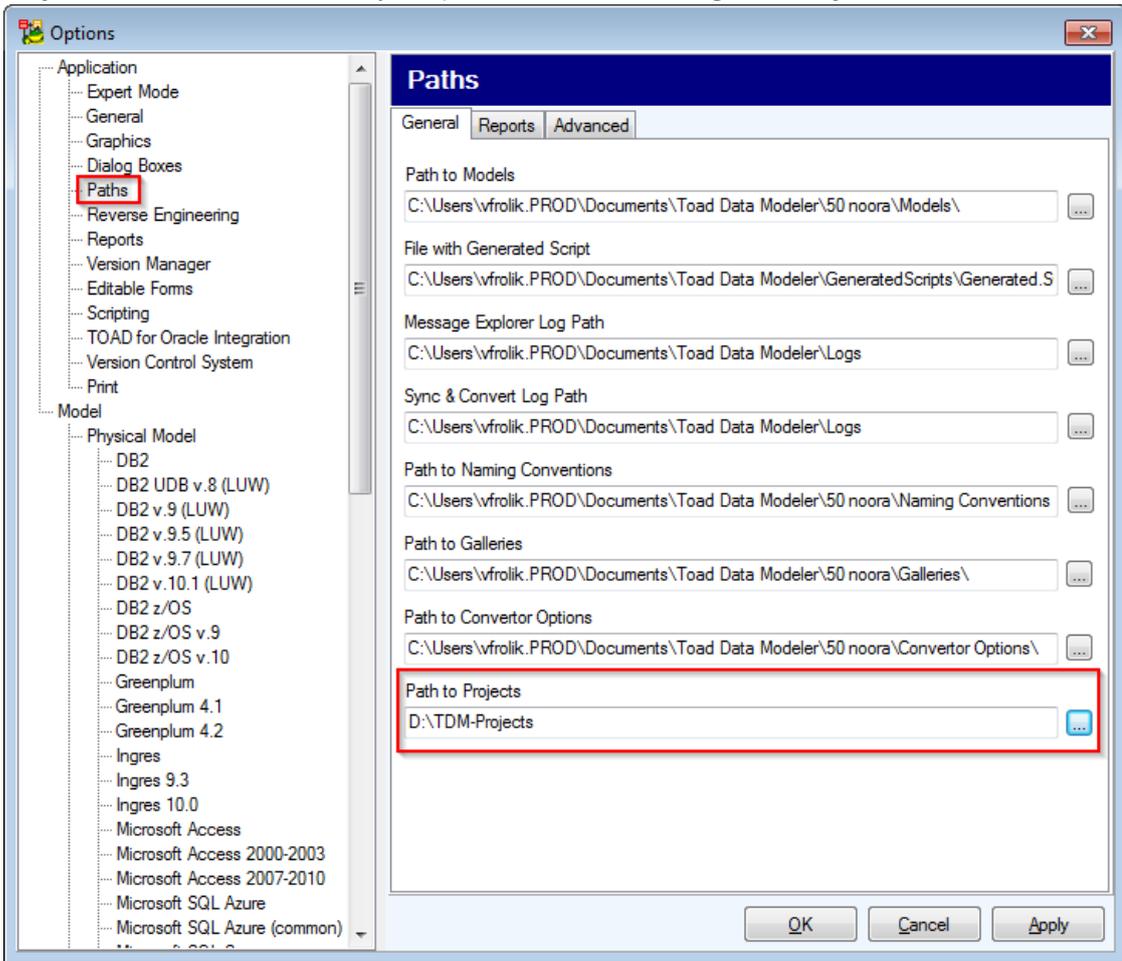
To save some time, make your **Visual Server SVN Manager** active, select your repository, right-click it and select **Copy URL to Clipboard**.



Then paste the content of your clipboard to the **Server Path** field.

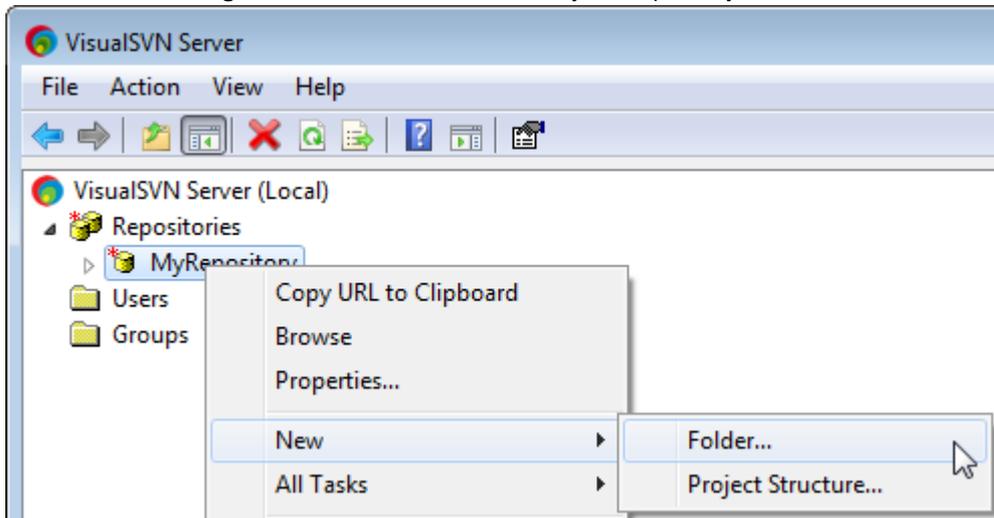


4. If you want to use Working folder for all your projects, select section Paths and define **Path to Projects** to the same folder as you specified in field **Working Directory**.

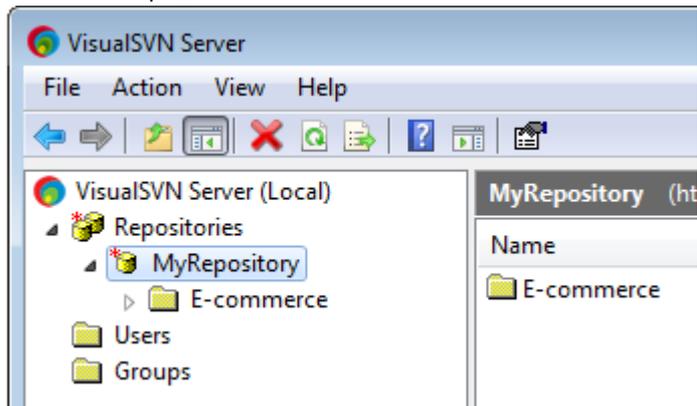


## Project Settings

1. Before you create a new project (with Version Control System) in Toad Data Modeler, run **Visual Server SVN Manager** and add a new folder to your repository.

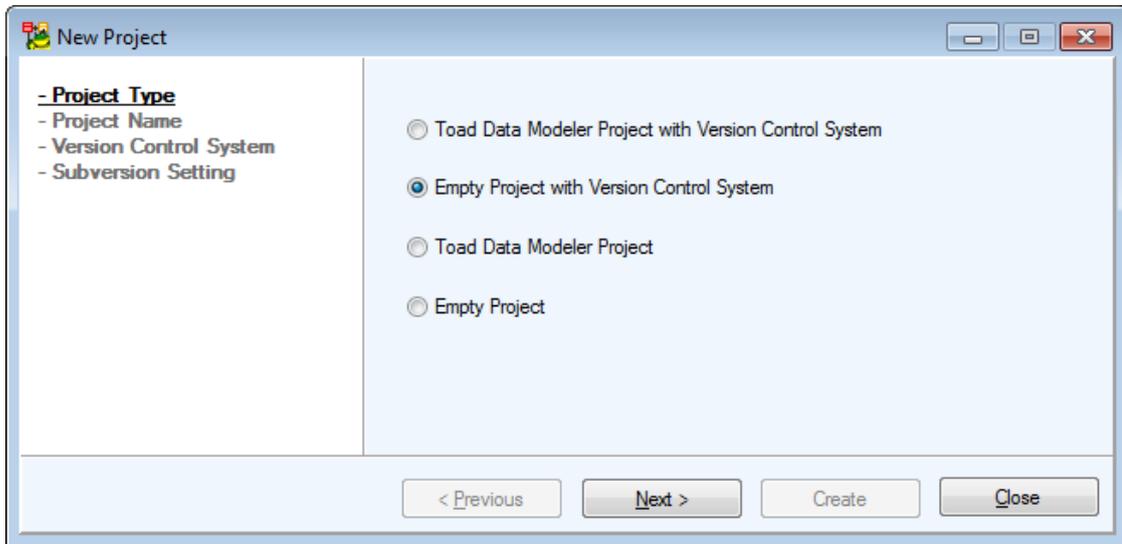


In this example, the new folder **E-commerce** will be created.



2. In Toad Data Modeler select **File | New | Project**.

3. If you want to define folders manually, select **Empty Project with Version Control System**. (Toad Data Modeler project has a predefined structure. See [Create new project](#) for more information.)

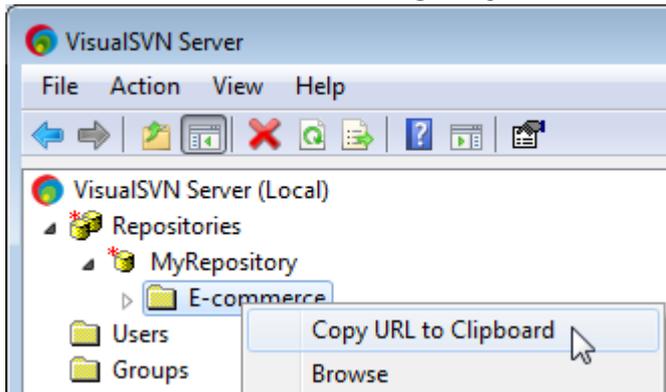


4. Define the project settings and to keep the configuration simple, use the recommended option - **Project and Repository Structures are Identical**.

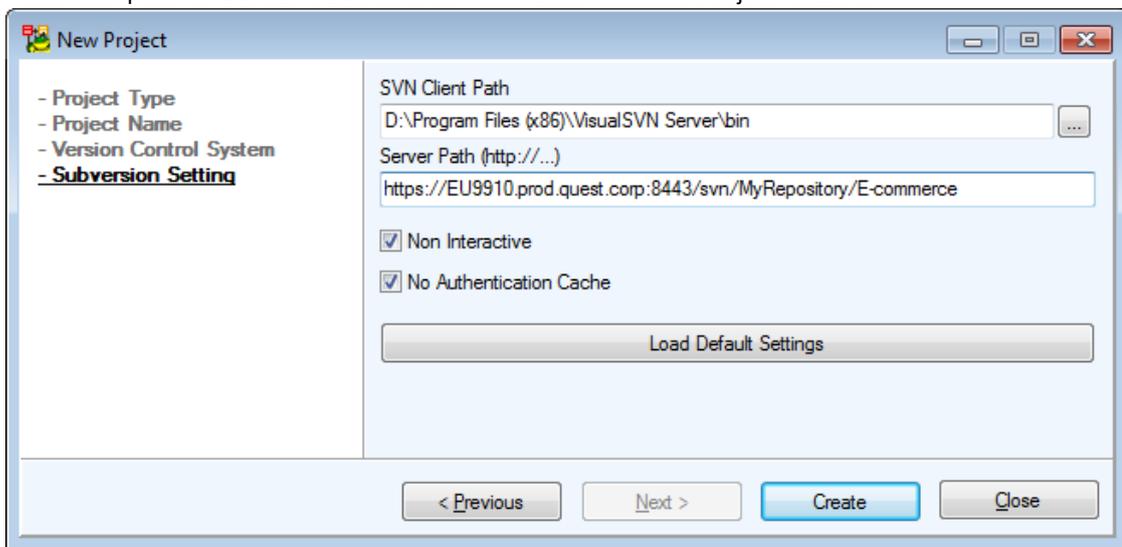
The screenshot shows the 'New Project' dialog box in the first step. The left sidebar contains a tree view with the following items: '- Project Type', '- **Project Name**', '- Version Control System', and '- Subversion Setting'. The main area is divided into two sections: 'Project Name:' with a text box containing 'New Project', and 'Path to Project:' with a text box containing 'C:\TDM-Projects\E-commerce' and a browse button (...). At the bottom, there are four buttons: '< Previous', 'Next >', 'Create', and 'Close'.

The screenshot shows the 'New Project' dialog box in the second step. The left sidebar contains a tree view with the following items: '- Project Type', '- Project Name', '- **Version Control System**', and '- Subversion Setting'. The main area is divided into two sections: 'Type' with a dropdown menu set to 'Subversion', and 'User Name' with a text box containing 'vaclav'. Below that is a 'Password' field with seven dots. The 'Working Directory and Project Structure' section contains three radio buttons: 'Project and Repository Structures are Identical (recommended)', 'Project is Part of Repository Structure' (which is selected), and 'Repository is Part of Project Structure'. Below the radio buttons is a text box containing 'C:\TDM-Projects\E-commerce' and a browse button (...). At the bottom, there are four buttons: '< Previous', 'Next >', 'Create', and 'Close'.

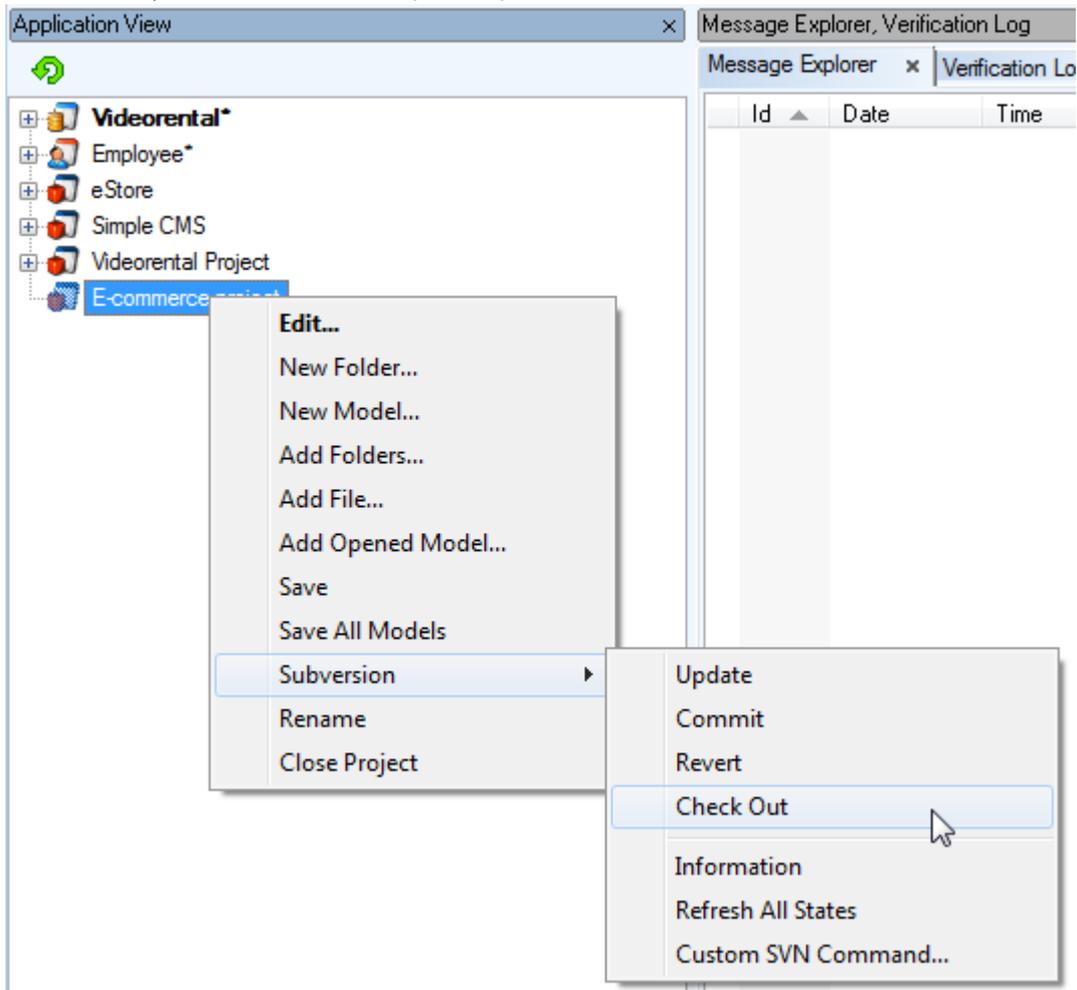
5. Finally, specify a path to the folder with `svn.exe` file and **Server Path**. To save time, select your folder in **Visual SVN Server Manager**, right-click it and select **Copy URL to Clipboard**.



6. Paste the path to **Server Path** field in Toad Data Modeler Project Wizard.



7. Important: Now you have to **right-click** your created project in the **Application View** and select **Subversion | Check Out**. This step is required!



## Subversion Actions

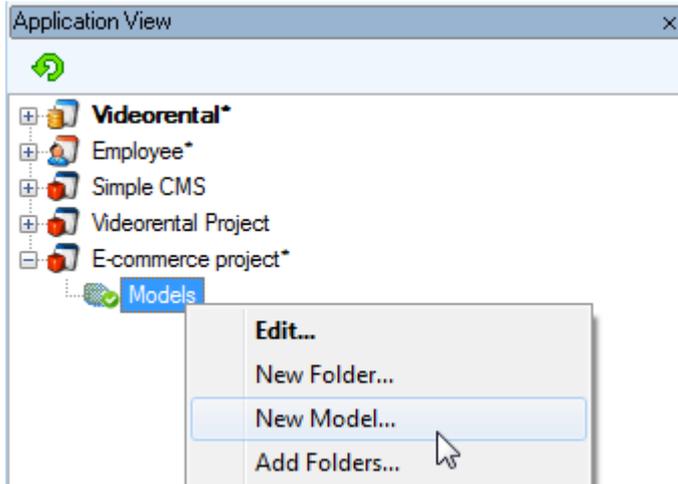
You can run Subversion actions from the **Application View**.

### **To add a new folder**

1. Right-click your project name and select **New Folder**. Create a new folder.
2. Right-click the folder and select **Subversion | Add**.
3. To commit changes right-click the folder again and select **Subversion | Commit**.

### To add a new model to project

1. Right-click the project name or folder name and select **New Model**.



2. Right-click the model name and select **Subversion | Add**. Confirm it by **Subversion | Commit**.

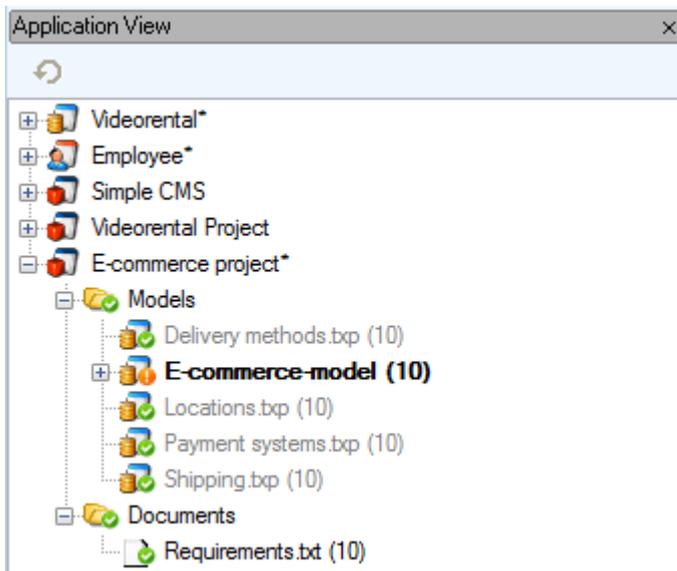
### Other Files in Project

You can add also \*.doc, \*.xls and other file formats to your projects (check them out from subversion etc.) Toad Data Modeler allows you to open them from the **Application View**.

### To open an existing file

Right-click the file name in the **Application View** and select **Open File**.

### Sample User Defined Structure of Custom Project with Version Control System



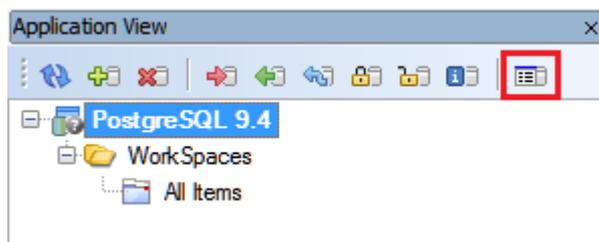
## Legend

-  Added - item was added to subversion, COMMIT is expected.
-  Conflict - something is wrong
-  Current - items in TDM project and subversion are identical
-  Deleted - item was deleted, COMMIT is expected.
-  Missing - item is in subversion and not in TDM project
-  Modified - item was modified in TDM project, COMMIT is expected.
-  No version control
-  Out of date - item was modified in subversion. UPDATE is expected.
-  Unknown - status is unknown, ADD action is expected.

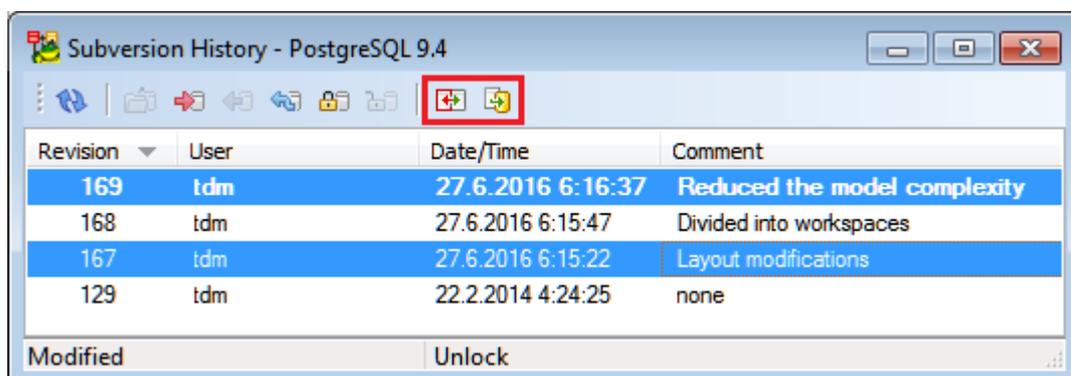
## Direct Comparison

To compare any two revisions of the same model:

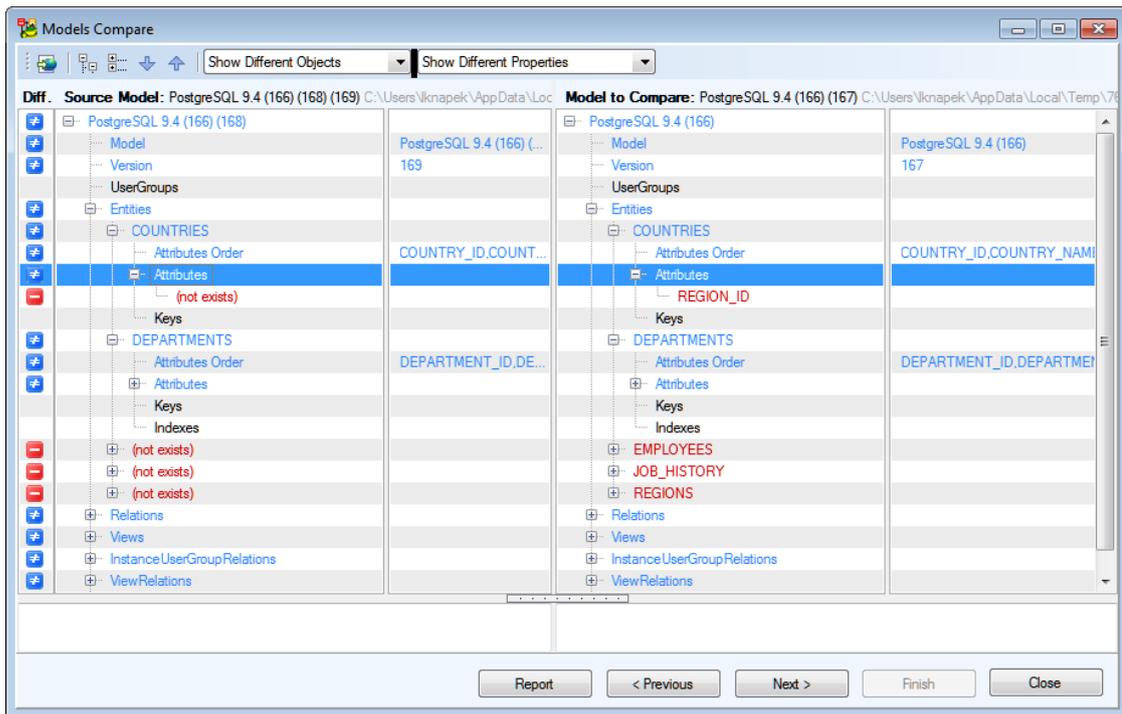
1. In **Application View**, locate your Model tracked by SVN and open **History Browser** either from toolbar, or context menu (**Subversion** | **History**).



2. In **History Browser**, select two of the available revisions and click either on **Compare Revisions** or **Generate Change Script** button on the toolbar.



3. In the opened dialog, you can see the differences between the models.



4. You can now choose an action depending on the button you have clicked on before:

- **Compare Revisions** - a comparison report can be generated by clicking the **Report** button.
- **Generate Change Script** - finish the comparison to generate a change script.

## About Version Manager

Version Manager allows you to create projects, add models (logical models, physical models) and other files (e.g. text files, images etc.) to your projects, create versions and revisions etc.

Toad Data Modeler allows you to create unlimited number of projects.

**i** Note: The version manager is meant to be used by single user only, multiple user collaboration is not supported. Please see Subversion section to learn how to set up multiple user collaboration environment.

### To open Version Manager

1. Enable Expert Mode: select **Settings | Options | General** | select the **Expert Mode** checkbox.
2. Click  on the toolbar (or select **Expert Mode | Version Manager**).

Version Manager window is a place from where you can access all files stored in Version Manager.

**i** TIP: You can open several instances of Version Manager, e.g. for each project.

# What Is a Project?

Project is a group of models and other files that logically match together although they are not of the same origin (physical data model, document file, image etc.).

**Example:** You have created a project for Company "X". This project contains various models (LER, PER models) and some other files (text files, images etc). The models are models of different databases and have a different number of versions/revisions. Nevertheless, all these models and files logically match together - all of them relate to Company "X".

As stated above, projects can contain models as well as other files of any type (e.g. any documentation to models, text files, images, other programs...) Nevertheless, Toad Data Modeler is not able to work directly with such files. It can only call appropriate programs or allows you to view them only (see the **Show Version (only to read)** option).

## Projects and Files in Version Manager

There are several ways how to create a project and add files to it:

- A. Create a new project from scratch (plus possibility to add several files to the project at one jump. - It is recommended when you store all files that you want to add to the new project in one directory.)
- B. Create a new project while adding an ER Diagram to the project. - It is recommended when you want to add a single Toad Data Modeler model to a new or already existing project.
- C. Add a single file to already existing project additionally (no matter if it is a model, text document, screenshot etc.).
- D. Summary - Add a model and file to already existing project.

### A. Create a New Project from Scratch

1. Click  on the toolbar.
2. Right-click the **Local Server** item and select **Add Project**.
3. Define properties of the project (see the following details).

General Tab	Description
Name	Define a project name. The project name has to carry out conditions for file names set in Windows, e.g. question mark "?" cannot be contained. The project name will become a directory name on a disk. See the path where it will be created in the <b>Settings</b> menu   <b>Options   Version Manager</b> . The default path can be: C:\Documents and Settings\user\Application Data\Quest Software\Toad Data Modeler\Installation Name\VersionManager\Server\Project\Project_Name. All files created in Version Manager will be saved here until you change the path.

**i** Note: Information on versions (and their relations to other versions and revisions) are saved by default to another file. Possible path is: C:\Documents and Settings\userName\Application Data\Quest Software\Toad Data Modeler\Installation Name\VersionManager\Server\projects.xml  
Again, you can change the path in the **Settings** menu.

**Load Files in Directory** Uncheck (or not to select) this option to create a new project and let it be empty.

Select this option to enable the **Directory** box. Click the icon on the right to define a path to a directory with files that you want to add to the project. All files stored in the directory will be added to the project automatically. Sub-directories will be ignored.

**Description Tab** You can enter the project description here.

4. If you want to add a group of files to this project, follow the next steps 5 and 6.  
You already need to have all the files stored in one directory.
5. Select **Load Files in Directory** and click the small icon on the right. Find a directory where files that you want to add to the project are stored.
6. Confirm **OK** to load all files stored in the directory to the project.

## B. Create a New Project + Add an ER Diagram to the Project

1. Create a new model or open an existing one.
2. Click  on the toolbar (or **Expert Mode | Version Manager | Add to Version Manager**) to open the **New Version Location** dialog.
3. If no project exists on your local server, click **Add Project** . (And follow step 4.)  
If the project where you want to add the model exists on your local server, simply select it and click **OK** to add the model to the project.
4. The **New Project** dialog opens. Define a name and description of your project.
5. Press **OK** to confirm and turn back to the **New Version Location** dialog.
6. Select the new project and confirm **OK** to add the model to the project.
7. The **Check Out** message displays and you are prompted to define a path where your file should be checked out
8. Confirm **OK**. The file doesn't exist and therefore will be created.
9. Click **Yes**. See the Version Manager now.

## C. Add a Single File to Existing Project

1. Click  on the toolbar to open **Version Manager**.
2. Right-click the selected project and click **Add File** to display a File browser.
3. From the **Files of Type** box, select the appropriate type, find the file and confirm **Open**.

## D. Summary - Add a Model and File to Existing Project

Model: Open the model and simply click  on the toolbar. Select a project and confirm **OK**. (See B.)

Model or any other file: In Version Manager, right-click the selected project and click **Add File**, find the file and confirm **Open**. (See C.)

## Version Manager Toolbar and Options

In Version Manager, the items are sorted this way:

- Projects are listed alphabetically.
- Under projects, files are listed in the alphabetical order too. Under files, their versions and revisions are displayed.
- Versions and revisions of files are sorted in the order they were created. This sorting allows you to see what version precedes which one, what version has been derived from which etc. No other sorting is available in Version Manager itself, however you can sort the items also in the List of Versions and dock the **List** on Version Manager.

### Version Manager Toolbar

Icon	Command
	Check Out
	Check In
	Lock
	Unlock
	Save Version as

These options are active for versions/revisions and also file (if you click a file in Version Manager, the options will relate to its latest version).

## Project Right-Click Options

**Right-click a project to see the following options:**

Option	Description
Lock Project	Locks project not to be overwritten.   Note: Lock option: Generally, there are two colors of padlock in Version Manager. Blue padlock means that you are the person who locked the project/file/version, so only you can modify it. Yellow padlock means that another person locked the project/file/version, so you are not allowed to make any changes in it unless the project/file/version is unlocked by the particular person again.
Unlock Project	Unlocks project.
Add File	Opens a dialog where you select a file that you want to add to the project. Via this option, you can add any file to the project - Toad Data Modeler model(s) as well as any other files.
Remove Project	Removes the selected project including all its files, versions and revisions. If any version/revision is locked, this option is disabled.
Synchronize Project	Synchronizes latest versions of files of selected project with files saved on your local computer. (On the server, the latest versions of files will be found and copied to your local disk.)
Properties	Opens the <b>Project Properties</b> dialog where you can edit a project name, add a description on the project etc.
List of Files	Opens a list of all files of the project. Here, you can sort the files by name, date of creation, owner etc.

## File Right-Click Options

**Right-click a file to see the following options:**

Option	Description
Lock File	To preserve the selected file from overwriting, select this option.
Unlock File	Unlocks previously locked file.
Last Version Check-Out	Opens the latest version for edit.
Last Version Check-In	Saves changes made in the latest version.
Add Version	You have modified a file, saved it and now you want to add it as a new

Option	Description
(2.0)	<p>version. For this purpose, select this option, and find the appropriate file in the <b>Open</b> dialog.</p> <p><b>i</b> Note: This option is available even though a project is locked, nevertheless, only provided that a user who's locked it and user who is logged in Version Manager is the same person. (If it was Administrator who locked the file and a User was logged in, the <b>Add Version</b> option would be disabled.)</p>
Remove from Project	Removes a file from project. This option is not available if any version or revision of this file is locked.
Save Version as	Saves the latest version as a standard file Toad Data Modeler models with extension .txp or .txl).
Properties	<p>Opens the <b>File Properties</b> dialog where information on the file name, location, date and time of creation and last modifications can be found. Tab <b>Lock</b> is read-only and provides information on the lock hierarchy:</p> <p><b>Ancestor Locked</b> - It's selected if ancestor (Project in this case) has been locked.</p> <p><b>Descendent Locked</b> - It's selected if any descendent of this file has been locked.</p>
List of Versions	Displays list of versions and revisions of the selected file.

## Version/Revision Right-Click Options

**Right-click a version/revision to see the following options:**

Option	Description
Lock Version	To prevent the selected version from overwriting, select this option. (If the version is checked out, the version is locked automatically.)
Unlock Version	Unlocks previously locked version. (If you check in the version, it will be unlocked automatically.)

### How does it work?

If you check out a version, the version locks and the corresponding file on the server becomes read-only, however it is editable in your local file.

When you check in the version, the version unlocks and the corresponding file on the server can be modified. In your local file, it is locked and becomes read-only.

Version lock properties show information on who locked the project and when.

**i** Note: For now, all users in Version Manager are Admins.

Check Out	<p>Opens the selected version/revision for edit.</p> <p><b>i</b> Note: Multiple Version Check Out is possible. - Use SHIFT key to select versions and click <b>Check Out</b> then.</p>
-----------	--

Option	Description
Check In	<p>Saves changes and closes the version/revision. This option is available from the pop-up menu of the selected version in Version Manager and also in the Application View.</p> <p><b>Description</b> - here, you can write description on the version/revision.</p> <p><b>Finish Work on Model</b> - select it to close your model during the Check In. Otherwise it remains open.</p> <p>After you confirm <b>OK</b>, the version will be saved to your local disk and possible changes will be applied on a server. Version will be automatically unlocked and will become accessible for other team members. (Projects.xml file is saved after every change made in the project. The file is being updated continuously - changes made and saved by user A will be visible to user B.)</p> <p> Note: Multiple Version Check In is possible. - Use SHIFT key to select versions and click <b>Check In</b> then.</p>
Show Version (only to read)	<p>Opens the selected version in appropriate associated program, e.g. Word, Notepad, Windows Viewer, Acrobat Reader etc. Toad Data Modeler models will open in the Application Window.</p> <p>If your file has an extension that does not associate with any program, it will open in the <b>Version Viewer</b> dialog. On tab <b>Content of Local File</b>, you can see the text.</p>
Add Version (2.0)	Adds another version.
Add Revision (1.1)	<p>Adds another revision.</p> <p>(If you checked out a version and saved the changes to new a version or revision, the result would be the same as if you used these options.)</p>
Remove Version	Removes the selected version and all its revisions.
Exclude Version	Excludes only the selected version, its revisions will remain.
Save Version as	Saves the selected version as a standard file (Toad Data Modeler models with extension .txp or .txl).
Properties	Opens the <b>Version Properties</b> dialog. On tab <b>Notes</b> , you can define notes on the version/revision.

## List of Projects, Files, Versions

The **List** displays information on items in Version Manager - projects, files and versions. Here, you can sort the items by different conditions (by name, version number, date of creation etc.), however you are not able to edit them.

The **List** opens together with Version Manager. You can dock it wherever you want. If you close it, you can find it later in appropriate pop-up menus of items in Version Manager. E.g. Right-click the server and select **List of Projects**

**i** Note: You don't have to close the **List of Projects** to open **List of Files**. To see all files of the selected project, simply click the project in the Version Manager tree and the List of Projects will change to List of Files automatically. To see all versions/revisions of a file, simply click a file in Version Manager tree to display the **List of Versions**.

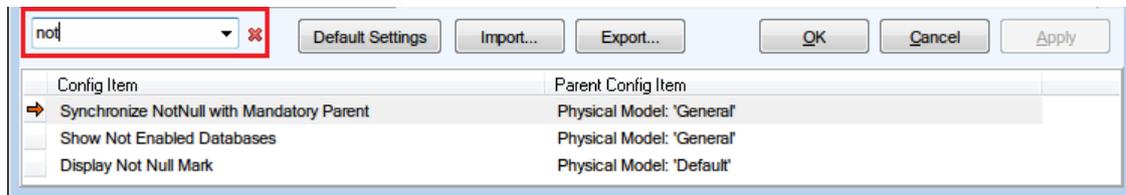
Click a column to sort the data by:

- **Name** - in alphabetical order
- **Owner**
- **Created** - chronologically by date of creation
- **Changed** - chronologically by date of a change
- **Locked** - locked items will be moved at the beginning or at the end of the list
- **Lock Time** - chronologically by time of locking an item
- **Description** - items with description will be moved at the beginning or at the end of the list

# Options

The **Options** contain most of the settings available in Toad Data Modeler. This topic divides all settings into categories and describes what each individual option controls.

**TIP:** To find a specific setting quickly, use **Search Inbox**



There are two kinds of Options:

- **Default Options** (most of the Model related settings) - these apply to newly created Models only, they do not influence existing Models
- **Other Options** (most of the general settings) - these are changeable at any time, the changes are applied immediately

The main Options Categories are:

- **Application Options** - affect the application behavior and user interface
- **Model Options** - affect every existing or created Model

The Options categories are:

- **General**
- **Graphics**
- **Dialog Boxes**
- **Paths**
- **Reverse Engineering**
- **Reports**
- **Scripting**
- **TOAD for Oracle Integration**
- **Version Control System**
- **Print**
- **Expert Mode**
- **Physical Model**
- **Specific Databases**
- **Logical Model**

The **Import and Export of Settings** is discussed at the end of the topic.

# Application

## General

Option	Description
Open Only One Instance of Forms	With this option unchecked it is possible to have multiple identical forms opened at the same time (such as various object properties, model actions wizards, etc...). When checked, TDM only opens one instance of a form and instead of opening another form, the one already opened is focused.
Expert Mode	Turns <b>Expert Mode</b> on/off. When on, several new options become available. Recommended for power users and developers.
Show Hints in Designers	When checked, shows objects notes and description on mouseover on Workspace.
Run Only One Instance of Application	Checked means you can open only one instance of Toad Data Modeler. When unchecked, you can open multiple instances of the application.
Open Workspace Properties Dialog after Add Workspace	Determines if <b>Workspace Properties</b> dialog will be shown every time you add a new workspace.
Save Models and Projects as Formatted XML Files	Models saved in XML format are by default not very readable when opened in editor. Checking this option will cause the models to be saved in more comprehensible format.
Always Use CTRL to Select Attribute	When holding CTRL key, you can click on attributes in entities to select them. Unchecking this option allows you to do this without holding CTRL. The downside is that you need to drag entities by their headers or whitespace to move them.
Use Shift to Dock Forms	While moving forms, Toad Data Modeler automatically suggests you docking positions. If you dislike this behavior, you can uncheck this option. In this case docking positions will be suggested only when you hold SHIFT during moving a form.
General Font	Sets the font application uses in most of the user interface. Doesn't include menus.
Select Unit of Length	You can choose between using millimeters or inches. This choice is reflected in many dialogs which contain any kind of size.
Icons Theme	You can switch between Toad Data Modeler and Toad for Oracle icon themes.
Number of "Undo/Redo" Steps	Sets the number of Undo/Redo steps available. Higher amounts require slightly more system memory.
Toolbars and Menu Style	There are several graphical styles available for Menu and Toolbars in TDM.
Toolbar Rows	Sets the number of rows dedicated to Toolbars.

Option	Description
Recent Files Count	Sets the number of recent files TDM remembers (these are files opened in the past, accessible in <b>File Menu   Recent Files</b> )

## Graphics

### General tab

Option	Description
Default	
<ul style="list-style-type: none"> <li>Mouse Scroll Speed</li> </ul>	Sets the scroll speed for scrolling in Workspace.
<ul style="list-style-type: none"> <li>Minimal Size of Displayed Grid</li> </ul>	Determines how large/small the grid must be to be displayed at various zoom levels.
<ul style="list-style-type: none"> <li>Grid Size</li> </ul>	Sets the grid size.
<ul style="list-style-type: none"> <li>Grid On</li> </ul>	Enables/disables Grid.
<ul style="list-style-type: none"> <li>Grid Visible</li> </ul>	Shows/hides Grid. The Grid may be enabled (and objects will still snap to it), but it will be invisible.
Move Objects by	Sets the distance the objects are moved by when using arrow keys.
Join Line Distance	Sets the distance needed for a part of a line to be snapped to another part of a line with the same orientation (horizontal, vertical).
Snap to Objects	When checked, TDM automatically aligns moved objects to other objects on the workspace using guidelines.
Snap to Objects Distance	Sets the distance from an object to the nearest guideline needed to activate Snap to Objects.
Visible Page Boundaries	Shows/hides page boundaries.
Visible Page Numbers	Shows/hides page numbers.
Max. Number of Characters for Displayed Data Type	Some enumeration or user data types might be very long and if the Recalculate Size option is turned on, the entity containing them can become wide. This option sets the maximal number of characters shown in the entity and therefore limits the entity maximum width.
Hints in Designer	Controls what is displayed in a popup hint in both physical and logical models. Notes are displayed for physical models and descriptions for logical models. <ul style="list-style-type: none"> <li>Disable - no popup hints are displayed</li> <li>Notes - notes/descriptions are displayed</li> <li>Technical Notes - technical notes/technical descriptions are displayed</li> </ul>

## Colors tab

- Colors used in Toad Data Modeler can be set here.

## Autolayout tab

Option	Description
Shapes	
<ul style="list-style-type: none"><li>• Horizontal Distance Coefficient</li></ul>	Sets the horizontal distance coefficient for left-to-right and top-to-down autolayout.
<ul style="list-style-type: none"><li>• Vertical Distance Coefficient</li></ul>	Sets the vertical distance coefficient for left-to-right and top-to-down autolayout.
<ul style="list-style-type: none"><li>• Alphabetic Autolayout - Sort By</li></ul>	When using Alphabetic Autolayout, this option determines if the objects are sorted by their Name or Caption.
Lines   Autolayout	
<ul style="list-style-type: none"><li>• Max. Calculated Variants on Shape</li></ul>	Sets the number of variants Toad Data Modeler generates. Generally, the higher the number, the better the final result and also the more resources used.
<ul style="list-style-type: none"><li>• Minimum Distance Between Lines</li></ul>	Sets the offset distance between lines, both horizontal and vertical.
<ul style="list-style-type: none"><li>• Cross Lines</li></ul>	When checked, Toad Data Modeler prefers crossed lines variants.
<ul style="list-style-type: none"><li>• Straight Lines</li></ul>	When checked, Toad Data Modeler prefers straight lines variants.
Lines   Optimal Styl On Move	
<ul style="list-style-type: none"><li>• Max. Calculated Variants on Shape</li></ul>	Sets the number of variants Toad Data Modeler generates every time an object is moved. Generally, the higher the number, the better the final result and also the more "choppier" moving objects is.
<ul style="list-style-type: none"><li>• Minimum Distance Between Lines</li></ul>	Sets the offset distance between lines, both horizontal and vertical.
<ul style="list-style-type: none"><li>• Cross Lines</li></ul>	When checked, Toad Data Modeler prefers crossed lines variants.
<ul style="list-style-type: none"><li>• Straight Lines</li></ul>	When checked, Toad Data Modeler prefers straight lines variants.

## Page Setup tab

- This tab contains options for default page setup configuration for printing.

Option	Description
Size	Allows you to select among common page sizes or even define your own.
Orientation	Choose between <b>Portrait</b> and <b>Landscape</b> paper orientation.
Margin	Allows you to define page margin.

## Dialog Boxes

### Hidden Dialog Boxes tab

- Most dialog prompts have a **Do Not Show Next Time** checkbox. When you check it, the specific dialog won't be shown anymore. It will also appear in this tab, so if you change your mind, you can allow it again by unchecking it.

### Other tab

- You can configure **Delete in Designer** dialog here. It determines what action will be performed by default when you delete an object on workspace.

Option	Description
Display Dialog	A dialog will be shown where you can choose to delete an object entirely, or only delete its graphical representation.
Remove Graphical Representative of Object	Removes the selected object from workspace, but not from model itself.
Delete Object	Deletes the selected object entirely.

## Paths

- This section contains default paths where Toad Data Modeler looks for objects and where objects are saved.

## Reverse Engineering

Option	Description
Path to Connections	The path where Toad Data Modeler saves all configured Connections.
Save Passwords with Connections	Sets the default behavior when creating Connections. If unchecked, you will be prompted to enter the password every time you work with a Connection.
Load Orphaned FK Constraints to Child Table After Script	If you Live Reverse Engineer a child entity without its parent entity, this function will create a SQL code for generating the foreign key constraint and save it into the entity After Script.

Option	Description
Check After Script	If you Live Reverse Engineer a parent entity and you already reversed the child entity, Toad Data Modeler will use the SQL code mentioned in the previous example and regenerate the foreign key constraint. Unchecking this option means the child entity after script won't be checked.
Search	
<ul style="list-style-type: none"> <li>Auto Hide Items</li> </ul>	Automatically hides all objects that do not match the search criteria.
<ul style="list-style-type: none"> <li>Search Delay</li> </ul>	Sets the delay between entering a term into Filter and actually filtering the objects.
Load Selection	Determines the behavior of your custom saved selection, if there are new items found and not included in it. They may be either added to the selection, excluded from it, or you may be alerted when such situation happens and decide for yourself.

## Reports

- You can set default file name and preferred language of generated reports here.

## Scripting

Option	Description
Show Windows Automatically	Shows Scripting Window parts when needed (e.g. Log when executing a script).
Show Log	Shows/hides log in Scripting Window.

## Toad for Oracle Integration

Option	Description
Toad for Oracle Version	Navigate to the folder, where you have installed Toad for Oracle. Toad Data Modeler will automatically detect its version and default paths.
<ul style="list-style-type: none"> <li>Load Toad for Oracle Connections as Connection in Toad Data Modeler</li> </ul>	If checked, all Connections stored in Toad for Oracle will be made available in Toad Data Modeler.
Use Toad for Oracle Icons in Toad Data Modeler	Check to use Toad for Oracle icons in Toad Data Modeler. This option is also available in the General section of Options.
Detect this Toad Data Modeler version via Toad for Oracle	When enabled, Toad Data Modeler creates a registry entry, which allows Toad for Oracle to identify it easily.

## Version Control System

Option	Description
General	
<ul style="list-style-type: none"><li>Type</li></ul>	Decide if you want to use Subversion, or no Version Control System.
<ul style="list-style-type: none"><li>User Name</li></ul>	User name of Version Control System user.
<ul style="list-style-type: none"><li>Password</li></ul>	Password of Version Control System user.
<ul style="list-style-type: none"><li>Working Directory</li></ul>	Enter path to the Working Directory.
<ul style="list-style-type: none"><li>Use Read Only Locking Mechanism</li></ul>	Flags all unlocked files as "Read Only".
Subversion	
<ul style="list-style-type: none"><li>SVN Client Path</li></ul>	Enter path to the SVN client of your choice.
<ul style="list-style-type: none"><li>Server Path</li></ul>	Enter address of your SVN server.
<ul style="list-style-type: none"><li>Non Interactive</li></ul>	When checked, disables interactive prompts in SVN (e.g. authentication credentials, conflict decisions).
<ul style="list-style-type: none"><li>No Authentication Cache</li></ul>	Doesn't store passwords in authentication cache (asks for user password every time).

## Print

Option	Description
Default	
<ul style="list-style-type: none"><li>Print Page Numbers</li></ul>	Include/exclude page numbers in printed document.
<ul style="list-style-type: none"><li>Print Frame</li></ul>	Prints/doesn't print a frame around the printed ER diagram.
<ul style="list-style-type: none"><li>Print Only Black and White</li></ul>	If checked, prints the diagram in black and white only.
Print Gradients	If checked, gradients in model objects (mostly in entities) will be printed. Uncheck to save some ink during printing.

## Expert Mode

Option	Description
Save the definitions to the 'My Package'	By default you can choose the package where you save

Option	Description
	customized forms definitions. With this option checked, all of these definitions will be saved to 'My Package'.
Allow to Modify System Selected OTPs	Checking this option allows you to edit default System selection of OTPs which is used in many dialogs such as generation of DDL scripts, reports, change scripts...
Check Dictionaries When Generating Reports	This function checks if there are not any missing terms in Dictionary which is used for report generation before generating the report itself.
Old Look of IE Notation	When checked, Toad Data Modeler uses the old look of the objects, assuming IE notation is currently used.
Work with System Dictionary	Checking this option allows you to edit the default English System dictionary.
Support for Import Old Dictionaries	Check to support importing dictionaries from older versions of Toad Data Modeler (older than 5.0).
Eureka Log	A log created whenever the application crashes. Contains information useful to the product developers.
<ul style="list-style-type: none"> <li>Freeze Activate</li> </ul>	When checked, an Eureka log is created after the application freezes for longer than the <b>Freeze Exception Timeout</b> is set.
<ul style="list-style-type: none"> <li>Send Email</li> </ul>	When checked, an email to developer team with Eureka log attached is created in case the application crashes or freezes.
Installation Information	Contains settings that are configured during the first launch of the program. You can change the path where Toad Data Modeler stores its configuration files or change the <b>Installation Name</b> and <b>Installation Number</b> .
Benchmark	When checked, measures the duration of some of the application actions and logs results to the Message Explorer.

## Expert Mode | Version Manager

- Paths to Version Control server and client files and projects can be set here.

## Expert Mode | Editable Forms

### Design Mode tab

- You can choose which windows should be visible when in Design Mode.

### Component Palette tab

- Shows small/large buttons in Component Palette when in Design Mode.

### Object Inspector tab

- Contains options to configure Object Inspector when in Design Mode.

### Form Explorer tab

- Has several options to configure Form Explorer when in Design Mode.

## Expert Mode | Eclipse

Option	Description
Eclipse Support On	With Eclipse support enabled, a new "Open in Eclipse" button appears when writing scripts. It allows you to write your scripts in Eclipse and then import the result back to Toad Data Modeler.
Path to Eclipse	Enter the path to your Eclipse folder which contains <b>eclipse.exe</b> file.
Path to Working Directory	Enter the path to your Javascript Project directory (created in Eclipse).
Delete Working File After Close Script in Toad Data Modeler	If checked, deletes script which has been closed in Toad Data Modeler from Working Directory.
Associate Eclipse as Default SQL Editor	Click a specific button to associate Eclipse as the default editor for Oracle/MySQL/PostgreSQL files.

## Model

Option	Description
New Model	
<ul style="list-style-type: none"> <li>• Default Model Type</li> </ul>	The model type selected here will be highlighted every time you create a new model, which means you only need to click OK to create it.
<ul style="list-style-type: none"> <li>• Last Model Type as Default</li> </ul>	Uses last model type as the default model type.
Model Verification	
<ul style="list-style-type: none"> <li>• Verification on Forms</li> </ul>	Allows you to choose what should be shown in <b>Form Verification</b> or disable it entirely (See <b>Projects and Models   Models   Physical Data Models   Model Verification</b> for more information).
<ul style="list-style-type: none"> <li>• Max number of messages</li> </ul>	Limits the maximum number of messages shown in <b>Form Verification</b> .
Other Settings	
<ul style="list-style-type: none"> <li>• Restore Last Open Models at Startup</li> </ul>	If enabled, open models from last session will be restored at startup.

## Logical Model

### General Tab

<b>Option</b>	<b>Description</b>
Auto Complete Workspaces	When checked, applies to all newly created workspaces. If you create an object on any workspace, this action will be executed on the other workspaces in the model with this function enabled as well.
Display Line Names	Shows/hides relationship captions.
Brush Color	Defines the main color of the newly created objects.
Pen Color	Defines the border color of the newly created objects
Background Color	Defines the background color of the newly created objects.
Pen Width	Sets the width of the newly created objects border.
Font	Sets the default font for all text in newly created objects.
Display Mode	Sets the default display mode for newly created workspaces.

### Shape tab

<b>Option</b>	<b>Description</b>
Recalculate Size	This function automatically resizes an object so it fits the length of the text contained inside. Checking/unchecking this option will activate/deactivate the function in newly created workspaces.
Shadow Effect	A shadow effect is applied to all objects. Checking/unchecking this option will activate/deactivate the shadow effect in newly created workspaces.
Use Brush Color for Full Shape	Uses brush color instead of background color in objects. Checking/unchecking this option will activate/deactivate this function in newly created workspaces.

### Note Line tab

<b>Option</b>	<b>Description</b>
End Type 1, 2	Sets the default note line end types in newly created workspaces.

### Entity tab

<b>Option</b>	<b>Description</b>
Display Level	Sets the default display level in newly created workspaces.
Align	When checked, entities in newly created workspaces will have their attributes aligned.
Display Data Types	Shows/hides the data types of attributes in newly created workspaces.
Display Keys Graphically	Shows/hides the key icons in entities in newly created workspaces.

Option	Description
Display Unique Identifier Mark	Shows/hides the unique identifier mark indicator in entities in newly created workspaces.
Display Mandatory Mark	Shows/hides the mandatory mark in entities in newly created workspaces
Gradient Effect	Uses/doesn't use gradient effect in objects in newly created workspaces.
Display Domains	Shows/hides the domains in entities in newly created workspaces
Description width (px)	Enter maximum width of descriptions in pixels. Descriptions exceeding this limit are broken into more lines
Attribute Colors	This section allows you to define your own colors for all kinds of attributes.

### Verification tab

- This tab contains **Model Verification** criteria. You can enable/disable them and Model Verification will/won't use them.

## Physical Model

### General Tab

Option	Description
Show Enabled Databases Only	Shows only enabled databases in several dialogs.
Word Wrap for SQL Preview	Enables/disables Word Wrap in SQL Preview. Useful when TDM generates long SQL statements.
Word Wrap for SQL, Before Script and After Script	Enables/disables Word Wrap in Before and After script.
Synchronize NotNull with Mandatory Parent	When checked, Mandatory Parent in Relationship Properties is synchronized with Not Null property in child entity Foreign Key.
Allow Null Attributes in Keys	When checked, you can create a Key without the Not Null property. This setting applies to <b>Universal Model</b> as well.
Alphabetic Order in Attribute Navigator List	Defines if the Attributes in Object Navigator Dropdown Menu in Entity properties should be alphabetically sorted, or not.
Inverse Relation Name	Allows you to set Inverse Relationship Name and Caption in Relationship Properties.
Show Non-printable Characters in SQL Editors	Shows/hides non-printable characters marking end of a line in SQL Editors.
Index to Foreign Key	Binds indexes to Foreign keys located in Child Entities.

Option	Description
in Child Entity	
Index to Foreign Key Name	Sets the default name of an index bound to a Foreign Key in Child Entity.
Attribute Properties Propagation	Defines what Attribute properties should migrate when an Attribute migrates to a child entity.
Self Relation Attribute Name	Sets the default name of the foreign key attribute in self-relationship. Application variables can be used.
Self Relation Attribute Caption	Sets the default caption of the foreign key attribute in self-relationship. Application variables can be used.
Relation Attribute Name	Sets the default name of the foreign key attribute in relationship. Application variables can be used.
Relation Attribute Caption	Sets the default name of the foreign key attribute in relationship. Application variables can be used.
Automatic FK Mapping	TDM can automatically map appropriate Foreign Key or create a new Foreign Key or prompt you to decide whenever both choices are available.
Primary Key Default Name	Sets the default name for primary keys. Application variables can be used.
Primary Key Default Caption	Sets the default caption for primary keys. Application variables can be used.

#### Generation SQL Script tab

Encoding Used for SQL Scripts	You can select which encoding should be used in all generated SQL scripts.
Code Editor Type	Defines what Editor Type should be used to open generated DDL and Change Scripts. <ul style="list-style-type: none"> <li>Internal - scripts are opened in TDM itself</li> <li>Associated Application - scripts are opened in application which is associated to the scripts format</li> <li>Custom editor - scripts are opened in an user defined editor</li> </ul>
External Editor for Generated Code	Enter the path to your custom editor for opening SQL scripts if you checked "Use Custom Editor" in previous option.
Verification	You can enable/disable <b>Model Verification</b> before SQL Script generation and choose in which cases should the Verification alert you.

#### Verification

Option	Description
Error Row	Defines the color used to highlight background of error text.

Option	Description
Background	
Error Item Text	Defines the color used to highlight the error text.

### Workspace Tab

Option	Description
Auto Complete	When checked, applies to all newly created workspaces. If you create an object on any workspace, this action will be executed on the other workspaces in the model with this function enabled as well.
Display Line Names	Shows/hides relationship captions.
Brush Color	Defines the main color of the newly created objects.
Pen Color	Defines the border color of the newly created objects
Background Color	Defines the background color of the newly created objects.
Pen Width	Sets the width of the newly created objects border.
Font	Sets the default font for all text in newly created objects.
Display Mode	Sets the default display mode for newly created workspaces.

### Shape tab

Option	Description
Recalculate Size	This function automatically resizes an object so it fits the length of the text contained inside. Checking/unchecking this option will activate/deactivate the function in newly created workspaces.
Shadow Effect	A shadow effect is applied to all objects. Checking/unchecking this option will activate/deactivate the shadow effect in newly created workspaces.
Use Brush Color for Full Shape	Uses brush color instead of background color in objects. Checking/unchecking this option will activate/deactivate this function in newly created workspaces.

### Note Line tab

Option	Description
End Type 1, 2	Sets the default note line end types in newly created workspaces.

### Entity tab

Option	Description
Display Level	Sets the default display level in newly created workspaces.
Align	When checked, entities in newly created workspaces will have their attributes aligned.
Display Data Types	Shows/hides the data types of attributes in newly created workspaces.
Display Dictionary Types as Data Types	Switches between displaying Dictionary Types or Data Types on which Dictionary Types are based in the entities.
Display Keys Graphically	Shows/hides the key icons in entities in newly created workspaces.
Display Key and Index Marks	Shows/hides the index and key marks in entities in newly created workspaces.
Display Indexes	Shows/hides indexes in entities in newly created workspaces.
Display Not Null Mark	Shows/hides not null (NN) marks in entities in newly created workspaces.
Gradient Effect	Uses/doesn't use gradient effect in objects in newly created workspaces.
Attribute Colors	This section allows you to define your own colors for all kinds of attributes.
Display Data Warehouse Type and Size	Checking this option will make several new options in Entity Properties available. These are all logical and used to organizing your model if you plan to use it for Data Warehouse.

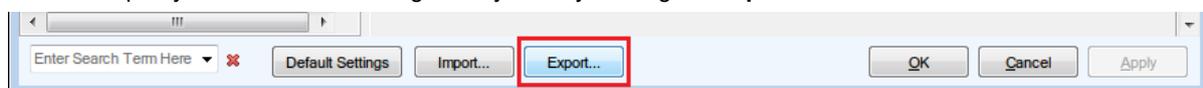
## Specific Databases

- Specific databases have their own settings, for more information see chapter **Databases**.

# Import and Export of Settings

## Export

You can export your customized settings at any time by clicking the **Export** button.

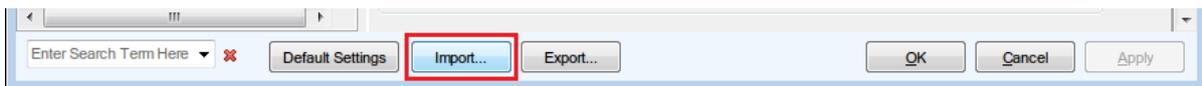


Choose a destination folder and Toad Data Modeler exports several `.txc` files to it. These are configuration files, one file matches one category in **Options**. The more categories settings you change, the more files will be exported.

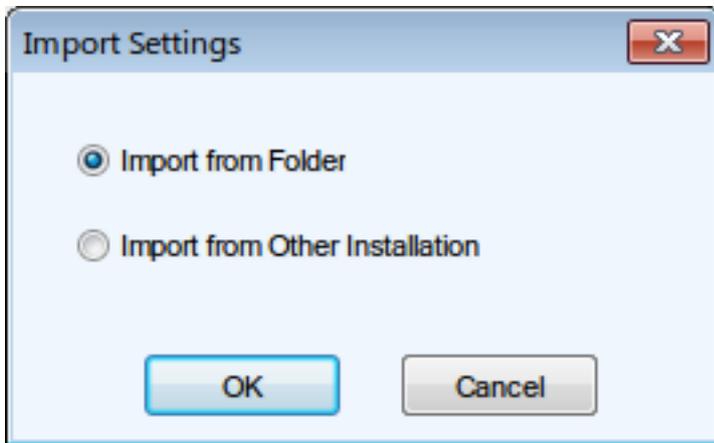
Name	Date modified	Type	Size
 Application.txc	3/16/2015 10:41 AM	Toad Data Modele...	9 KB
 ApplicationServer.txc	3/16/2015 10:41 AM	Toad Data Modele...	1 KB
 Debug.txc	3/16/2015 10:41 AM	Toad Data Modele...	1 KB
 LER.txc	3/16/2015 10:41 AM	Toad Data Modele...	4 KB
 MD.txc	3/16/2015 10:41 AM	Toad Data Modele...	1 KB
 Model.txc	3/16/2015 10:41 AM	Toad Data Modele...	1 KB

## Import

To load your exported customized settings, click the **Import** button.



You can either import settings from a folder or from an existing Toad Data Modeler installation.

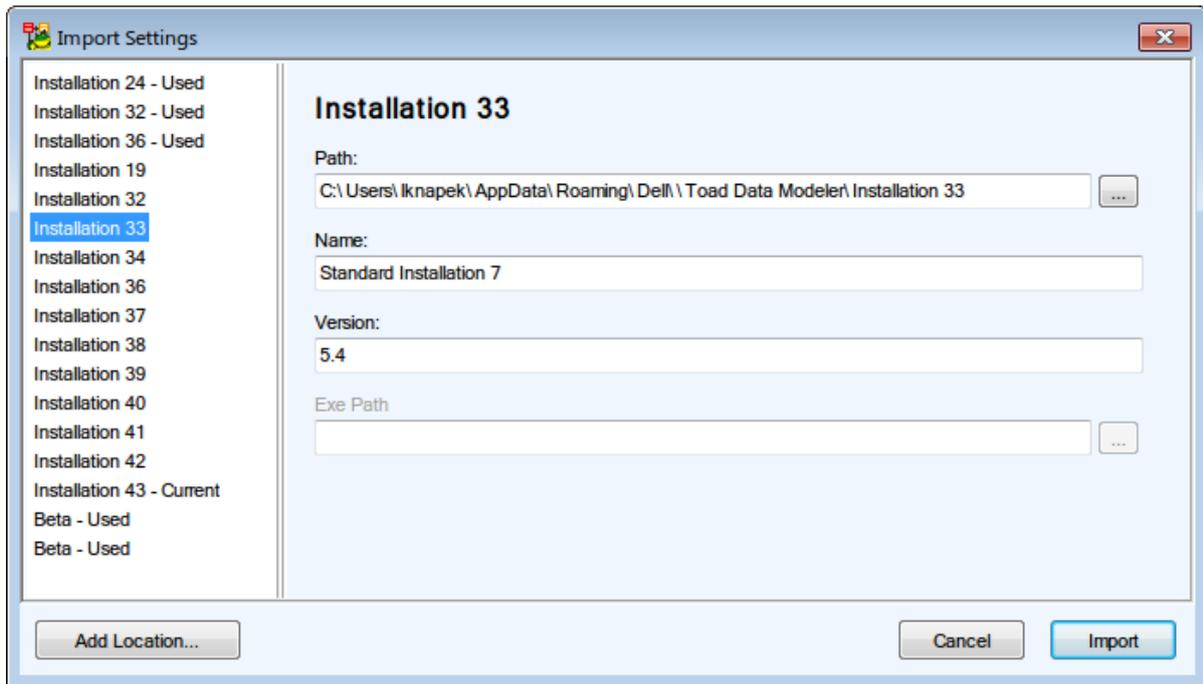


### Import from Folder

Select the **Import from Folder** option and in the following dialog navigate to your folder where your .txc files are stored. Toad Data Modeler will then import the configuration.

### Import from Other Installation

**Import Settings** dialog is displayed. You can choose from which Toad Data Modeler installation should the settings be imported. If your installation directory is not shown, you can add it manually by clicking **Add Location**.



## Default Values

Toad Data Modeler allows you to define and change default values of objects in your models.

### Examples:

- You want to define default values for referential integrity type in your model.
- You want to define Not Null property for all new attributes that you create in your model.
- You want to define a name for your relationships in the following format: 'parent table - child table'.

**i** TIP: You can use application variables in default values. [Application Variables and Default Values](#)

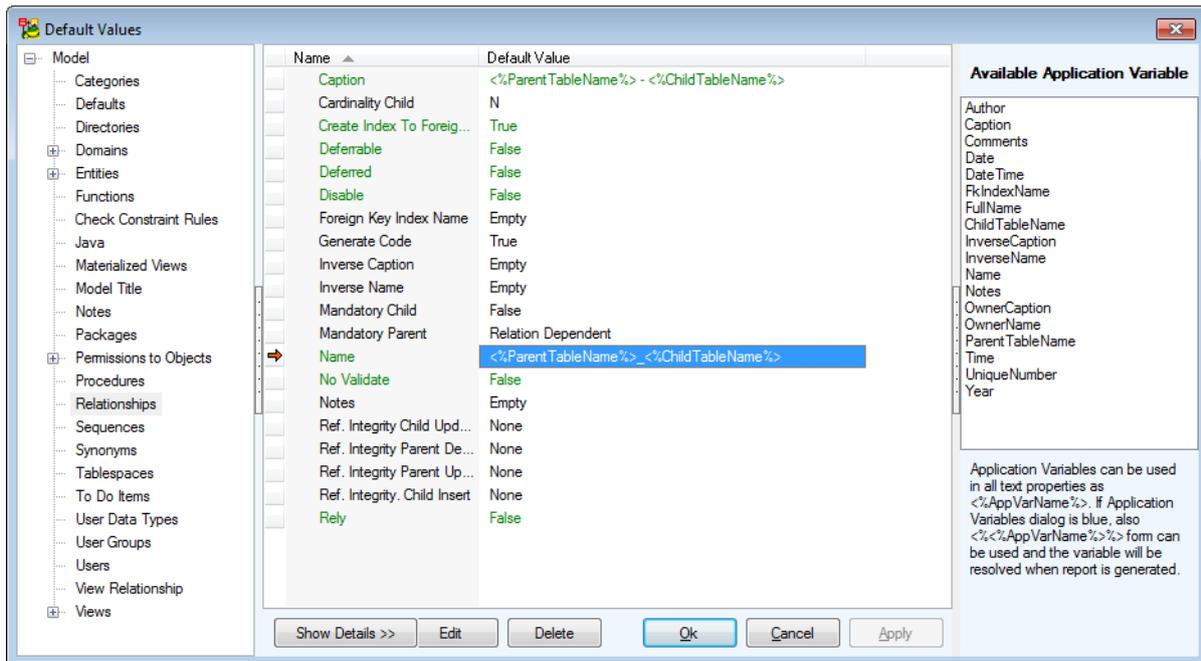
### To open the Default Values dialog

Select **Settings | Default Values...**

or

Open the **Properties** dialog of specific object (e.g. **Relationship Properties** form) | right-click the form and select **Default Values for Class**.

**i** Note: The second option is available only in Expert Mode.



### To define a default value e.g. for relationships in your Oracle 10g model

1. Open your Oracle 10g model.
2. Select **Settings | Default Values**.
3. Select the **Relationships** item in the list.
4. Select the default value you want to modify.
5. Press F2 in the **Default Value** column.
6. Define the value and click anywhere else in the dialog.
7. Confirm **OK**.
8. Restart Toad Data Modeler.

**i** Note: When you change the default values, you make modifications in package (by default in *My Package.txg* file). So, to apply the changes, you need to save the package.

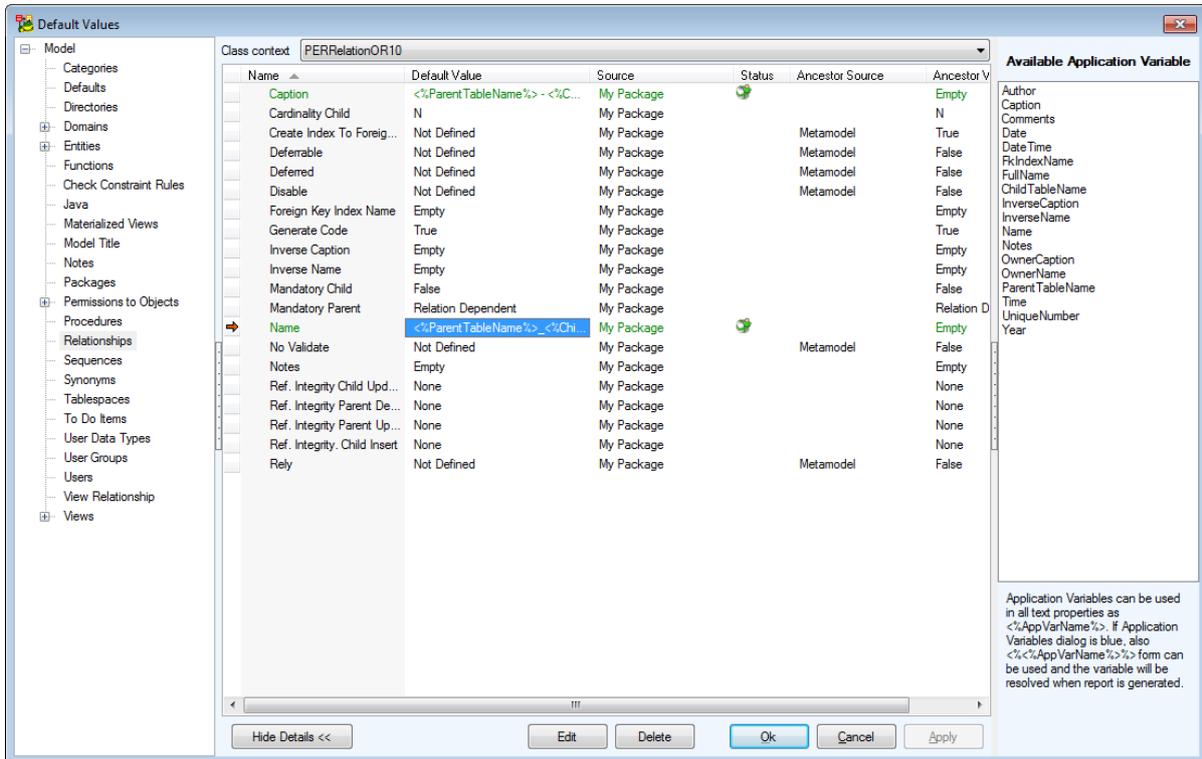
9. Create a new relationship. - All newly created relationships will have the new default value.

### To restore the original default values

1. Open the **Default Values** dialog.
2. Select the default value that you want to change back to original. (For multiple selection use CTRL or SHIFT.)
3. Click **Delete** to restore the original default values.
4. Confirm **OK**.
5. Restart Toad Data Modeler.

# Default Values Dialog

See the dialog for the Entities item and after the **Show Details** button has been clicked.



## Option

## Description

Model Objects

List of objects for which you can define or modify default values.

Class Context

Level on which you want to apply the default values.

**Example:** PEREntityOR10 (entities in Oracle 10g models only), PEREntityOR (entities in all Oracle models), PEREntity (entities in physical model of any database.)

Name

Name of default value

Default Value

Values of specific default values. Press F2 to change the selected default value.

Source

A place where changed or newly modified default values are saved. By default they are saved to *My Package.txg* file.

Status



- Default values stored in Metamodel.



- Default values defined or modified by user, saved in *My Package*.



- Default values defined or modified by user, saved in add-on

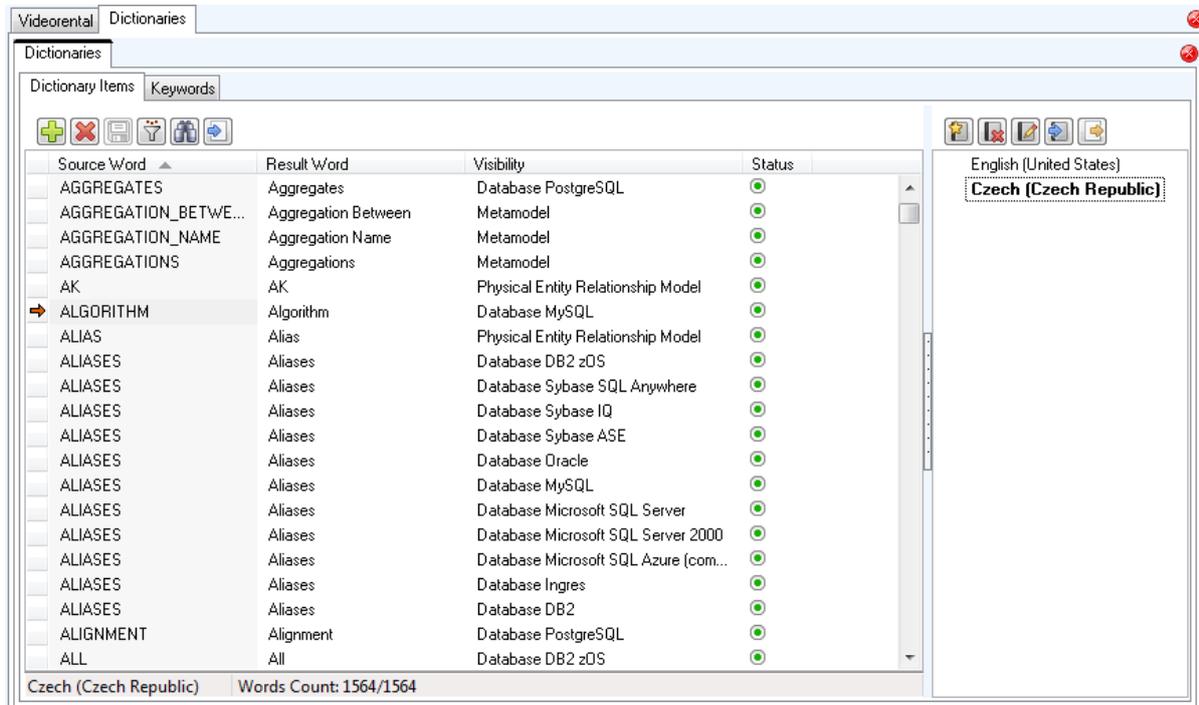
Option	Description
	package.
Ancestor Source	<p>Add-on Package - A place where some default values are stored. Such default values can be modified, however the changes can be saved only to <i>My Package</i>.</p> <p>Metamodel - A place where some default values are stored. Such default values can be modified, however the changes can be saved only to Metamodel.</p>

## Dictionarys

Dictionary allows you to add and translate new terms in your current dictionary from other dictionaries, import web-based dictionaries and export dictionaries to the web (in CSV file format).

### To open Dictionarys

Select **Settings | Dictionarys**.

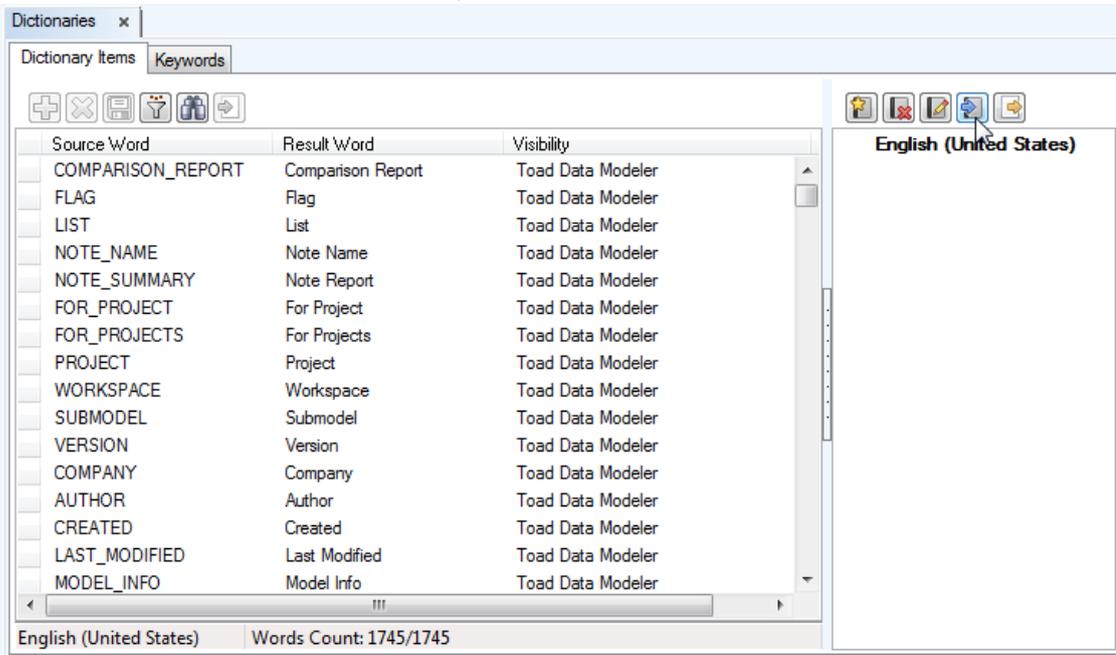


**i** TIP: To modify system dictionaries (such as the default English one), you need to enable **Work with System Dictionary** option in **Settings**.

# Localized HTML, RTF and PDF Reports

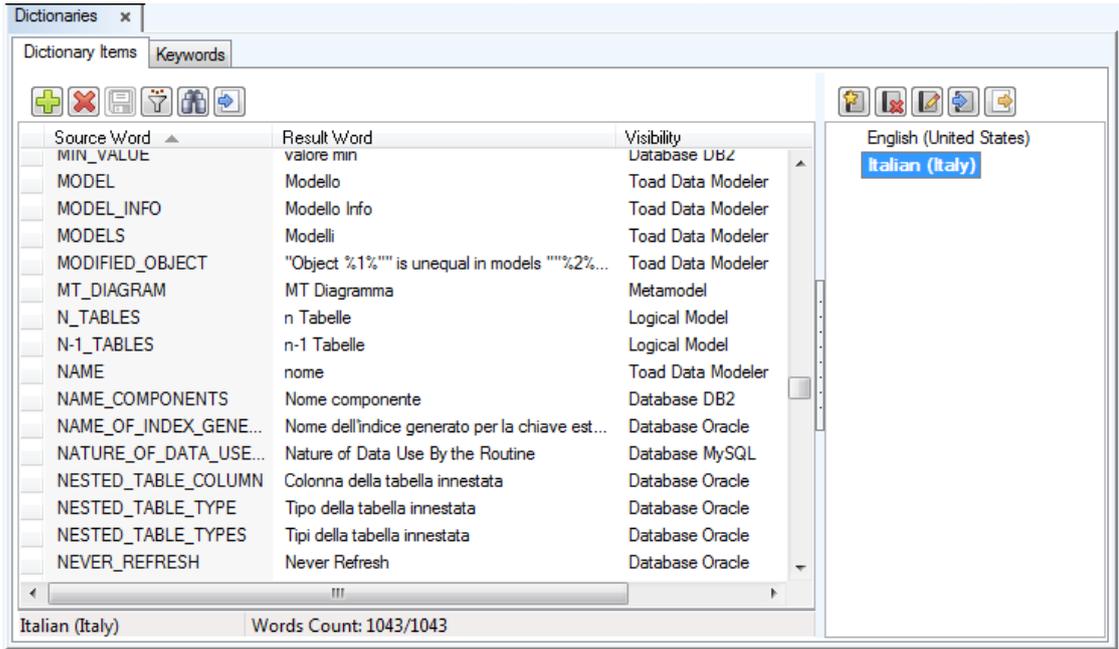
## To generate localized HTML, RTF or PDF reports

1. Download dictionaries from [community website](#).
2. In Toad Data Modeler select **Settings | Dictionaries**. New **Dictionaries** tab opens.
3. On tab **Dictionaries** click the **Dictionary Items** sub-tab.

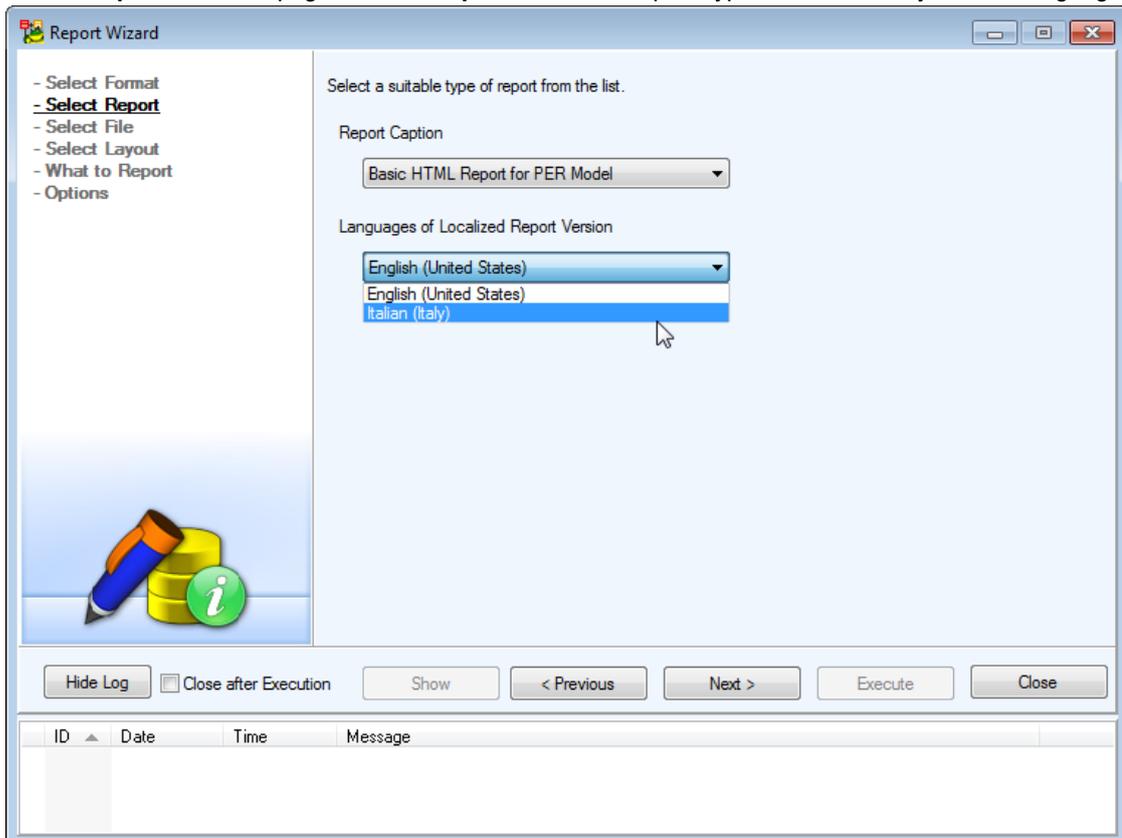


4. Click **Import Dictionary** .

5. Select the downloaded .csv file and click **Import**. All the imported words will be displayed.



6. Make changes if necessary. Otherwise close the **Dictionaries** tab and generate .
7. In the **Report Wizard**, page **Select Report**, select a report type and the newly added language.



8. Go through other steps in the **Report Wizard**.

# File Extensions

Toad Data Modeler works with the following files:

File	Description
*.TXP	Physical data models in XML format
*.TXL	Logical data models in XML format
*.TXM	Metamodels in XML format
*.TXG	User packages in XML format
*.TBG	System packages in binary format
*.TXD	Default Config files
*.TXC	Config files
*.TXE	File with saved environment configuration
*.TXS	Style definition files for HTML reports
*.XSLT	File for XSL transformation
*.XSD	File describing structure of XML file
*.TXA	File with Aliases
*.TXV	File with versions and revisions of the Version Manager
*.TXN	Exceptions
*.TXI	Export/Import of Dictionary
*.TXO	OTPs settings
*.CSV	File for import/export of glossaries to naming conventions
*.TXN	File with defined naming convention rules



Note: Meaning of the extensions:

“T” - Toad Data Modeler

“X” - XML format

'last letter' is intuitive, e.g. “P” for Physical model, “V” for Version Manager etc.

## \*.TXP Files

Physical models created in Toad Data Modeler have \*.TXP extension. These files are in XML format.

## \*.TXL Files

Logical models created in Toad Data Modeler have \*.TXL extension. These files are in XML format.

By default, the .TXP and .TXL files are saved to a path defined in the **Settings** menu | **Options** | **Application** | **Paths** | **Advanced** tab | **Paths to Models**.

## \*.TXM Files

Metamodels in XML format.

System metamodels are saved together with the application installation package. Possible path is: C:\Program Files\Quest Software\Toad Data Modeler 3\Packages\System\MetaModels.

Path to user's metamodels can be set in the **Settings** menu | **Options** | **Application** | **Paths** | **Paths to Metamodels**.

## \*.TXG and \* TBG Files

Packages where definition of database or its part, scripts, forms, data types etc. are saved. System packages are in binary format, user packages in XML format.

**System packages** are saved together with the application installation package. Possible path is:

C:\Program Files\Quest Software\Toad Data Modeler 3\Packages\System.

**User packages** are saved in user's Documents and Settings directory, e.g.:

C:\Documents and Settings\user name\My Documents\Toad Data Modeler\Installation Name\Packages\{SOME GUID Number}.

## \*.TXD Files

Default config files that are saved together with the application installation package, e.g.:

C:\Program Files\Quest Software\Toad Data Modeler 3\Configs.

## \*.TXC Files

Config files that contain settings of Toad Data Modeler. The settings are accessible in the **Settings** menu | **Options**.

.TXC files are modified .TXD files. If .TXC file doesn't exist, it will be created in user's Documents and Settings directory automatically after running the application. Settings of .TXD file will be copied to the new .TXC file then.

## \*.TXE File

Changed environment of Toad Data Modeler will be saved to this file after the application is closed.

This file is saved in user's Documents and Settings directory.

## \*.TXS Files

Files with Style definitions for HTML reports.

CSS styles are saved in user's Documents and Settings directory.

## \*.XSLT Files

Templates for XSL transformation. Default path: C:\Program Files\Quest Software\Toad Data Modeler 3\XSL

## \*.XSD Files

XSD file describes structure of XML file - of your physical model created in Toad Data Modeler (TXP file). XSD shows how the TXP file looks like, how it is structured etc.

By default, the XSD file is generated to:

C:\Documents and Settings\user\My Documents\Toad Data Modeler\Reports

## \*.TXA Files

Files where aliases created during reverse engineering are saved. They are saved in user's Documents and Settings directory.

The path to aliases can be set in the **Settings** menu | **Options** | **Application** | **Reverse Engineering** | **Paths to Aliases**.

## \*.TXV Files

Local files created after the **Check Out** operation in Version Manager. These files are contained in projects saved in Version Manager. They are saved in user's Documents and Settings directory.

The paths can be set in the **Settings** menu | **Options** | **Application** | **Version Manager**.

## \*.TXN Files

Files that contain exceptions for data type conversion between databases.

## \*.TXI Files

Files where dictionary items (User Data Types, Dictionary Types and Domains) are saved during export/import between models. You can save the .TXI file where you want, no default path is defined.

## \*.TXO Files

Files where selected OTPs settings are saved.

1. Default (System) Selected OTPs are stored by default at:

C:\Program Files\Quest Software\Toad Data Modeler\Selected OTPs

2. User Selected OTPs are stored by default at:

C:\Documents and Settings\user\My Documents\Toad Data Modeler\Installation Name\Selected OTPs

## \*.CSV Files

Import/export of glossaries (\*.CSV files) also from/to other tools is possible. See the **Naming Convention Properties** dialog | **Glossary** tab | **Import** button. You can find some CSV files with diacritical marks at: C:\Program Files\Quest Software\Toad Data Modeler 3\Naming Conventions\CSV.

## \*.TXN Files

Every defined naming convention is stored in external .TXN file stored by default in user Documents folder, directory Toad Data Modeler\ Installation name\Naming Conventions.

### **i** Note:

1. Generally, it stands that system files are saved together with the application installation package. (E.g.: C:\Program Files\Quest Software\Toad Data Modeler 3).

Files modified by a user are saved in the user's Documents and Settings directory. (E.g.: C:\Documents and Settings\user name\My Documents\Toad Data Modeler, or C:\Documents and Settings\user name\Application Data\Quest Software\Toad Data Modeler.

2. The Documents and Settings directory is empty until you run Toad Data Modeler. Then, all appropriate files will be copied to this directory from the application installation package.

All changes you make for the files (e.g. changes in the application layout, new aliases, modified styles in reports, new default path to generated SQL/DDDL code, modified *My Package.txg* file etc.) will be saved here and will be preserved when you update the application.

If you need to restore the original default settings, you can simply delete appropriate file in the Documents and Settings directory. When you run Toad Data Modeler then, appropriate original default file from C:\Program Files\Quest Software\Toad Data Modeler 3 will be copied to the Documents and Settings directory again.

# Enabled/Disabled Databases

Starting with version 5.4, Toad Data Modeler implements a new system of managing databases.

## **Toad Data Modeler 5.3 and older:**

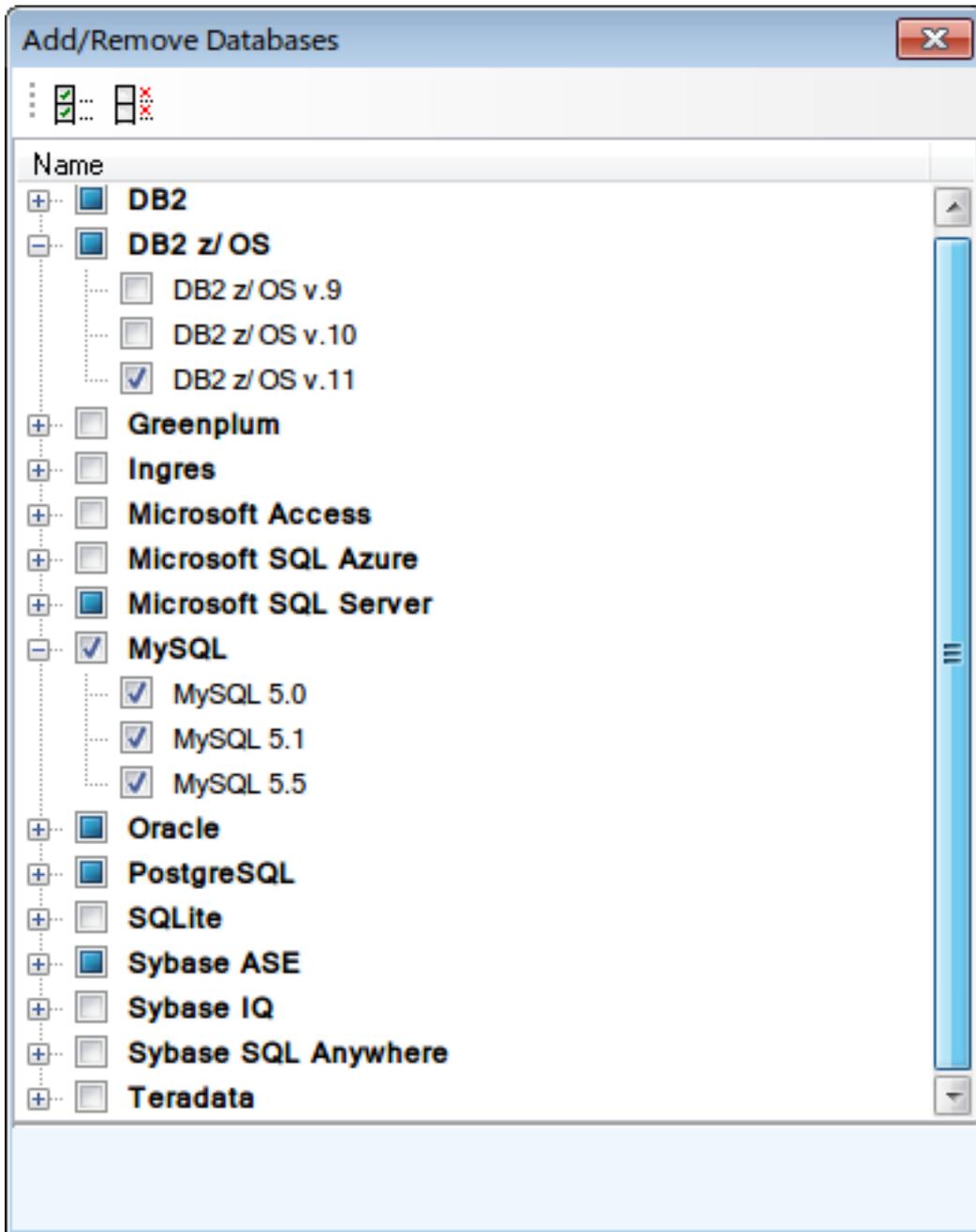
- During application installation you were able to choose which databases should/should not be installed. This could not be changed in the application, you had to launch installation program again. By not installing unused databases, you were able to save space on drive.

## **Toad Data Modeler 5.4 and newer:**

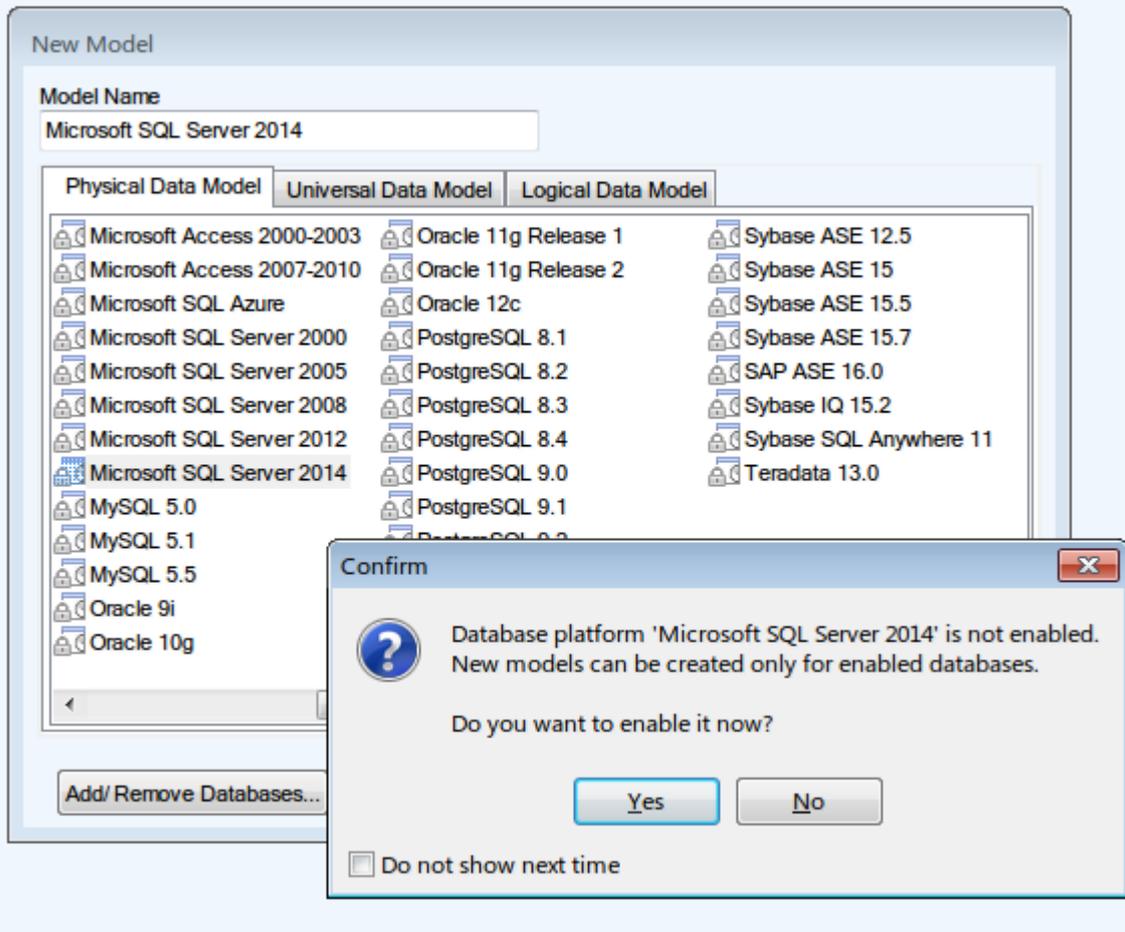
- All databases are now installed with the application. A default set of databases is enabled and you can enable/disable databases at any time. By disabling unused databases you can decrease the application startup time.

## **How to enable/disable databases**

You can configure enabled/disabled databases by going to **Settings Menu | Add/Remove Databases**.



Disabled databases cannot be used in certain application actions (create new model, open existing model, reverse engineering...). However, most of the time you are able to enable a specific database at the time you want to work with it (see screenshot below).



# Supported Databases

Toad Data Modeler provides full support to the databases listed below:

- Amazon® Aurora MySQL 5.6
- Amazon® Aurora PostgreSQL 9.5
- Amazon® Redshift 1.0
- IBM® DB2® z/OS® 11
- IBM® DB2® LUW 9.7, 10.1, 10.5, 11.1, 11.5
- Greenplum Database® 4.2
- Ingres 9.3, 10.0
- EDB Postgres Advanced Server 10
- Microsoft® Access® 2010-2019, incl. Office 365
- Microsoft® Azure® SQL Database V12
- Microsoft® SQL Server® 2012, 2014, 2016, 2017, 2019, 2022
- MySQL 5.6, 5.7, 8.0
- Oracle® 11g R1, 11g R2, 12c R1, 12c R2, 18c, 19c, 21c
- PostgreSQL 9.5, 10, 11, 12
- SQLite 3.7, 3.32
- SAP® SQL Anywhere 17
- SAP® ASE 16.0
- Sybase® ASE 15.7
- Sybase® IQ 15.2
- Teradata 13, 16.2
- Vertica Database 8.0
- Other (Universal Model)

**i** **IMPORTANT:** You can also create and work with models from other versions of databases that have been deprecated by their production companies. Toad Data Modeler cannot provide fixes and provide support for more features of these databases. Uncheck **Show Supported Databases Only** to display all databases that you can create and open models for.

# Details of Database Support

Supported Database System	Reverse Engineering		Change Script Generation	SQL/DDDL Code Generation
	From a Database	From a SQL File		
Amazon Aurora MySQL	●	●	●	●
Amazon Aurora PostgreSQL	●	●	●	●
Amazon Redshift	●	●	●	●
IBM DB2 z/OS	●	●	●	●
IBM DB2 LUW	●	●	●	●
Greenplum	●	●	●	●
Ingres	●	●	●	●
EDB Postgres Advanced Server	●	●	●	●
Microsoft Access	●	●	●	●
Microsoft Azure SQL Database	●	●	●	●
Microsoft SQL Server	●	●	●	●
MySQL	●	●	●	●
Oracle	●	●	●	●
PostgreSQL 9.5 and newer	●	●	●	●
SQLite	●	●	●	●
SAP ASE	●	●	●	●
Sybase ASE	●	●	●	●
Sybase IQ	●	●	●	●
SAP SQL Anywhere	●	●	●	●
Teradata	●	●	●	●
Vertica Database	●	●	●	●
Other databases (Universal Model)	●	●	●	●

**i** Note: Toad Data Modeler includes also support of Universal DB/ANSI Models. See [Universal DB/ANSI Model](#) for more information.

# Specifics - Amazon Redshift 1.0

Toad Data Modeler offers support for Amazon Redshift 1.0, including support for Reverse Engineering and DDL Script Generation. There are however the following limitations to Redshift support:

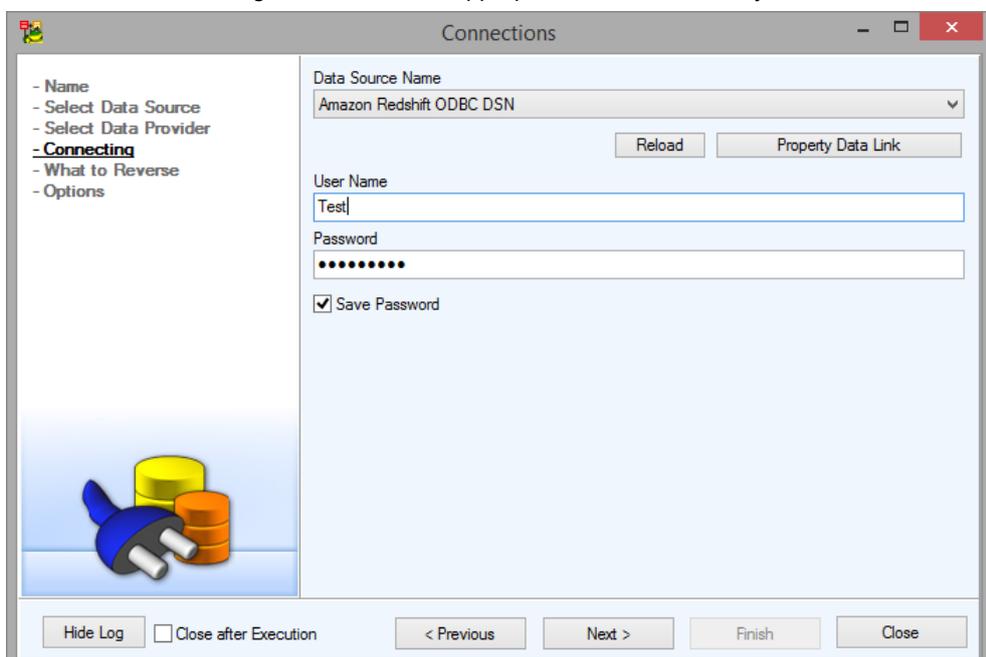
- Databases and Libraries (and grants for them) are not supported
- CREATE TABLE: IF NOT EXISTS clause is not supported

## Reverse Engineering - Amazon Redshift 1.0

Available **Data Providers** are:

- **Connection via ODBC**

In order to connect using ODBC, install an appropriate ODBC driver for your database first.



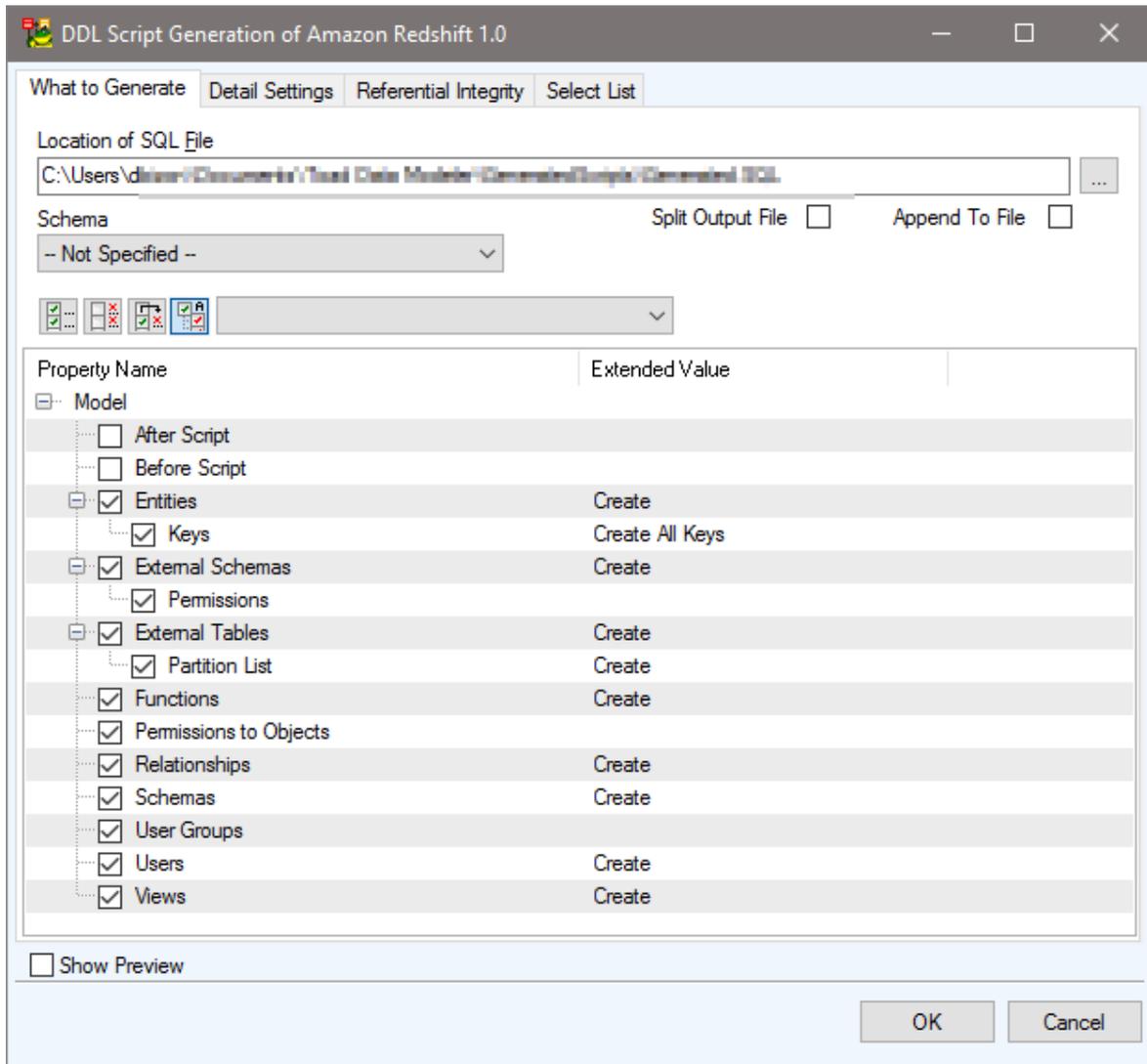
### **To create Amazon Redshift connections**

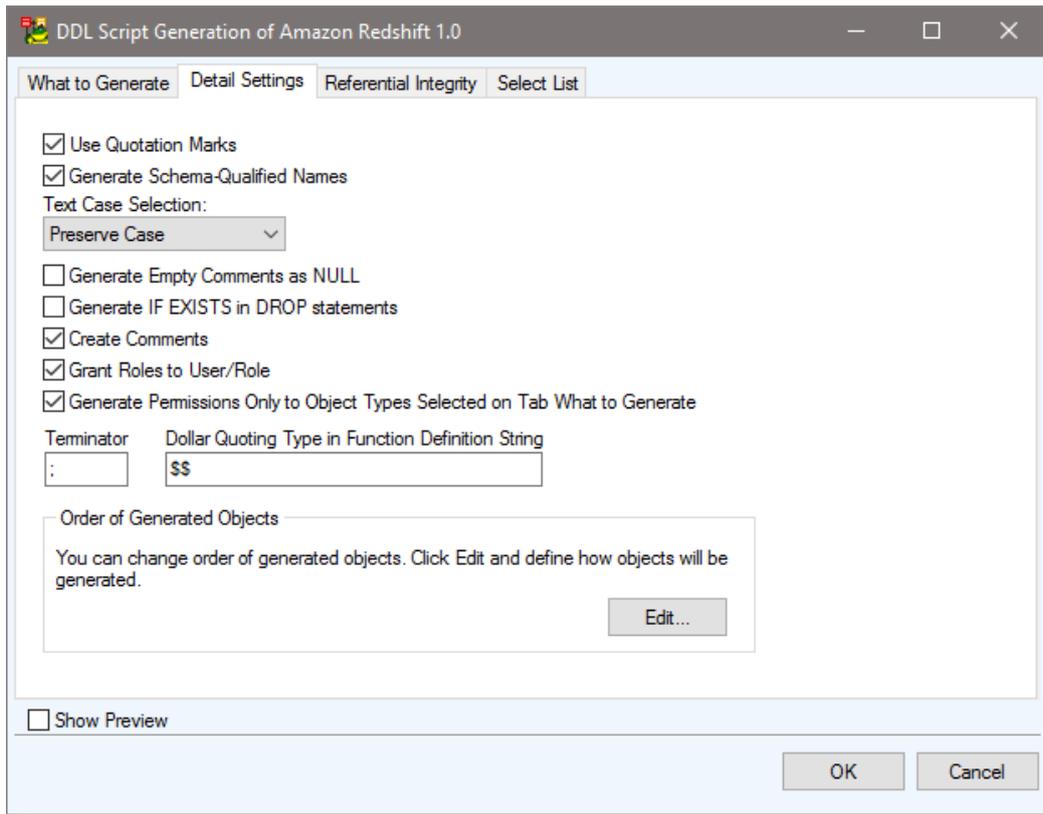
1. Select Amazon Redshift ODBC driver in **Data Source Name**
2. Click **Property Data Link** to configure connection properties
3. Enter the address to your Redshift server into **Server**
4. Enter your port number (default: 5439) and database name
5. Enter your user name and password
6. Change any other necessary settings and **Test** or click **OK**

**NOTE:** In case of access violation during **Reverse Engineering** of Amazon Redshift when using ODBC driver perform the following steps:

1. Select your ODBC driver in **ODBC Data Source Administration**
2. Click **Configure**
3. Click **Additional Options**
4. In **Additional Configuration**, check **Use Multiple Statements**

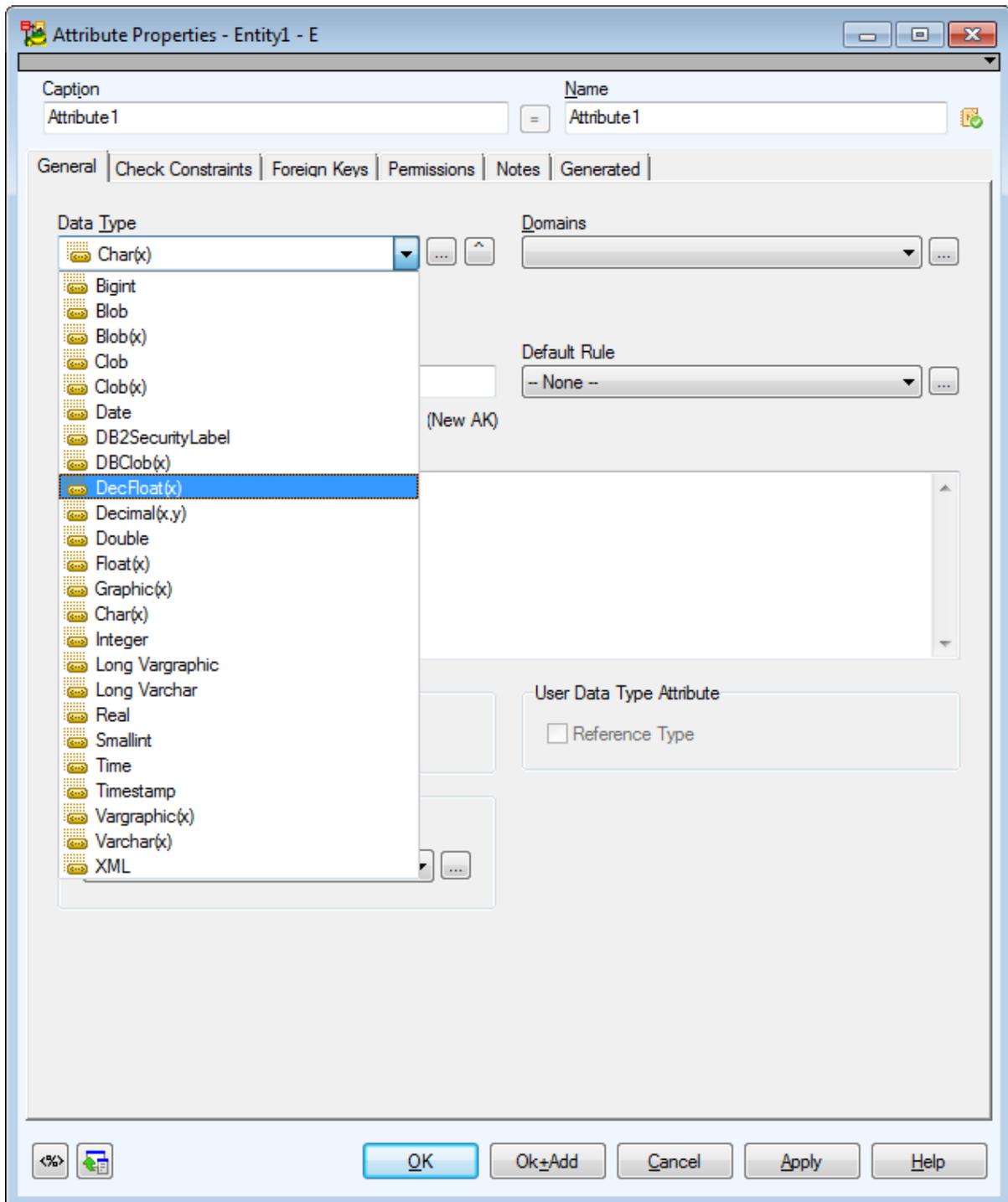
## Script Generation - Amazon Redshift 1.0



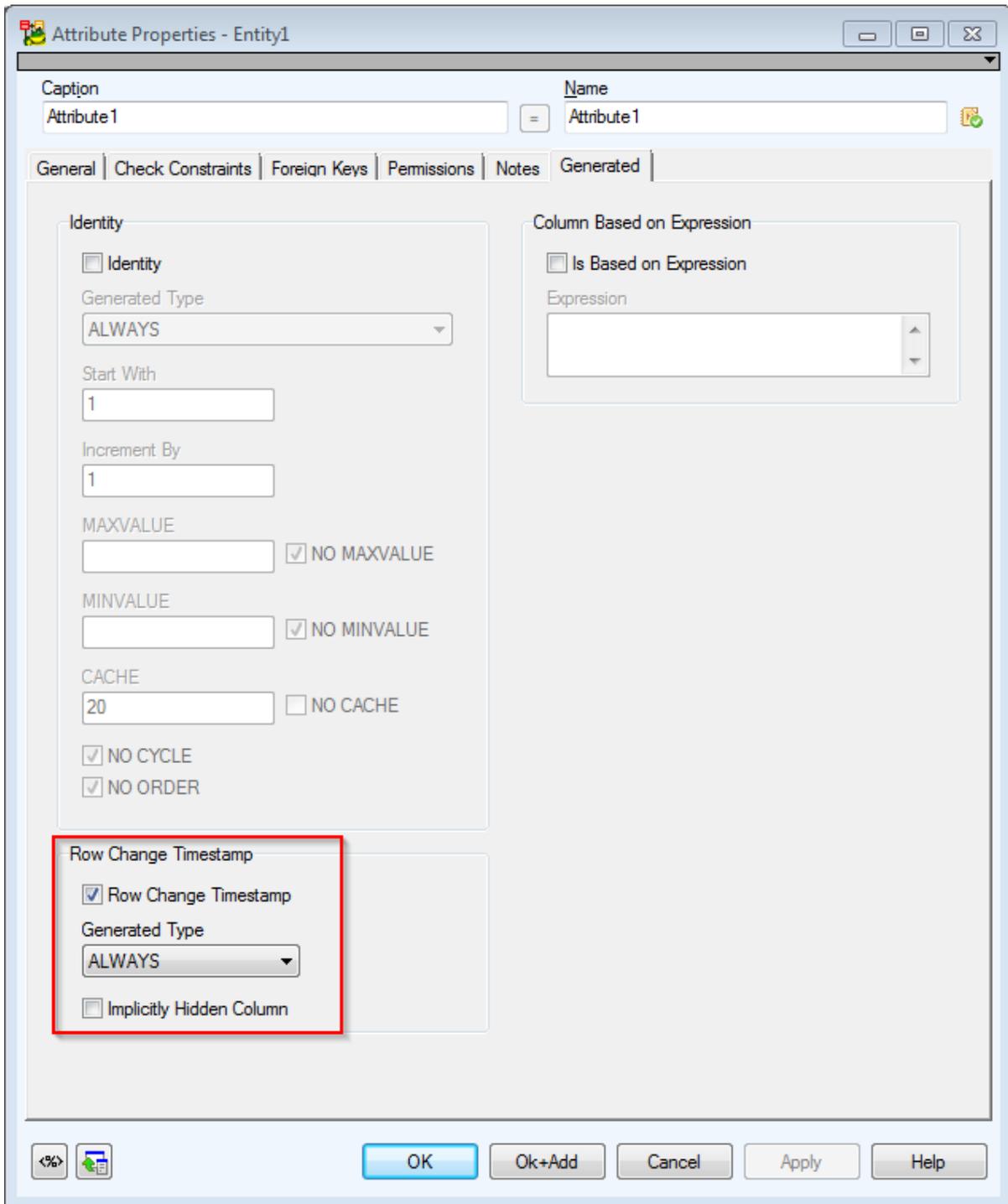


# Specifics - DB2 9.5 (LUW)

# Attribute

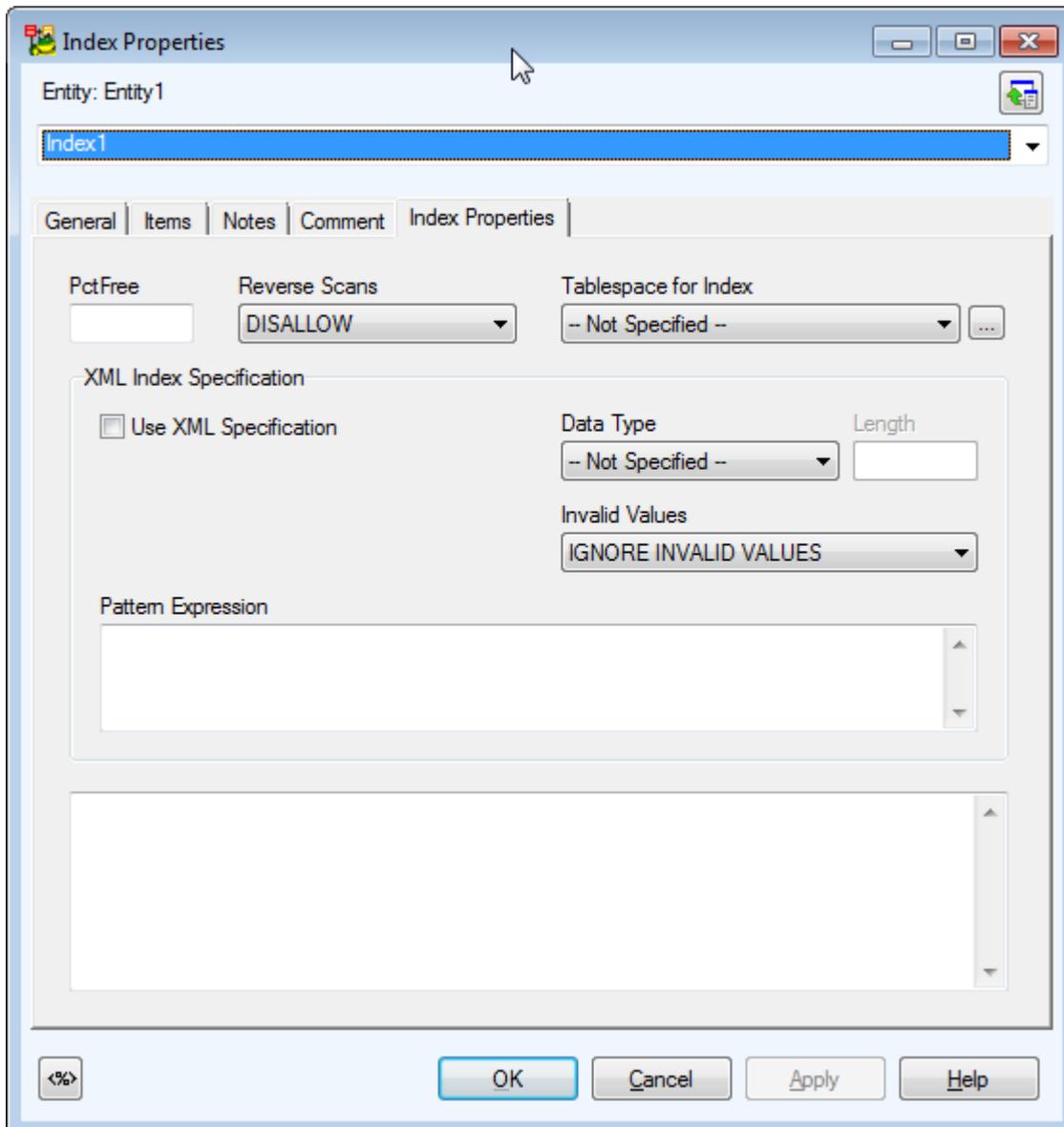


Data type *DecFloat(x)*.

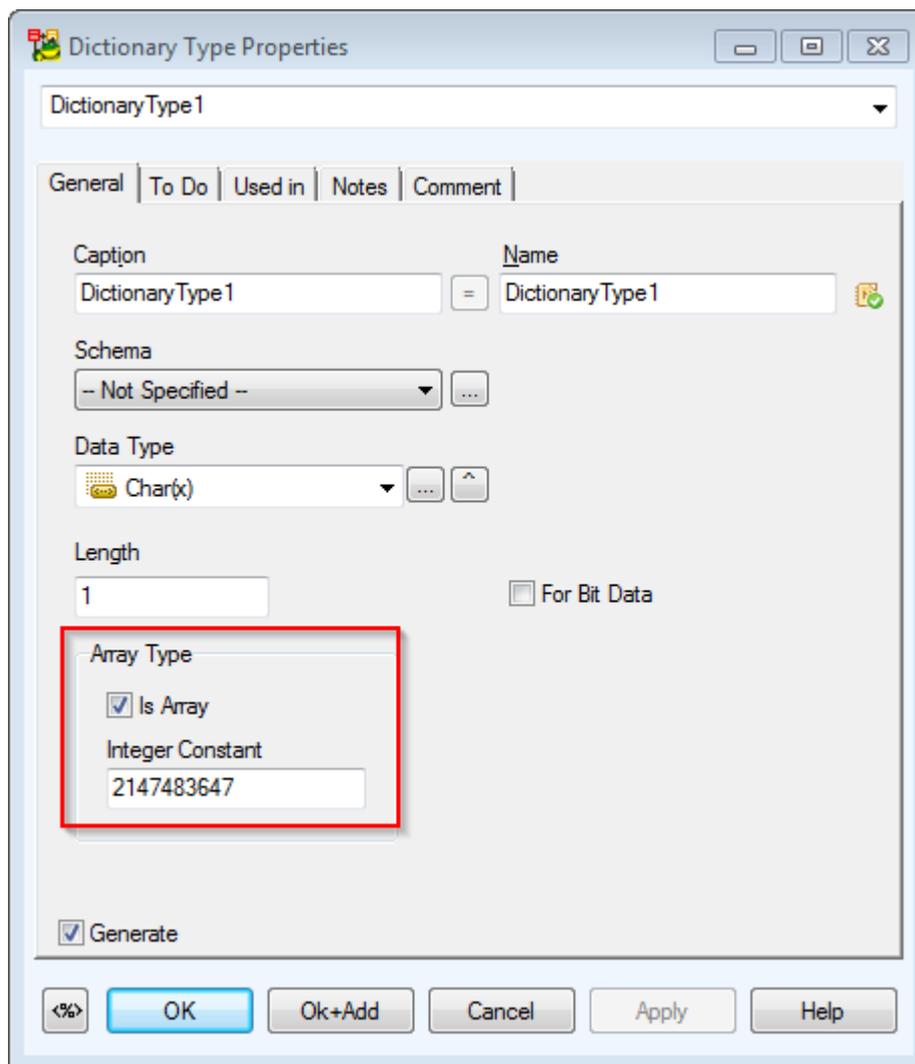


For data type *Timestamp*, the **Row Change Timestamp** area is available on tab **Generated**.  
To enable the **Implicitly Hidden Column** checkbox, select the **Row Change Timestamp** checkbox.

# Index



# Dictionary Type



New Array Type.

## Extra Objects - DB2 9.5 (LUW)

See other objects in Model Explorer:

- Security Labels
- Security Policies
- Sequences
- Tablespaces

# Reverse Engineering - IBM DB2 LUW

Available **Data Providers** are:

- **Native Connection**
- **Connection via ADO**
- **Connection via ODBC**

**Native Connection:**

Connections

- Name  
- Select Data Source  
- Select Data Provider  
- **Connecting**  
- What to Reverse  
- Options

Host

Host Name: 10.11.40.26      Port: 50000

User

User Name: db2admin      Password: ●●●●

Save Password

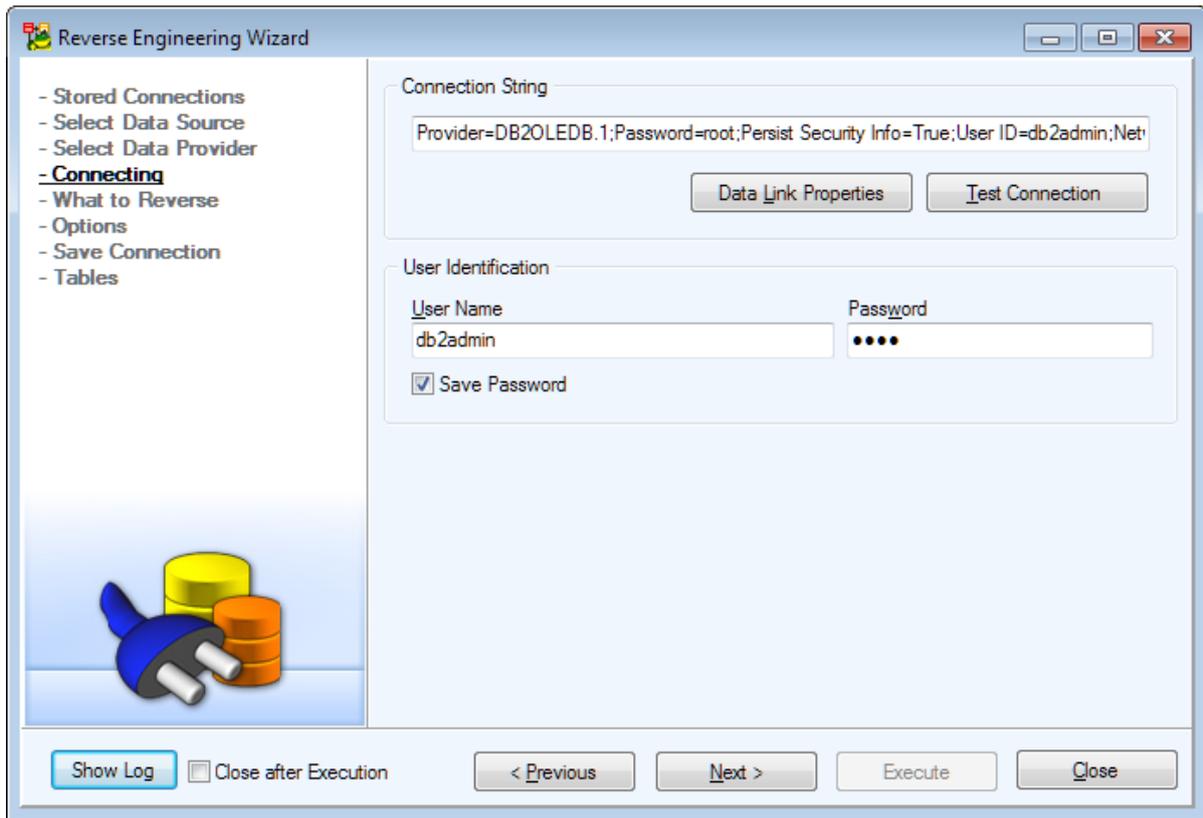
Database

Database Name: SAMPLE

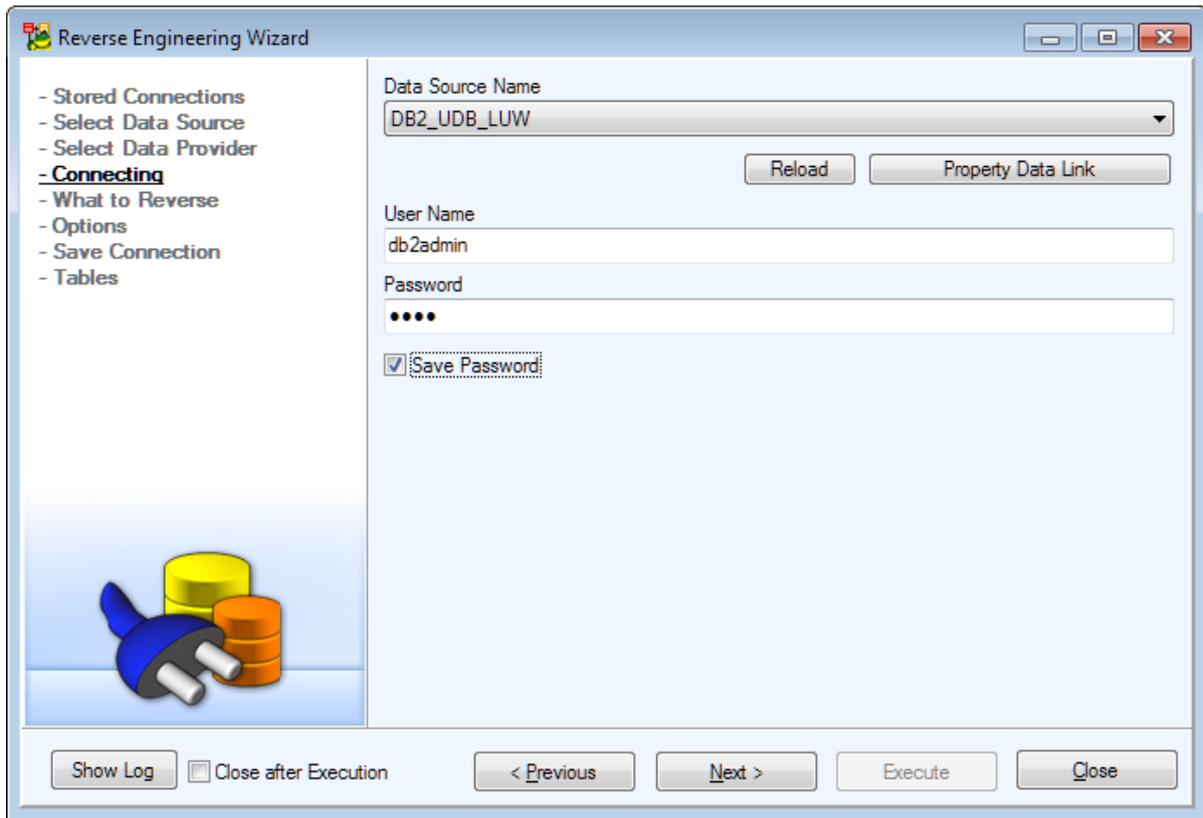
Connection String  
HostName=10.11.40.26; Port=50000; Database=SAMPLE; UID=db2admin;

Show Log     Close after Execution    < Previous    Next >    Finish    Close

**Connection via ADO:**

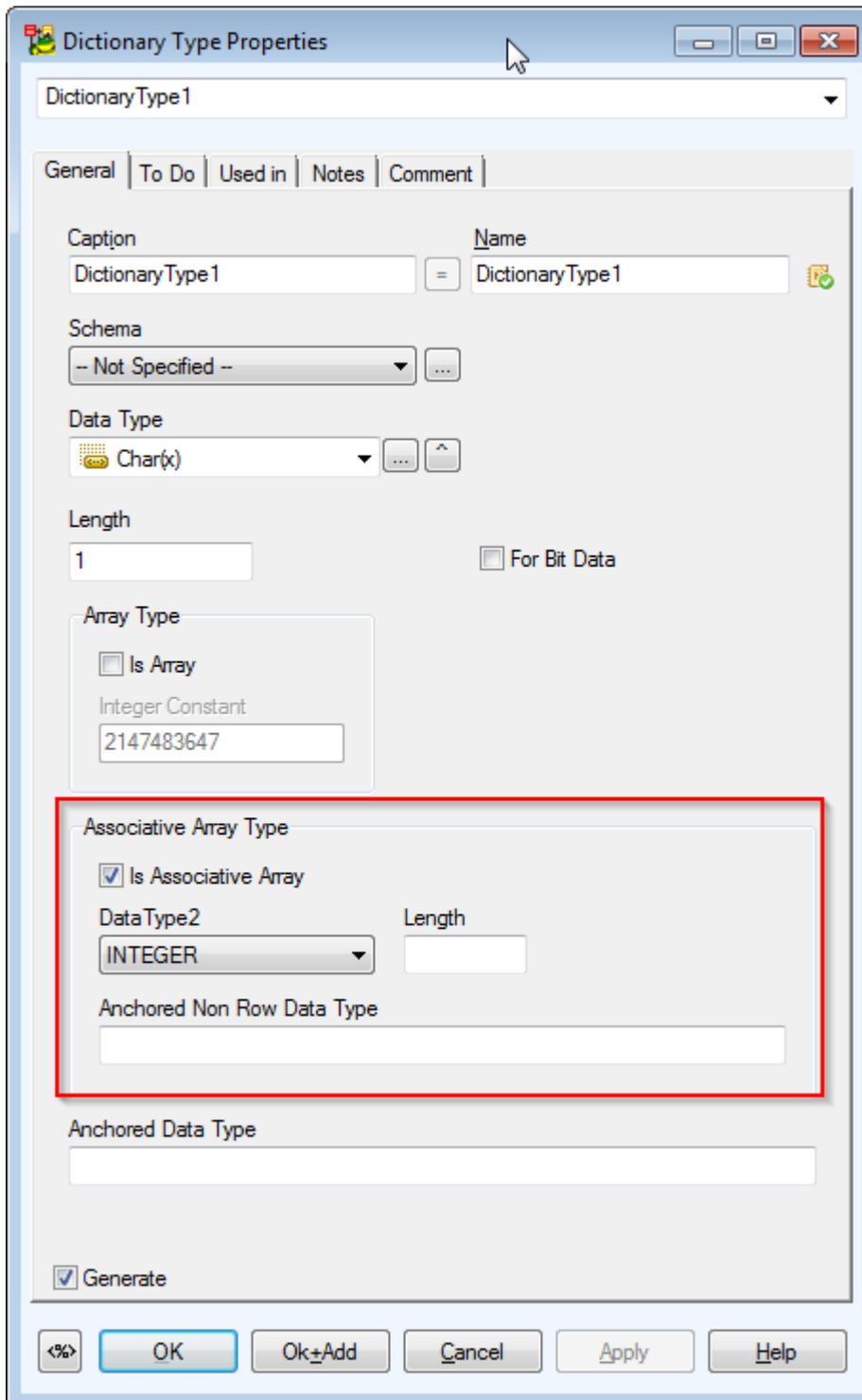


Connection via ODBC:

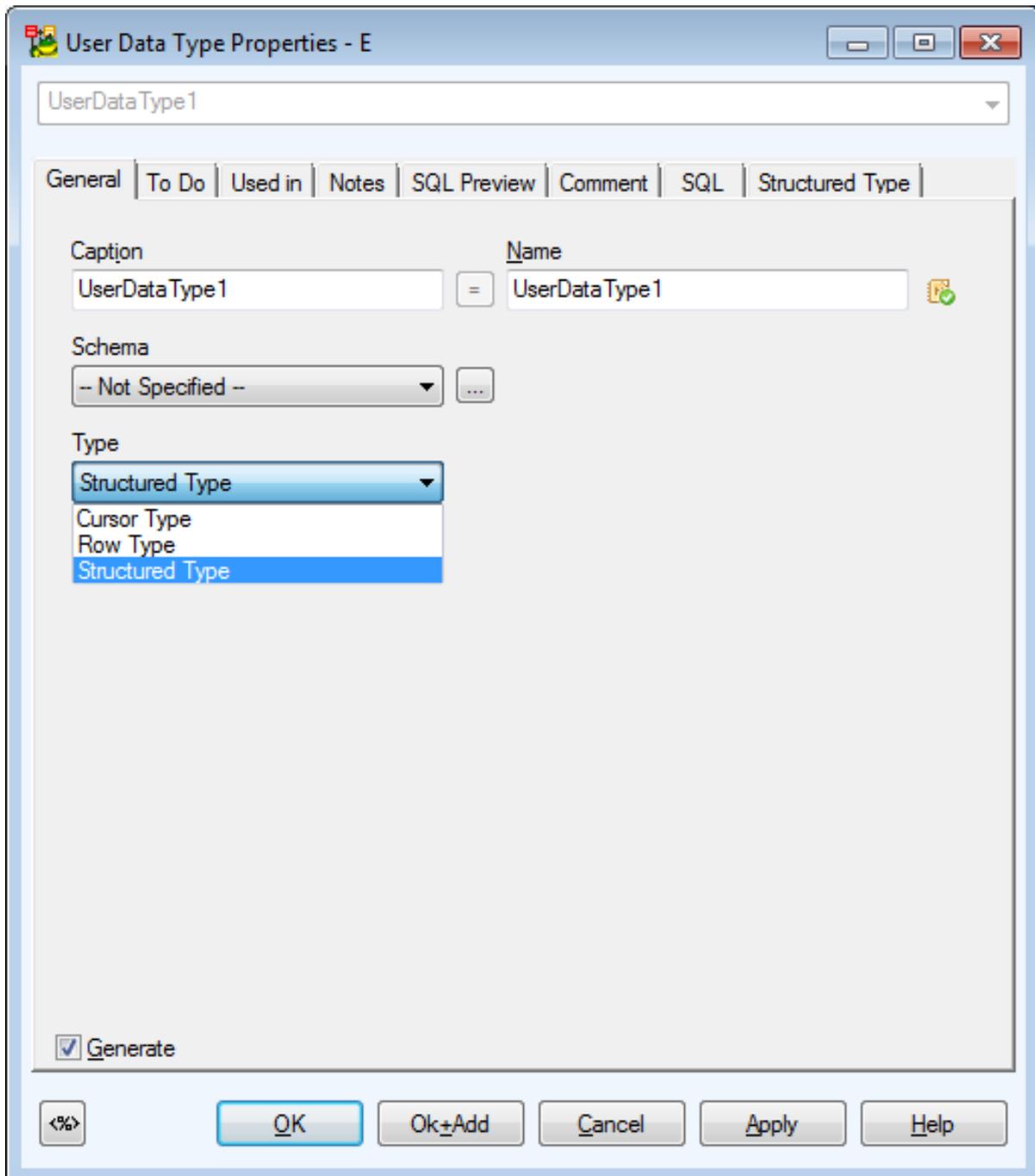


# Specifics - DB2 9.7 (LUW)

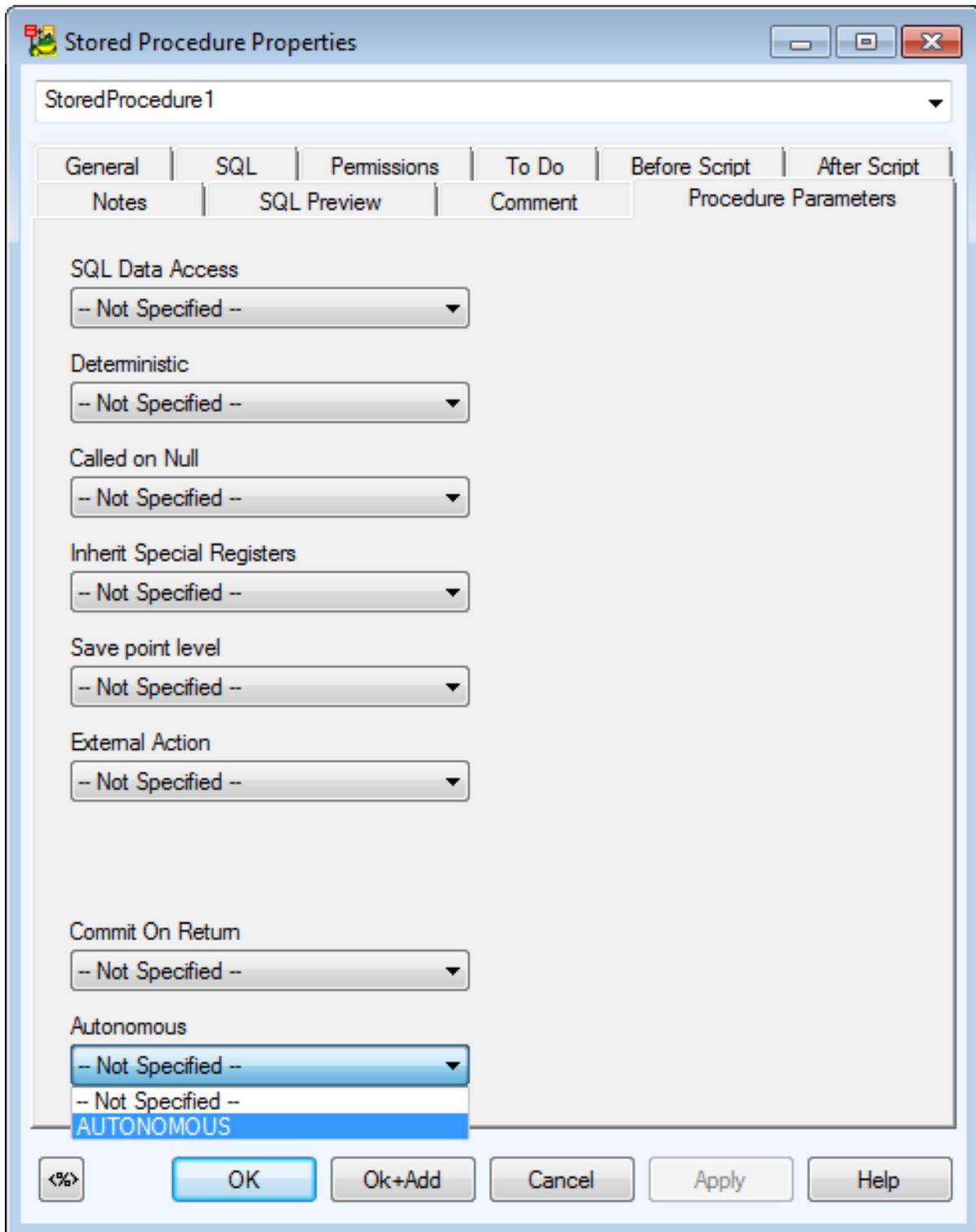
# Dictionary Type



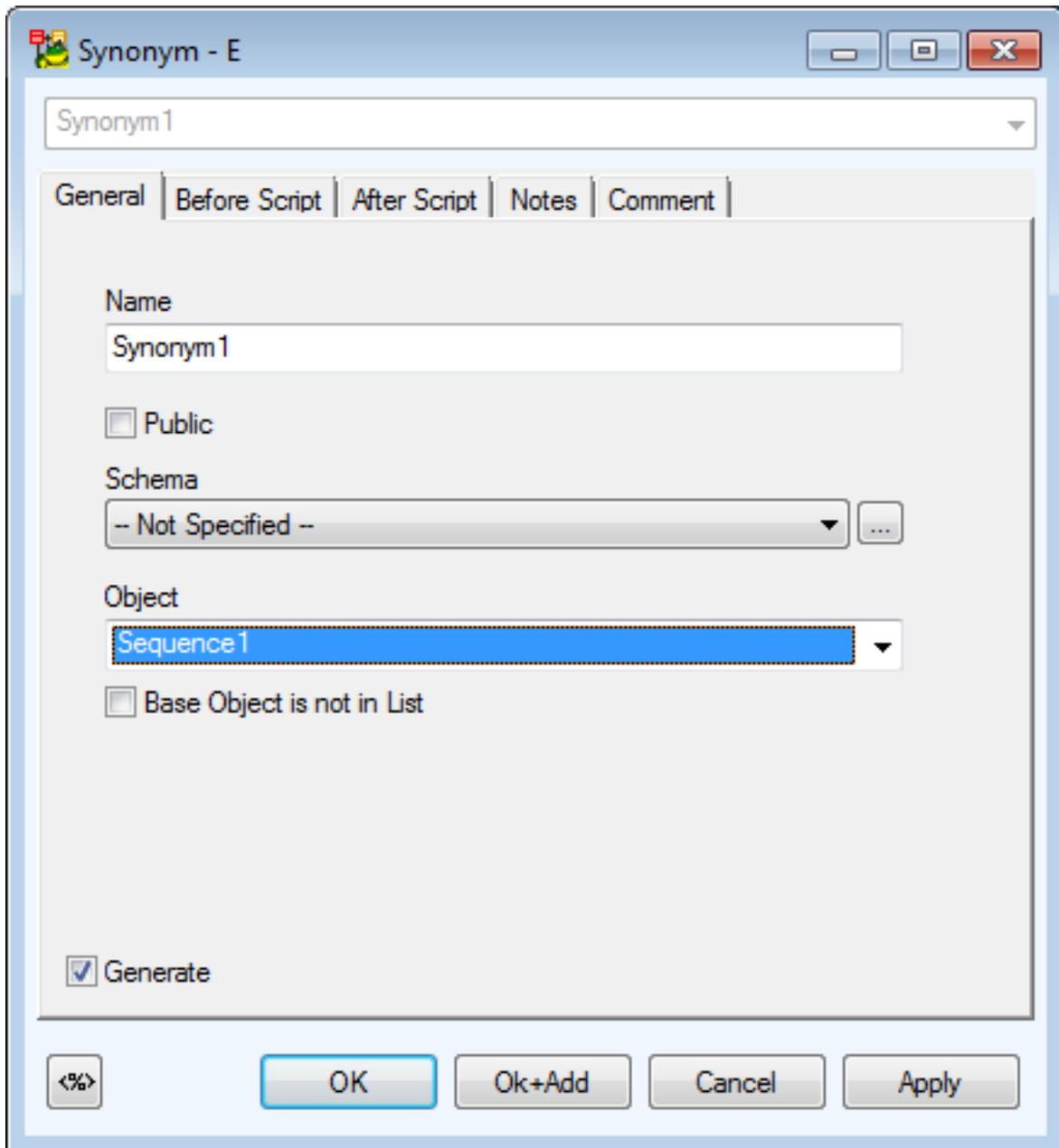
# User Data Type



# Stored Procedure



# Synonym



Aliases (Synonyms) are supported, including Public Synonym:

Public Synonym specifies that the alias is an object in the system schema SYSPUBLIC.

To select a sequence in the list of objects, open the **Synonym Properties** dialog | **General** tab | **Object** box.

If the sequence is not defined in the model, select the **Base Object is not in List** checkbox and write the sequence to box **Base Object Name** (start with text SEQUENCE).

Other objects in Model Explorer:

- Security Labels
- Security Policies
- Sequences
- Tablespaces

## Reverse Engineering - IBM DB2 LUW

Available **Data Providers** are:

- **Native Connection**
- **Connection via ADO**
- **Connection via ODBC**

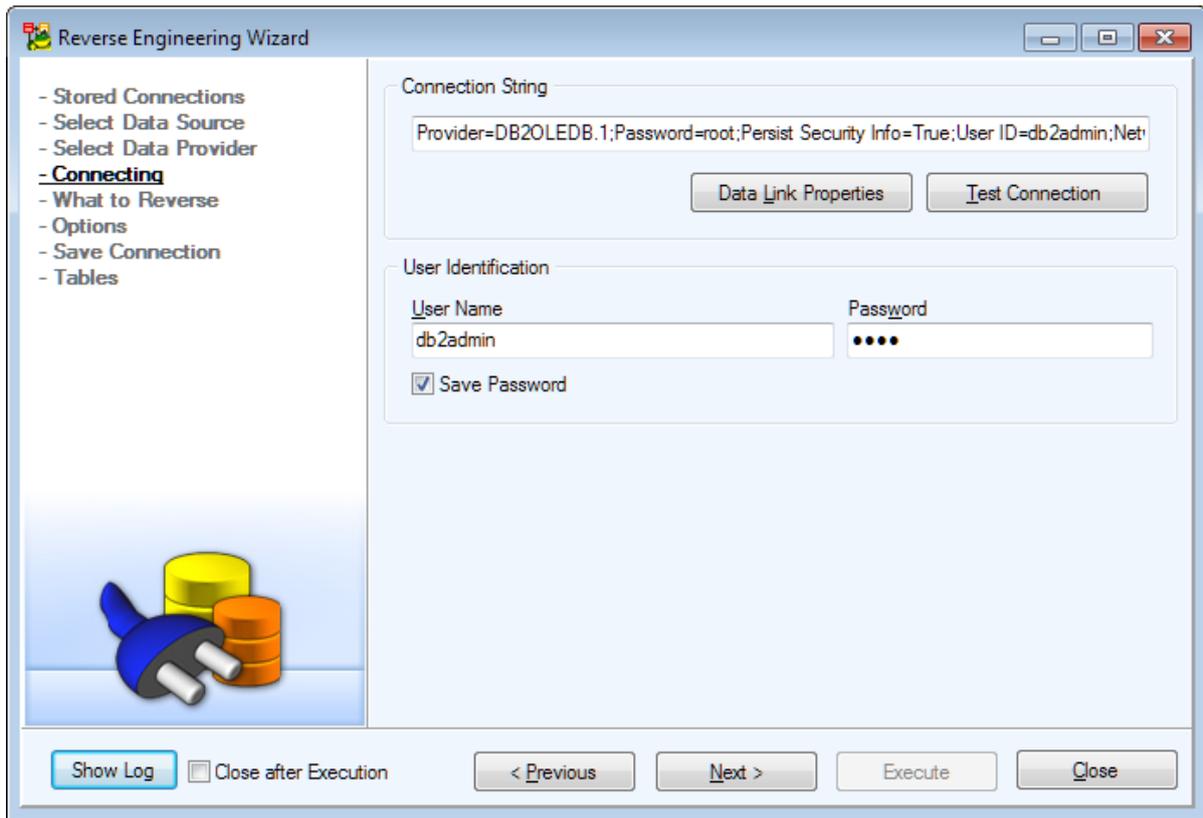
**Native Connection:**

The screenshot shows the 'Connections' dialog box in Toad Data Modeler. The 'Connecting' step is selected in the left-hand navigation pane. The main area contains the following fields and options:

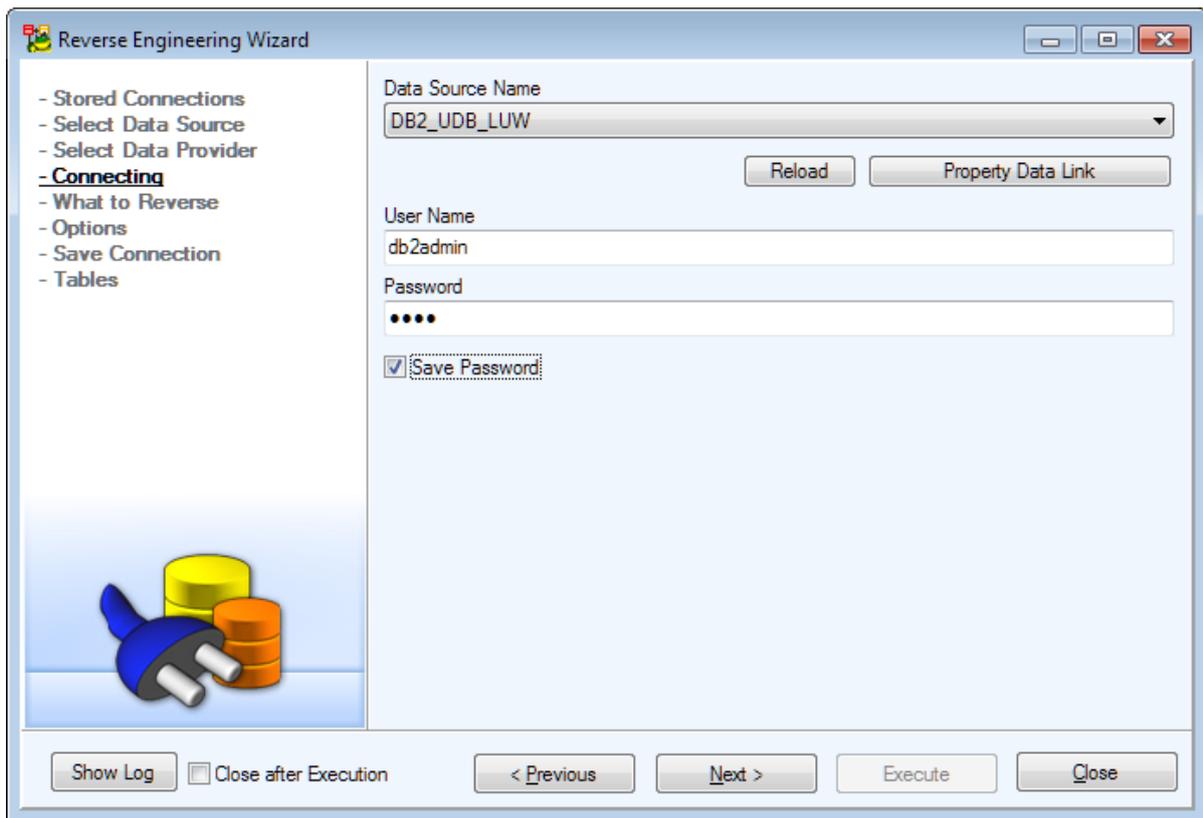
- Host:** Host Name: 10.11.40.26, Port: 50000
- User:** User Name: db2admin, Password: [masked],  Save Password
- Database:** Database Name: SAMPLE
- Connection String:** HostName=10.11.40.26; Port=50000; Database=SAMPLE; UID=db2admin;

At the bottom of the dialog, there are buttons for 'Show Log', 'Close after Execution' (checkbox), '< Previous', 'Next >', 'Finish', and 'Close'.

**Connection via ADO:**

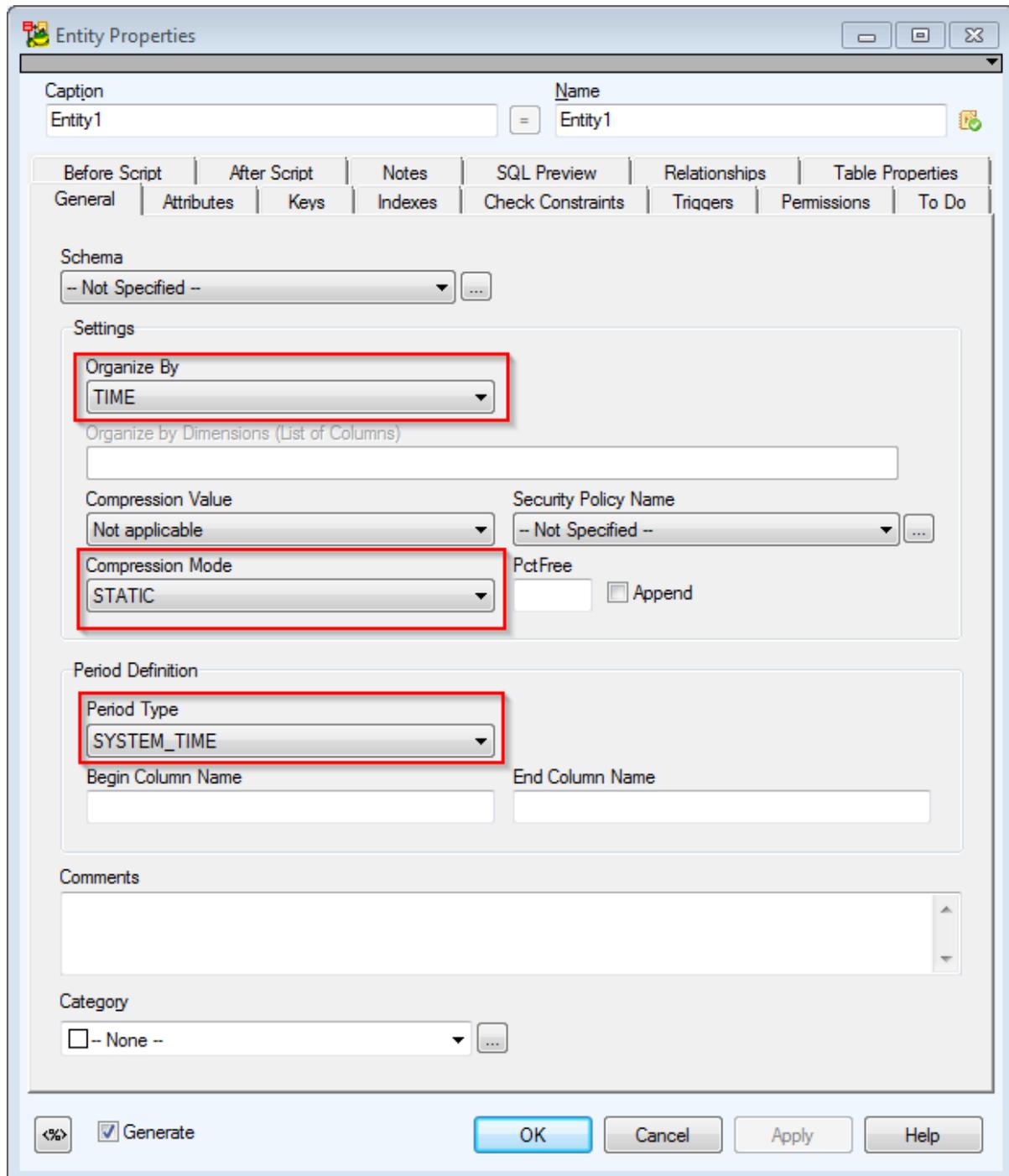


Connection via ODBC:



# Specifics - DB2 10.1 (LUW)

# Entity

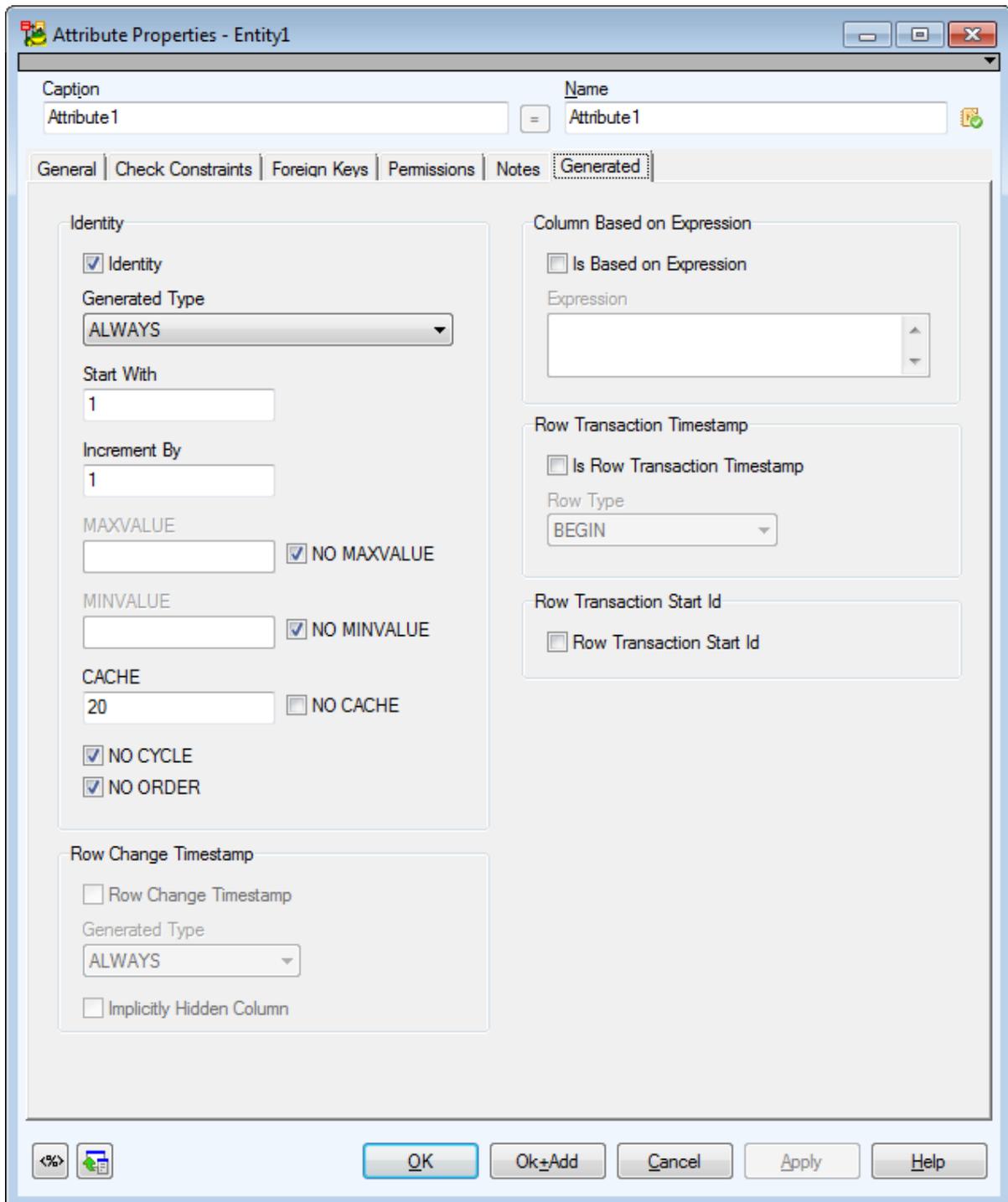


**Organize by Insert Time** option added.

**Compression Mode** option added with values ADAPTIVE/STATIC.

**Period Definition** (BUSINESS\_TIME/SYSTEM\_TIME values) option added.

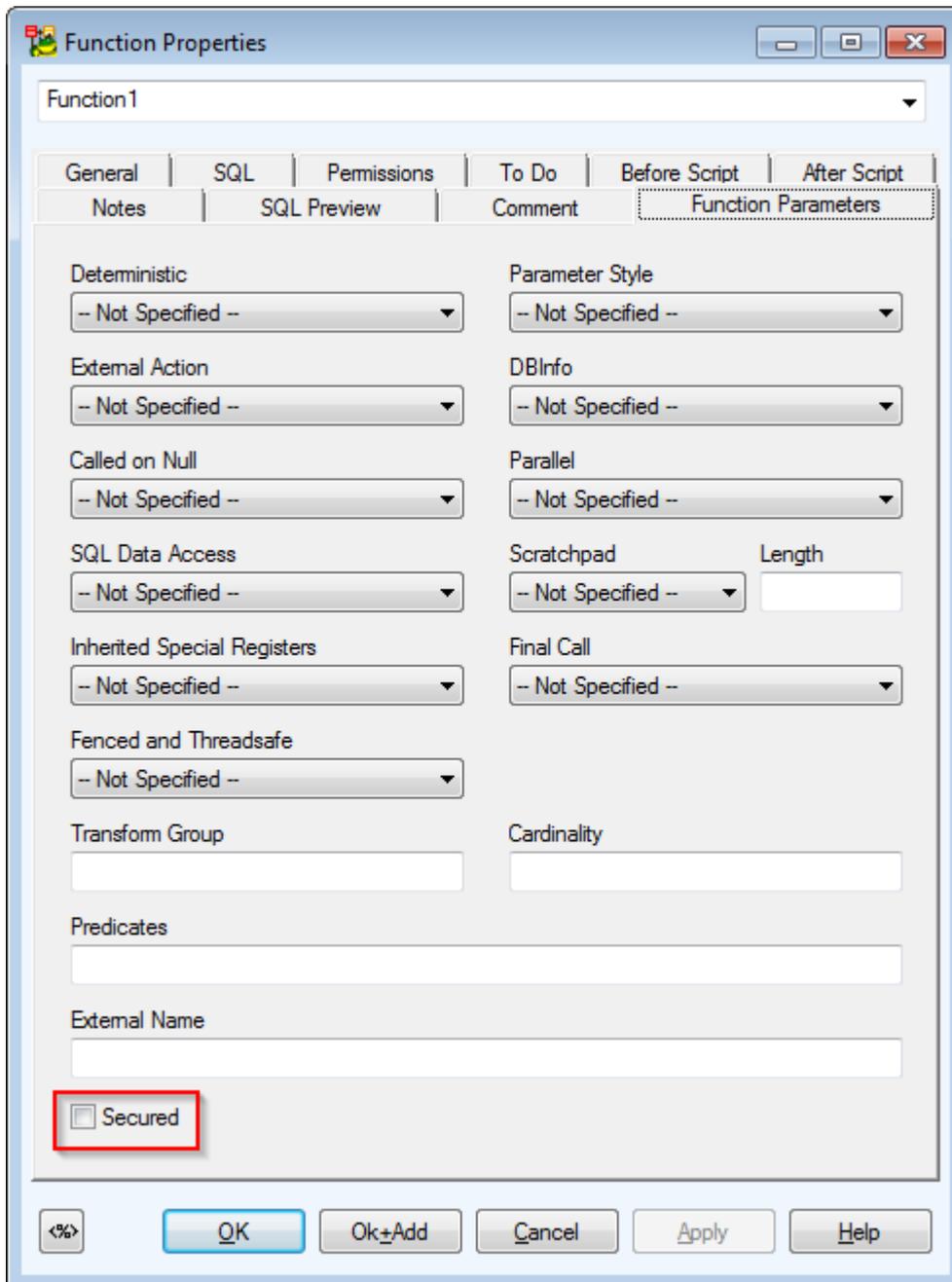
# Attribute



New data types - *nchar*, *nvarchar*, *nclob*.

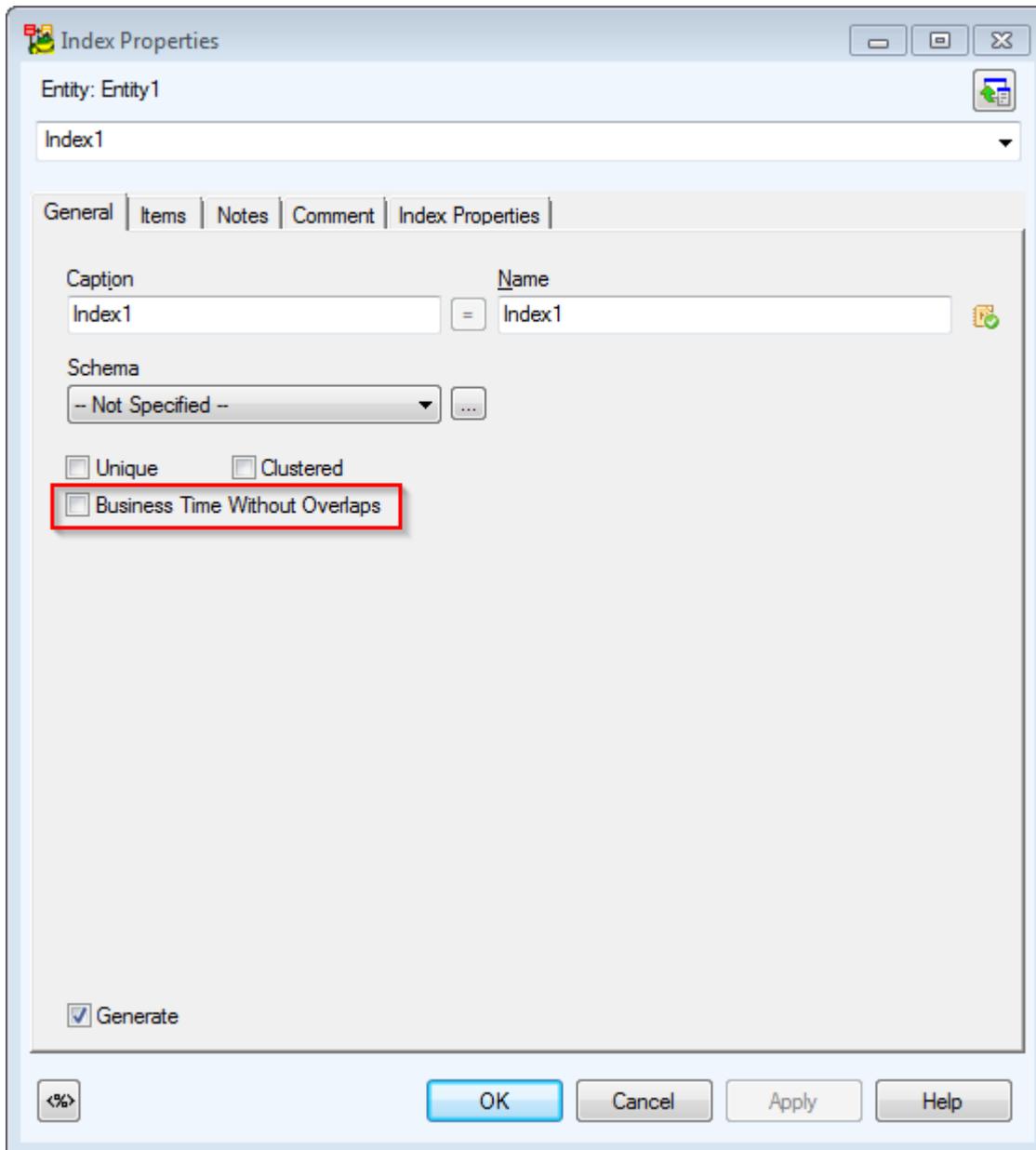
New tab **Generated** in the **Attribute Properties** dialog.

# Function



For function types *External Scalar*, *External Table*, *OLE DB*, *SQL* defined on tab **General**, the **Secured** checkbox is available on tab **Function Parameters**.

# Index



**Business Time Without Overlaps** option added on tab **General**.

# Trigger

Trigger event allows OR option, set for generating SQL script in Extended Value, option Create or Replace.

See other objects in Model Explorer:

- Security Labels
- Security Policies
- Sequences
- Tablespaces

## Reverse Engineering - IBM DB2 LUW

Available **Data Providers** are:

- **Native Connection**
- **Connection via ADO**
- **Connection via ODBC**

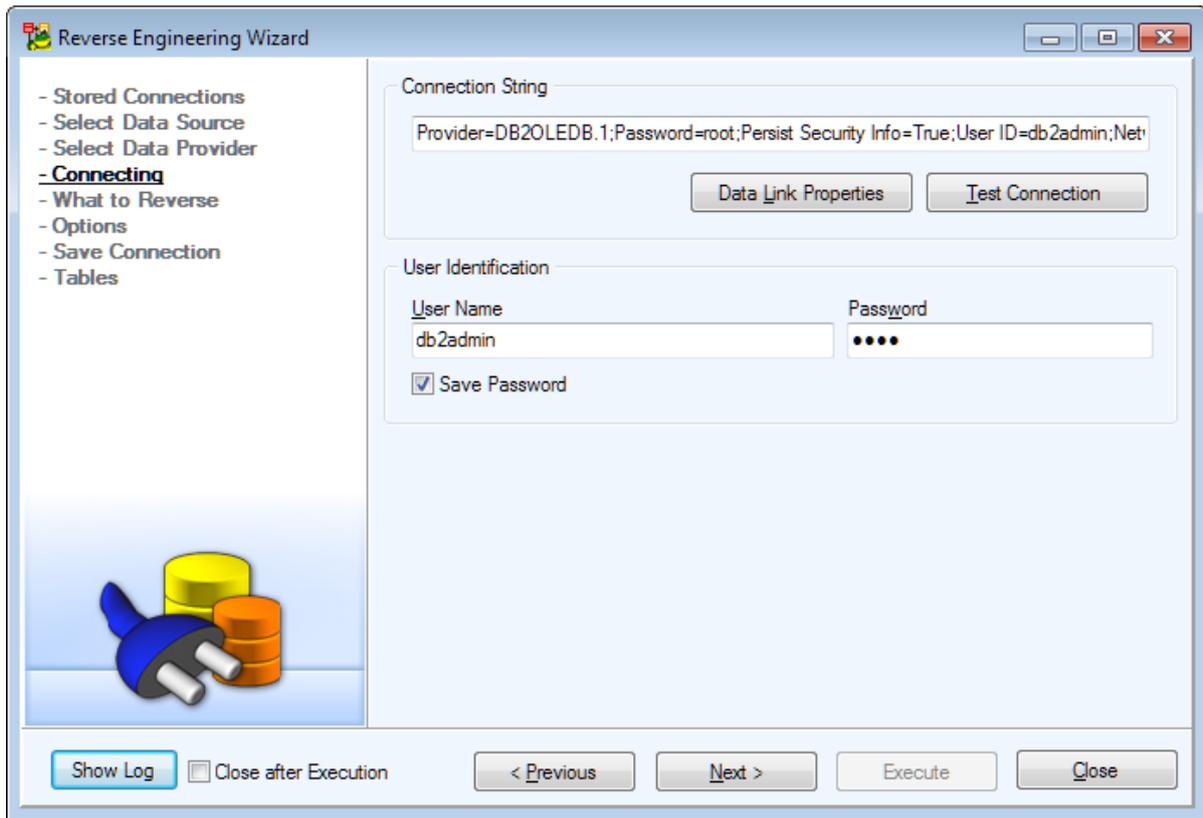
**Native Connection:**

The screenshot shows the 'Connections' dialog box in Toad Data Modeler. The dialog is titled 'Connections' and has a progress bar on the left with the following steps: - Name, - Select Data Source, - Select Data Provider, - **Connecting** (highlighted), - What to Reverse, and - Options. The main area contains the following fields and controls:

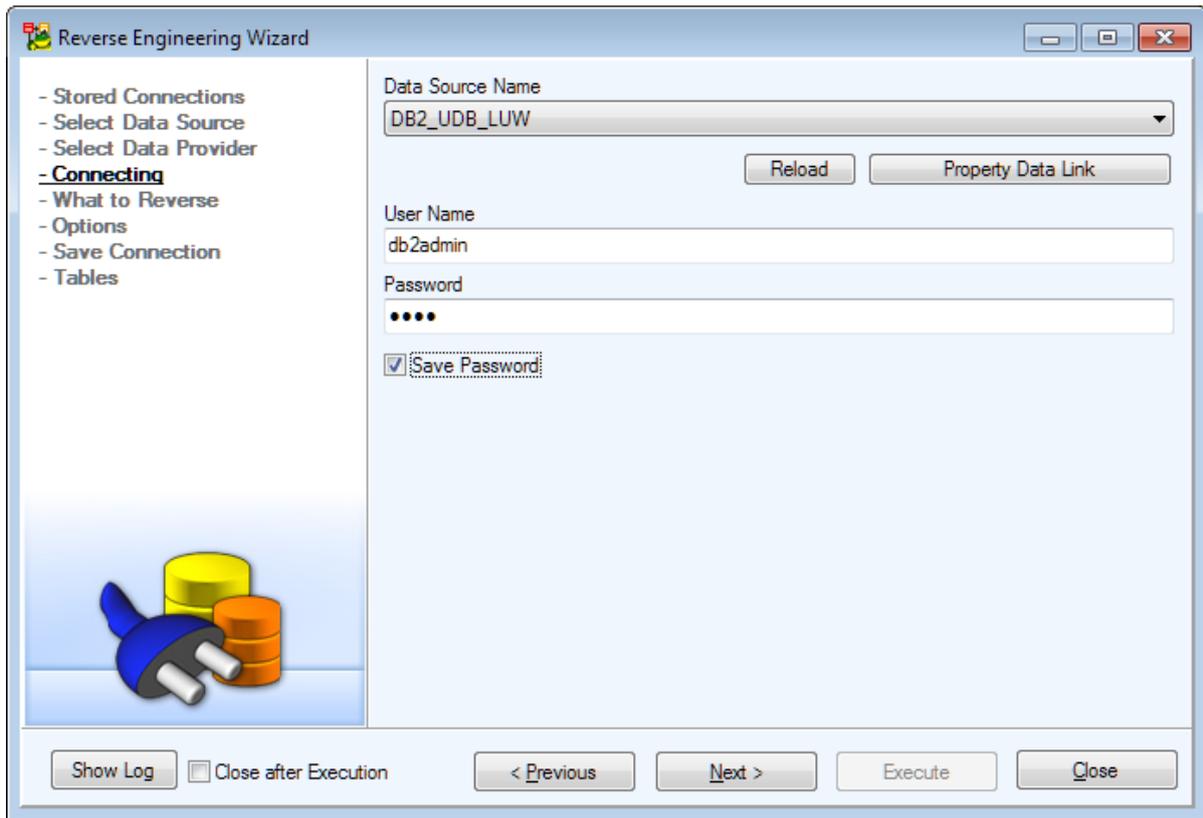
- Host:** Host Name: 10.11.40.26, Port: 50000
- User:** User Name: db2admin, Password: masked (masked with dots), Save Password checkbox checked.
- Database:** Database Name: SAMPLE
- Connection String:** HostName=10.11.40.26; Port=50000; Database=SAMPLE; UID=db2admin;

At the bottom of the dialog, there are several buttons and checkboxes: Show Log, Close after Execution (checkbox), < Previous, Next > (highlighted), Finish, and Close.

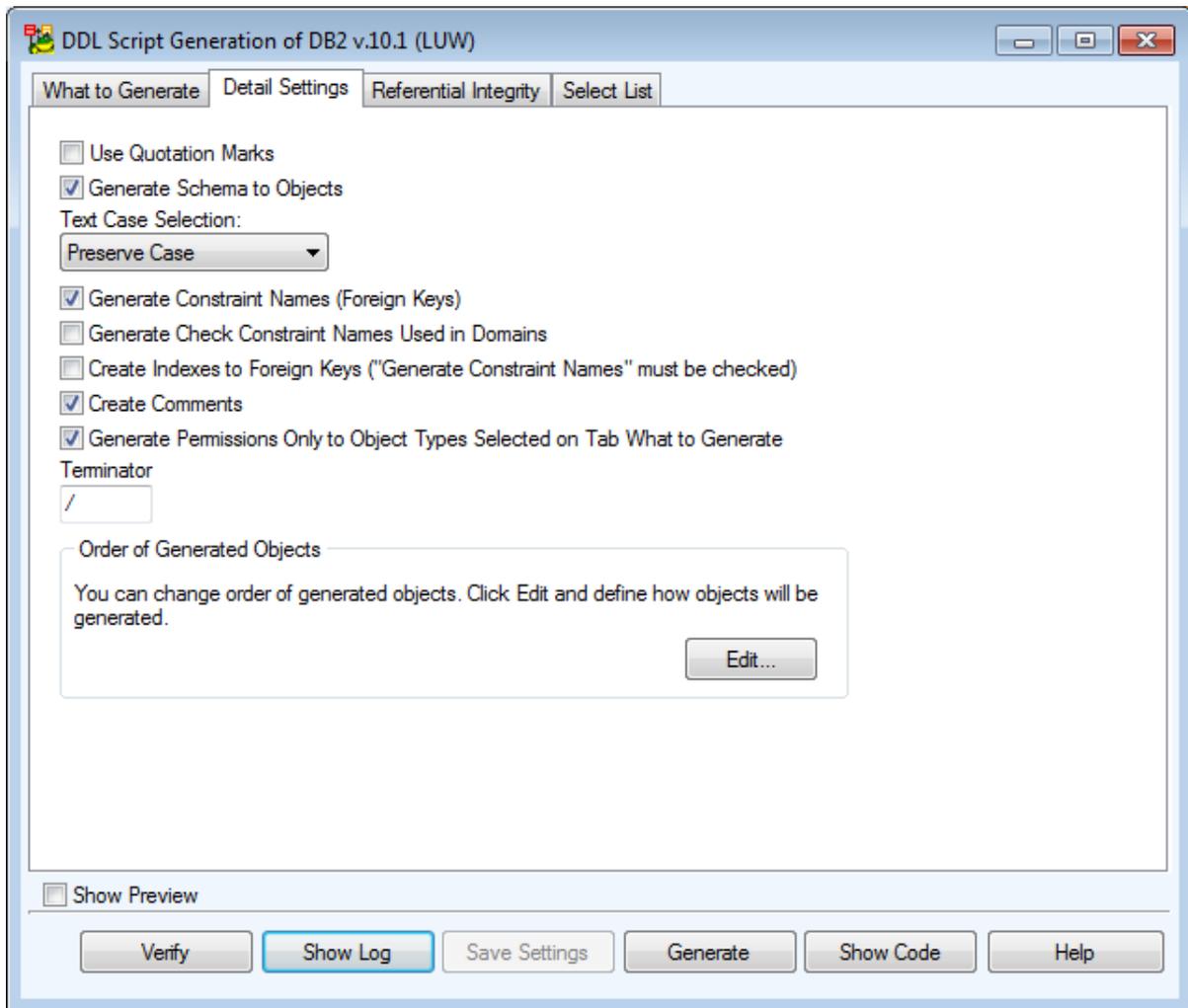
**Connection via ADO:**



Connection via ODBC:



# Script Generation - DB2 v.10.1 (LUW)



# Specifics - DB2 10.5 (LUW)

# Entity

The screenshot shows the 'Entity Properties - E' dialog box. At the top, there are fields for 'Caption' (containing 'Entity') and 'Name' (containing 'Entity'). Below these are several tabs: 'Before Script', 'After Script', 'Notes', 'SQL Preview', 'Relationships', 'Table Properties', 'General', 'Attributes', 'Keys', 'Indexes', 'Check Constraints', 'Triggers', 'Permissions', and 'To Do'. The 'Schema' dropdown is set to '-- Not Specified --'. The 'Settings' section contains the 'Organize By' dropdown, which is highlighted with a red box and set to 'DIMENSIONS'. Below it is a text field for 'Organize by Dimensions (List of Columns)'. Other settings include 'Compression Value' (Not applicable), 'Compression Mode' (Not Enabled), 'Security Policy Name' (Not Specified), and 'PctFree' (with an 'Append' checkbox). The 'Period Definition' section has 'Period Type' (Not Specified), 'Begin Column Name', and 'End Column Name' fields. There is a large 'Comments' text area and a 'Category' dropdown set to '-- None --'. At the bottom, there is a '<%>' button, a checked 'Generate' checkbox, and 'OK', 'Cancel', 'Apply', and 'Help' buttons.

**Organize by Row Using Insert Time, Organize by Row Using Dimensions and Organize by Column** options added.

Example:

```
CREATE TABLE p_tab4 (a varchar(20), b char(10), c integer) organize by
dimensions (b,c)
```

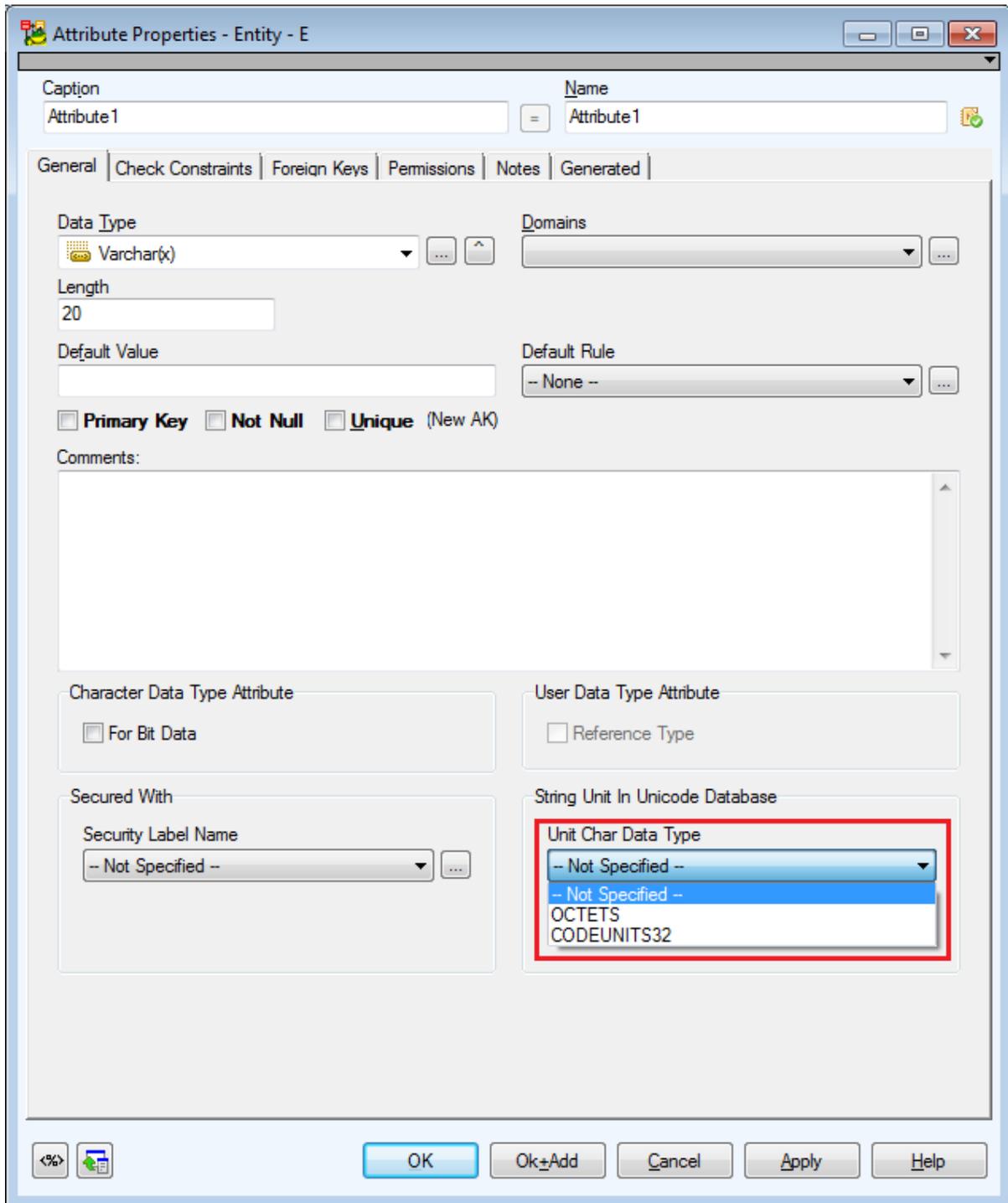
```
CREATE TABLE p_tab11 (a varchar(20), b char(10), c integer) organize by insert time
```

```
CREATE TABLE p_tab5 (a varchar(20), b char(10), c integer) organize by row
```

```
CREATE TABLE p_tab6 (a varchar(20), b char(10), c integer) organize by row using
DIMENSIONS (b)
```

```
CREATE TABLE p_tab7 (a varchar(20), b char(10), c integer) organize by row using
insert time
```

```
CREATE TABLE STAFF (
ID SMALLINT NOT NULL,
NAME VARCHAR(9),
DEPT SMALLINT)
ORGANIZE BY COLUMN;
```



New option to set parameters OCTETS/CODEUNITS16/CODEUNITS32 for the following Data Types:

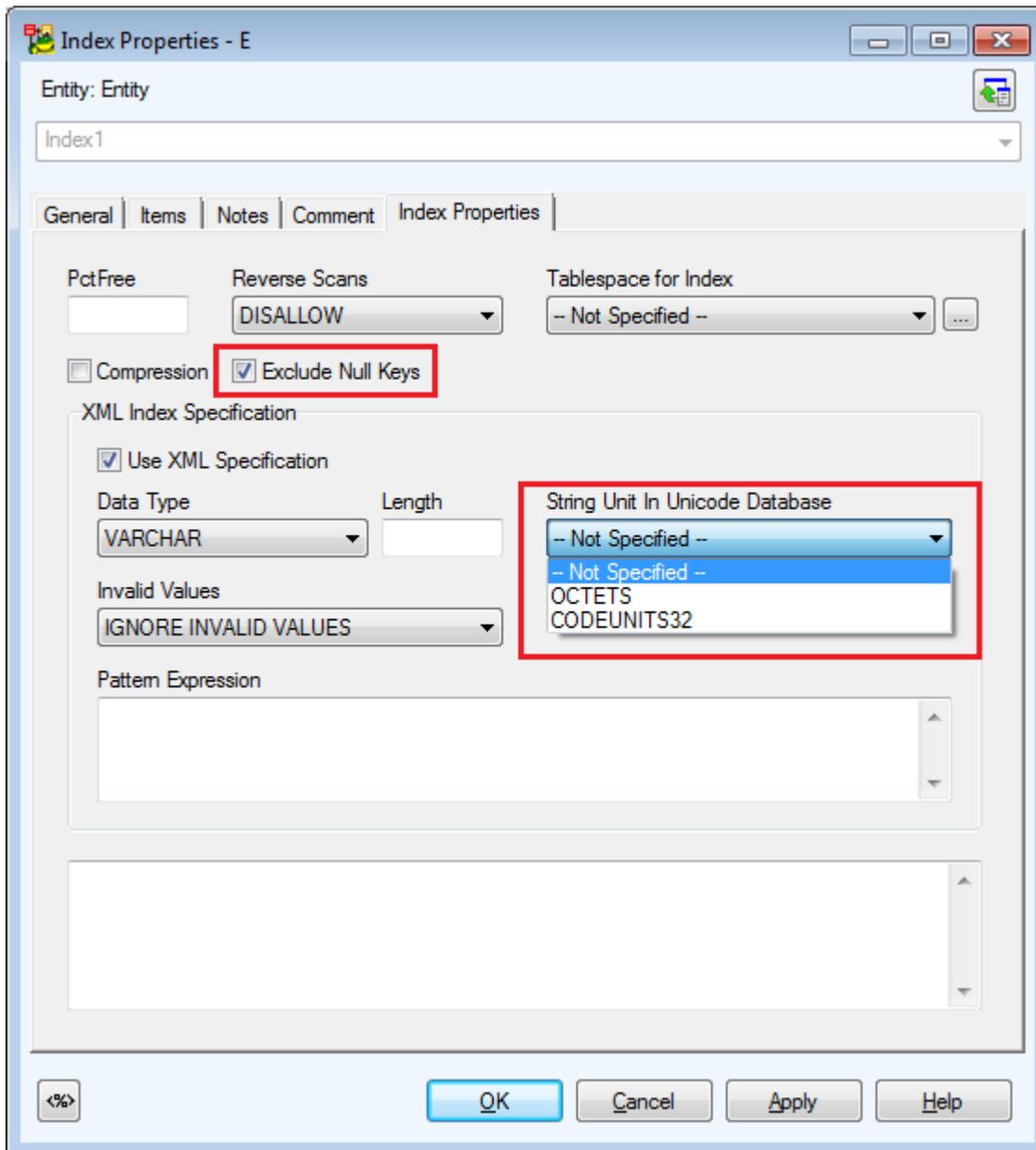
- CHAR, VARCHAR, CLOB **allow** OCTETS/CODEUNITS32
- GRAPHIC, VARGRAPHIC, DBCLOB **allow** CODEUNITS16/CODEUNITS32

**i** | Note: The attribute length must be always set!

Example:

```
CREATE TABLE A_TEST
(
a integer,
b char,
c char(21),
d graphic,
e graphic(12),
f graphic (12 CODEUNITS16),
g char(1 BYTE),
h char (20 OCTETS)
)
```

# Index



- INDEX INCLUDE NULL KEYS / EXCLUDE NULL KEYS

Example:

```
CREATE TABLE P_TAB20 (A char, B char, C char)
```

```
CREATE INDEX ix0 ON P_TAB20 (C) EXCLUDE NULL KEYS
```

```
CREATE INDEX ix11 ON P_TAB20 (C) INCLUDE NULL KEYS
```

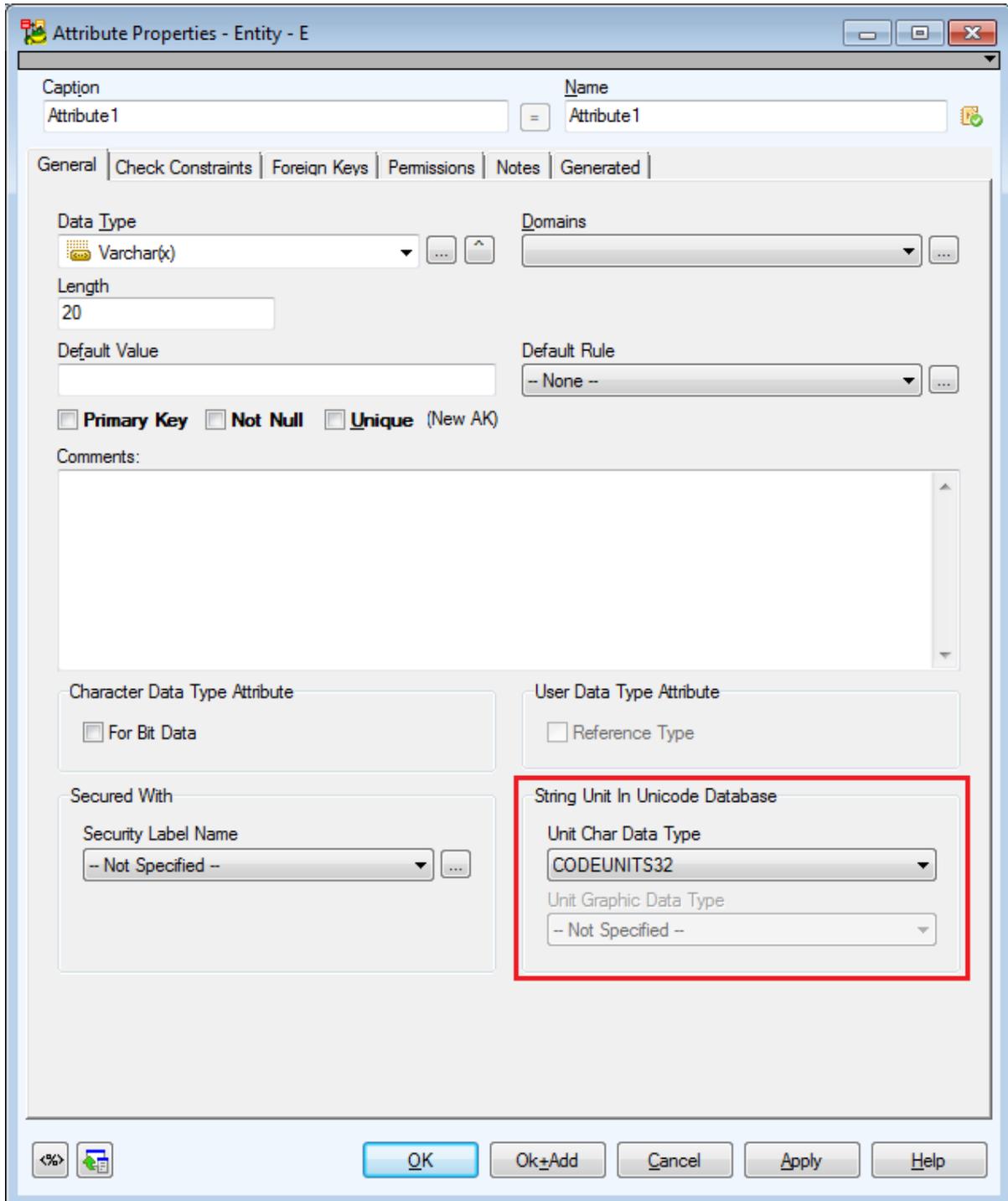
- XML Index Specifications allows parameter OCTETS/CODEUNITS32

**Example:**

```
CREATE TABLE TEST_C (A INTEGER, B XML)

CREATE UNIQUE INDEX MYIDX2 ON TEST_C(B)
GENERATE KEY USING XMLPATTERN '/book/title'
AS SQL VARCHAR(20 OCTETS)
```

# Type



- New option to set parameters of data types
  - CHAR, VARCHAR, CLOB **allow** OCTETS/CODEUNITS32
  - GRAPHIC, VARGRAPHIC, DBCLOB **allow** CODEUNITS16/CODEUNITS32

Example:

```
CREATE TYPE dict3 AS Char(20 OCTETS) WITH COMPARISONS
```

```
CREATE TYPE dict5 AS GRAPHIC(20 CODEUNITS16)
```

```
CREATE TYPE arr7 AS GRAPHIC(1 CODEUNITS16) ARRAY[VARCHAR(8 OCTETS)]
```

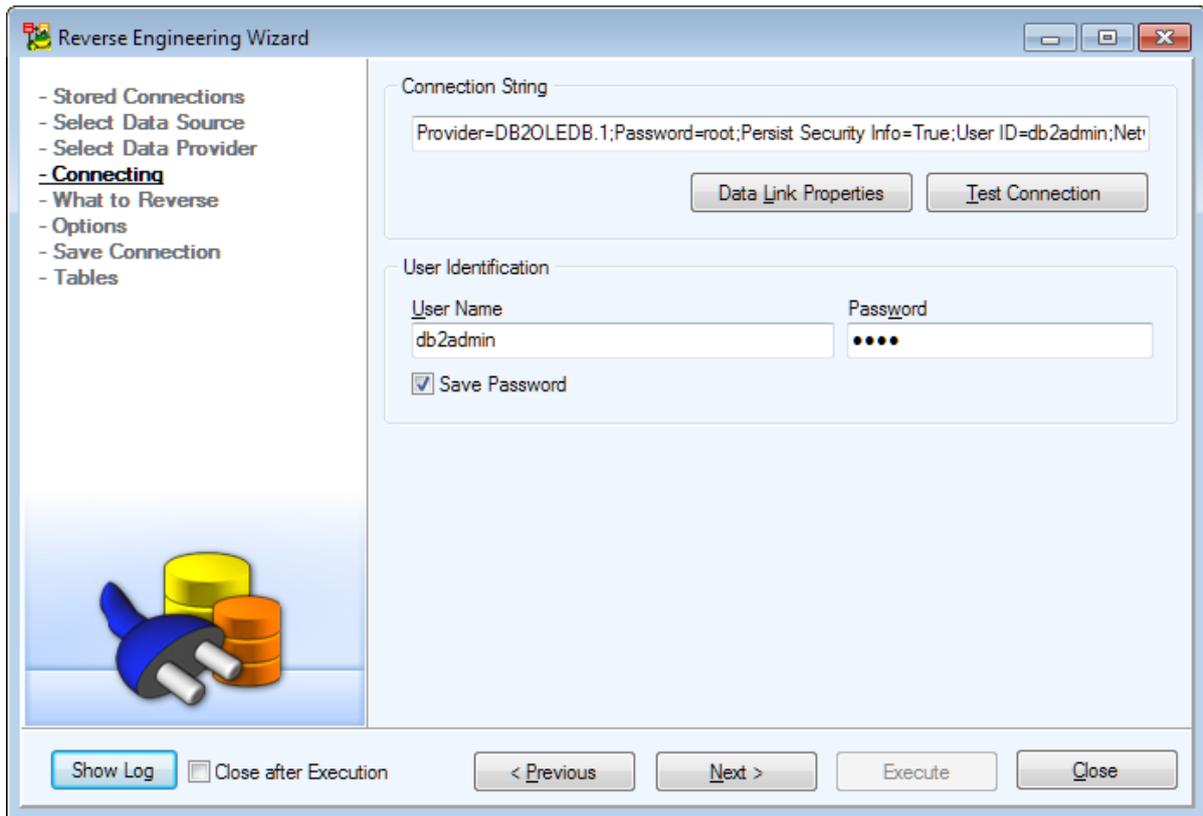
## Reverse Engineering - IBM DB2 LUW

Available **Data Providers** are:

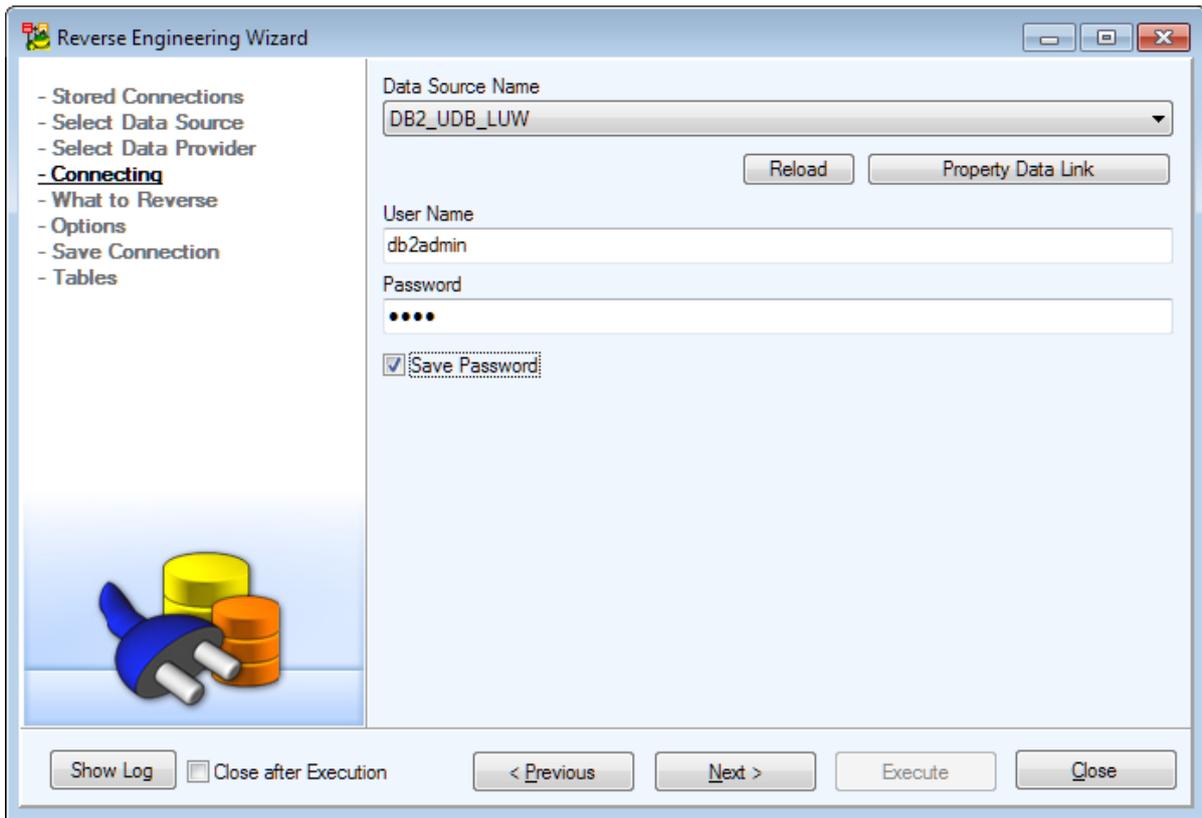
- **Native Connection**
- **Connection via ADO**
- **Connection via ODBC**

**Native Connection:**

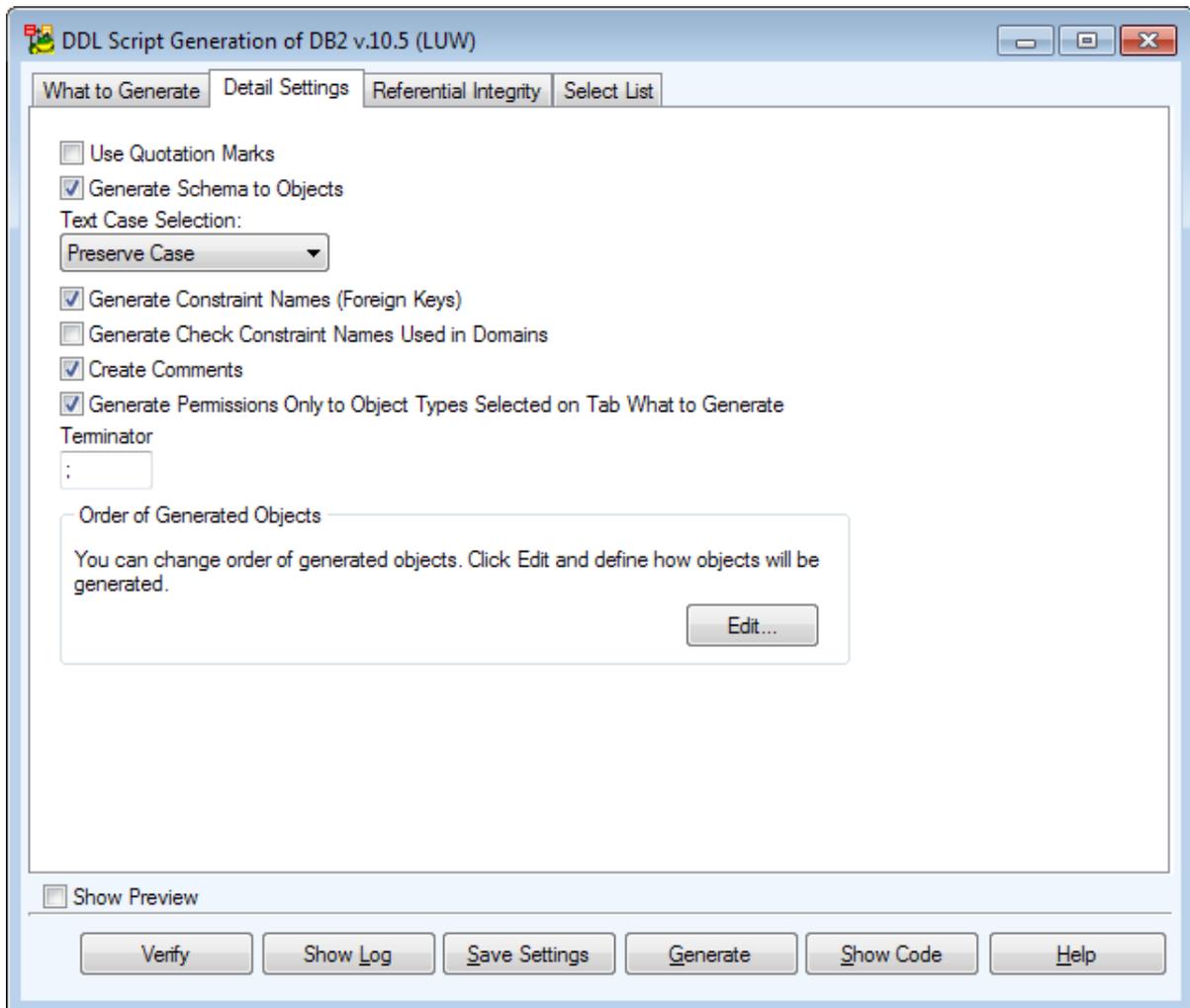
**Connection via ADO:**



Connection via ODBC:



# Script Generation - DB2 v.10.5 (LUW)



# Specifics - DB2 11.1 (LUW)

## Functions and Procedures

- New option STAY RESIDENT NO on the **Function/Procedure Parameters** tab in **Properties**
- Available for external scalar and external table functions and for external procedures
- Set the type of Function/Procedure in **Properties | General**

## Datatypes

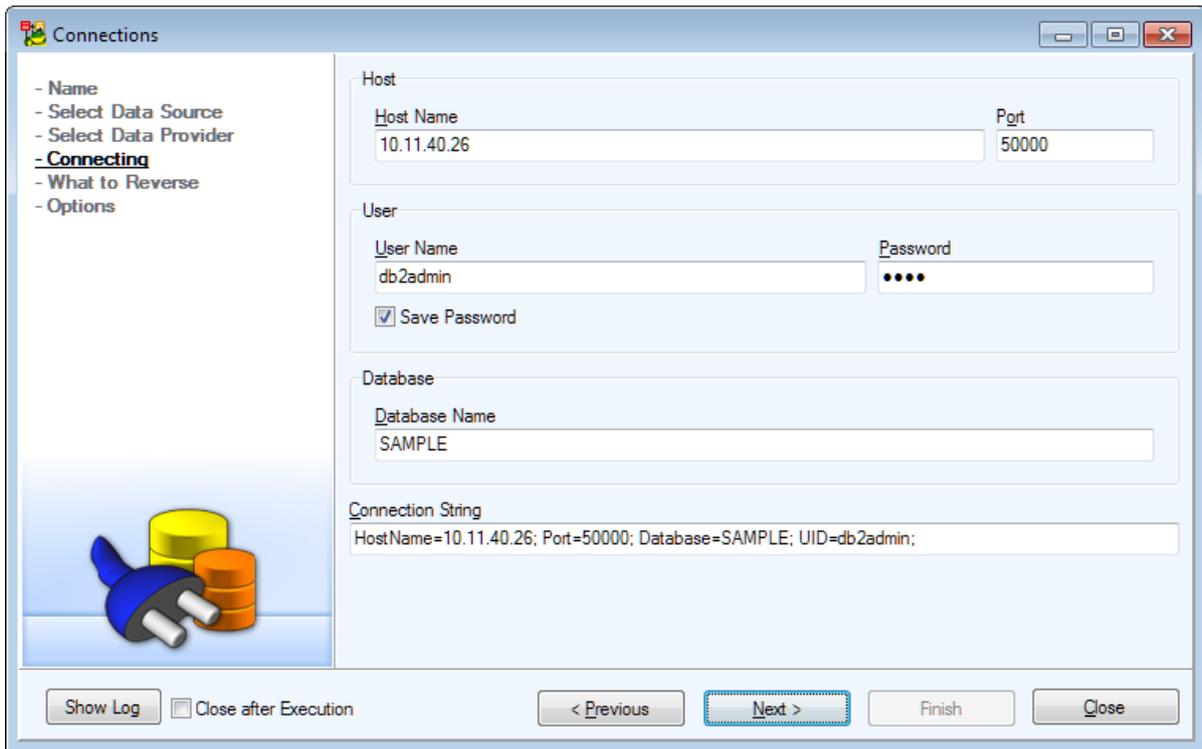
- New datatypes (BINARY, VARBINARY, BINARY VARYING, BOOLEAN) have been implemented

## Reverse Engineering - IBM DB2 LUW

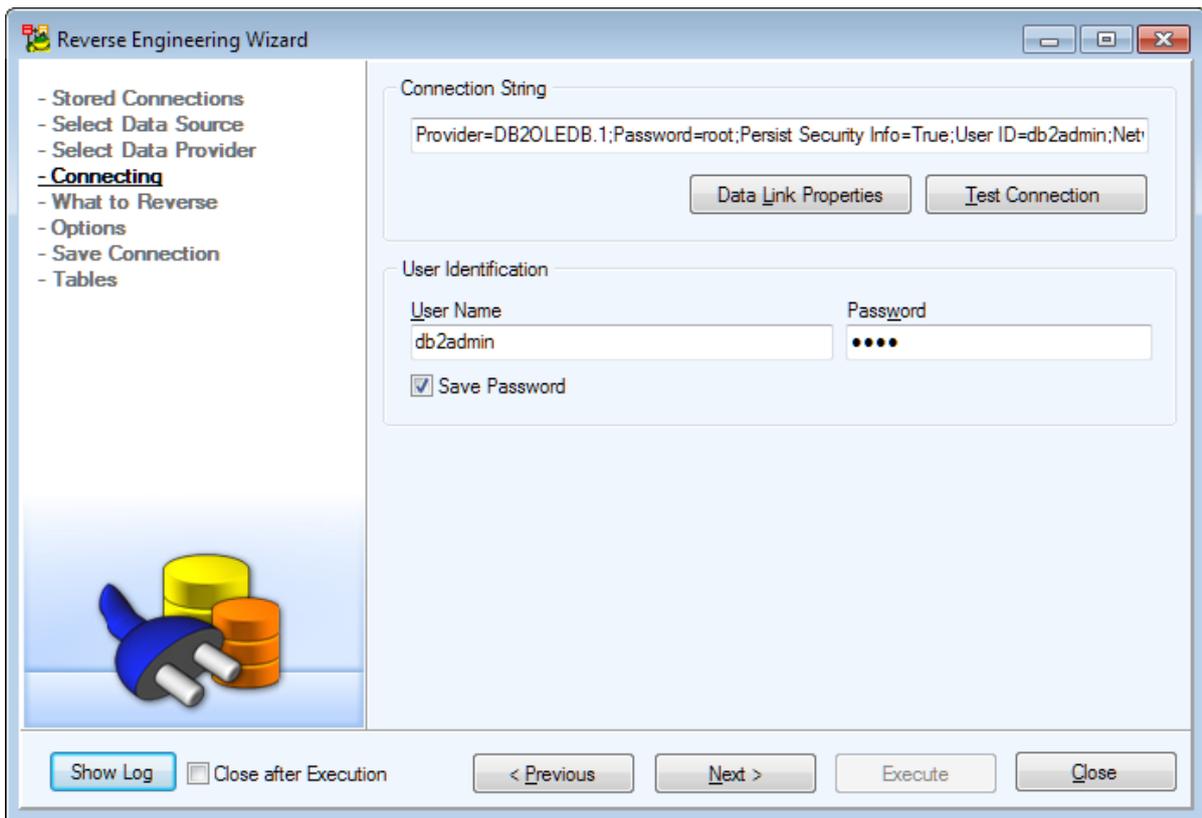
Available **Data Providers** are:

- **Native Connection**
- **Connection via ADO**
- **Connection via ODBC**

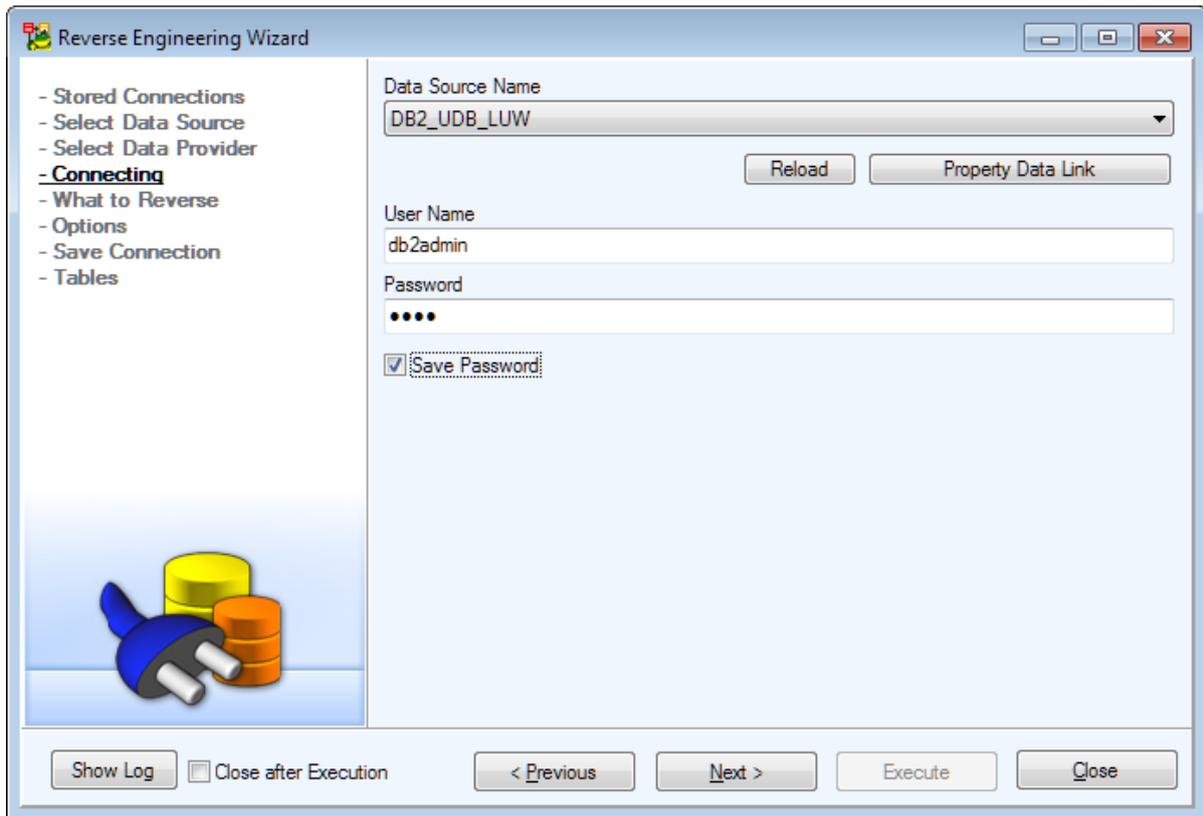
**Native Connection:**



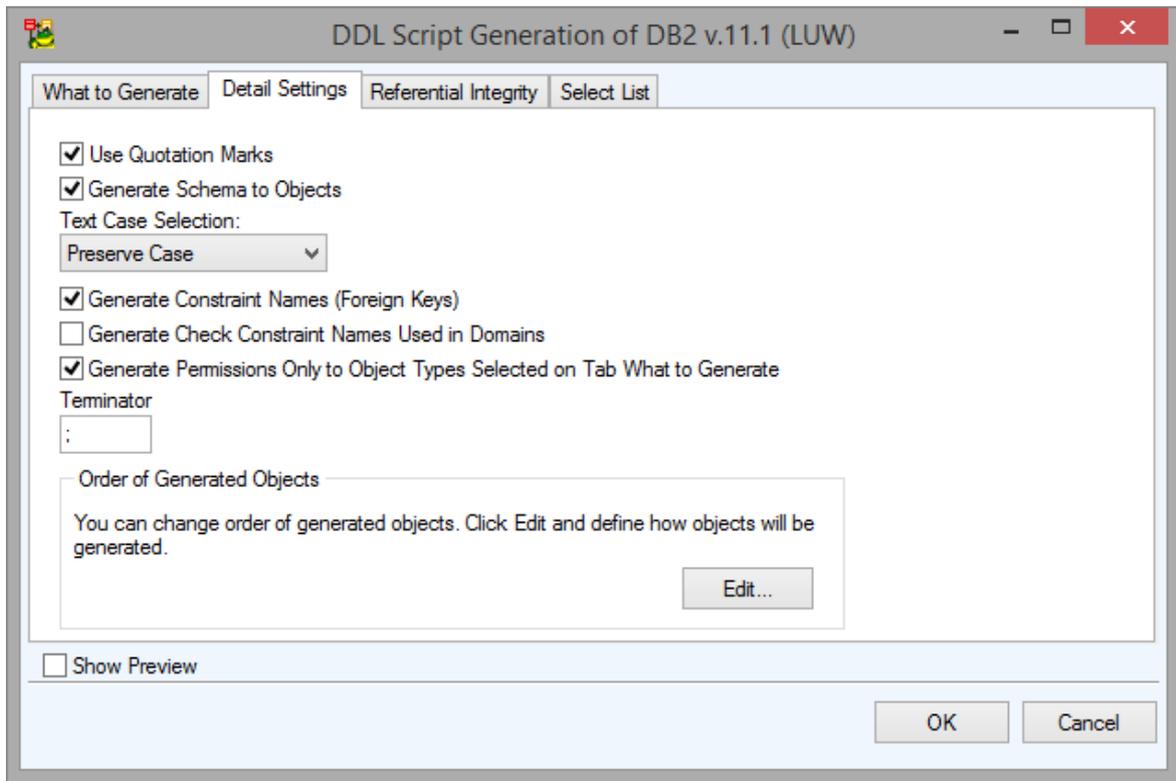
**Connection via ADO:**



**Connection via ODBC:**



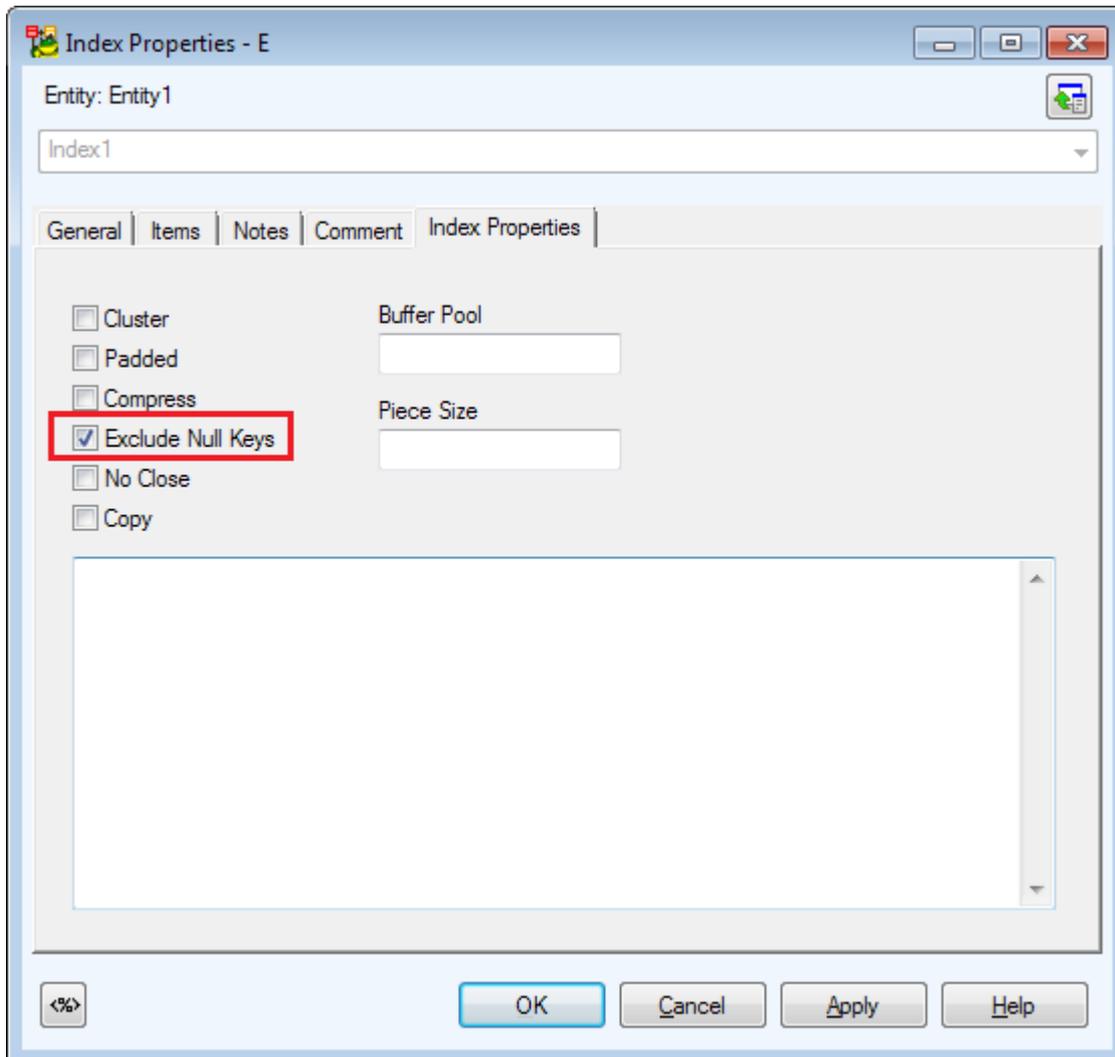
# Script Generation - DB2 v.11.1 (LUW)



# Specifics - DB2 z/OS v. 11

## Index

### Exclude Null Keys



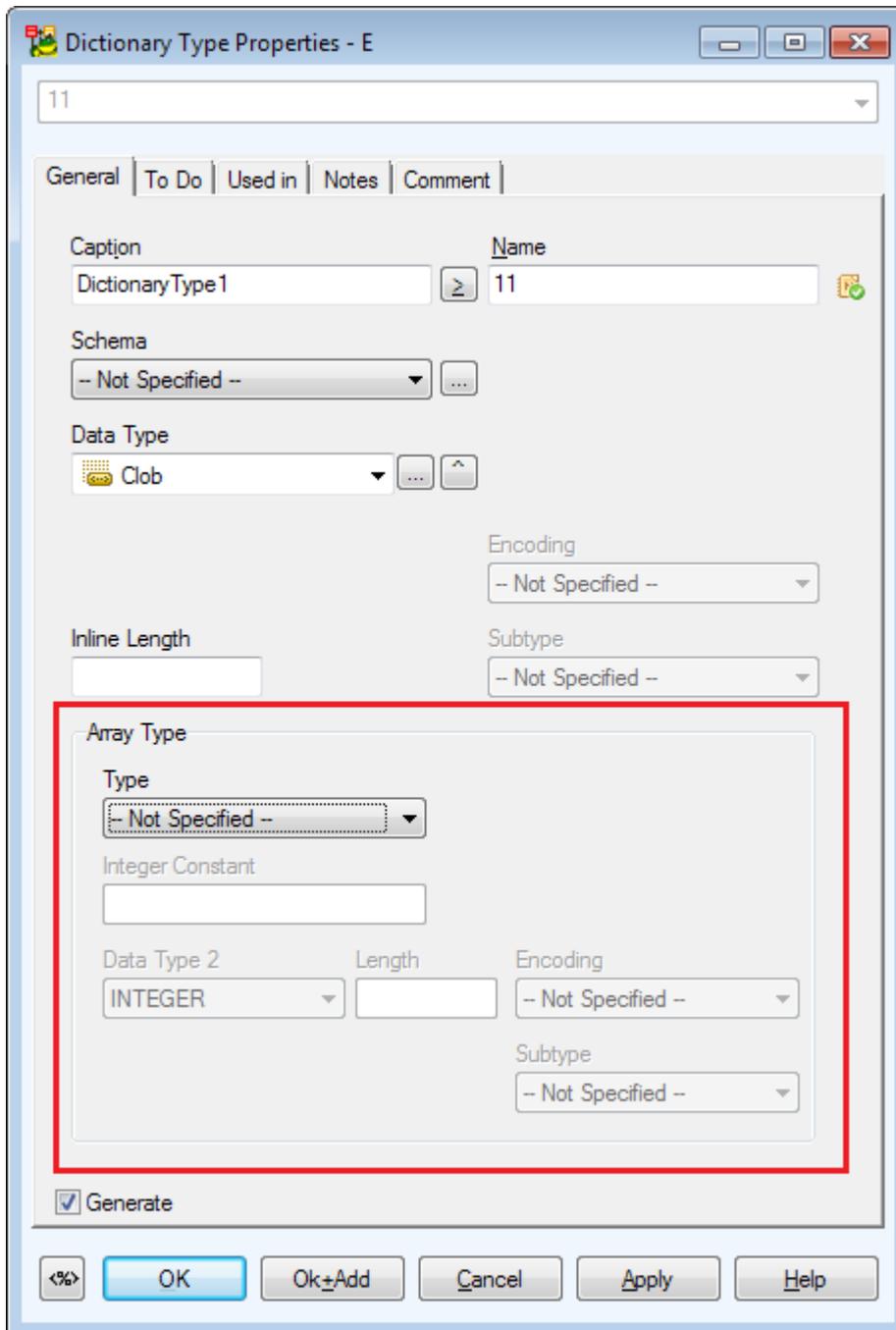
New option to **Include/Exclude Null Keys**

## Business Time Period

**Business Time Period** added in **Entity Properties** dialog, tab **General**.

# Dictionary Type

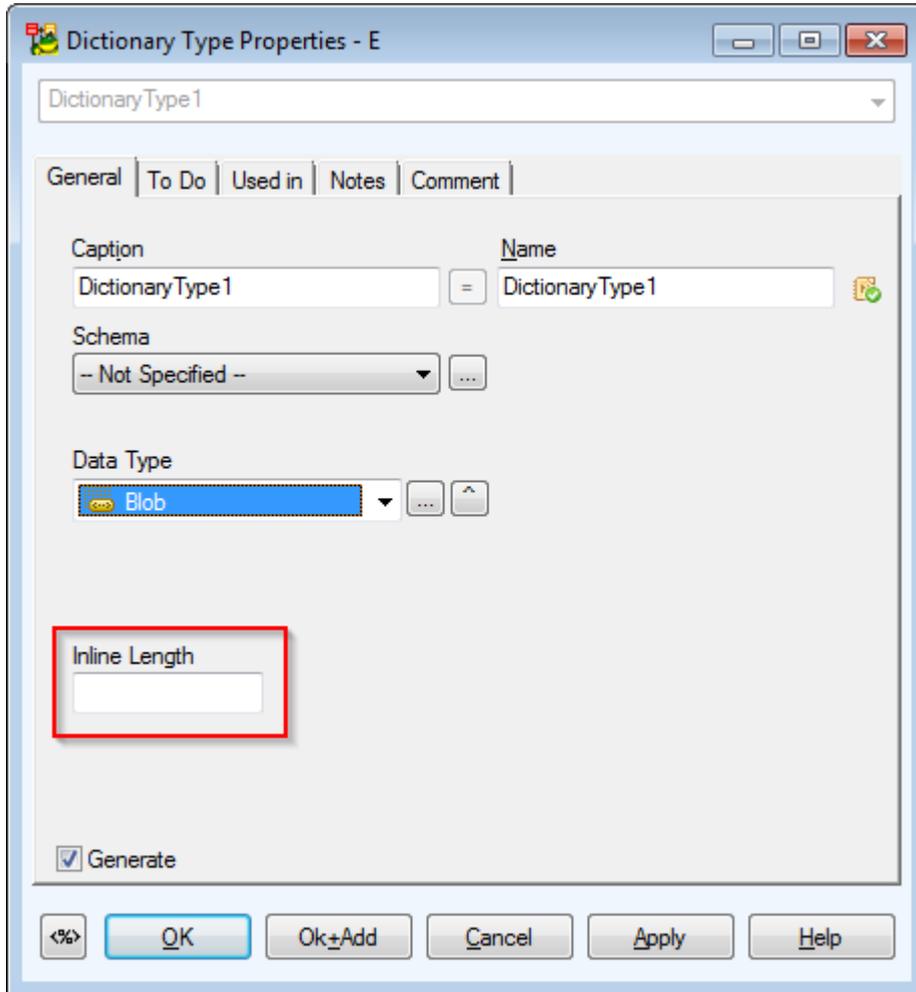
## Array Type



New Type available for Dictionary Types - **Array Type**.

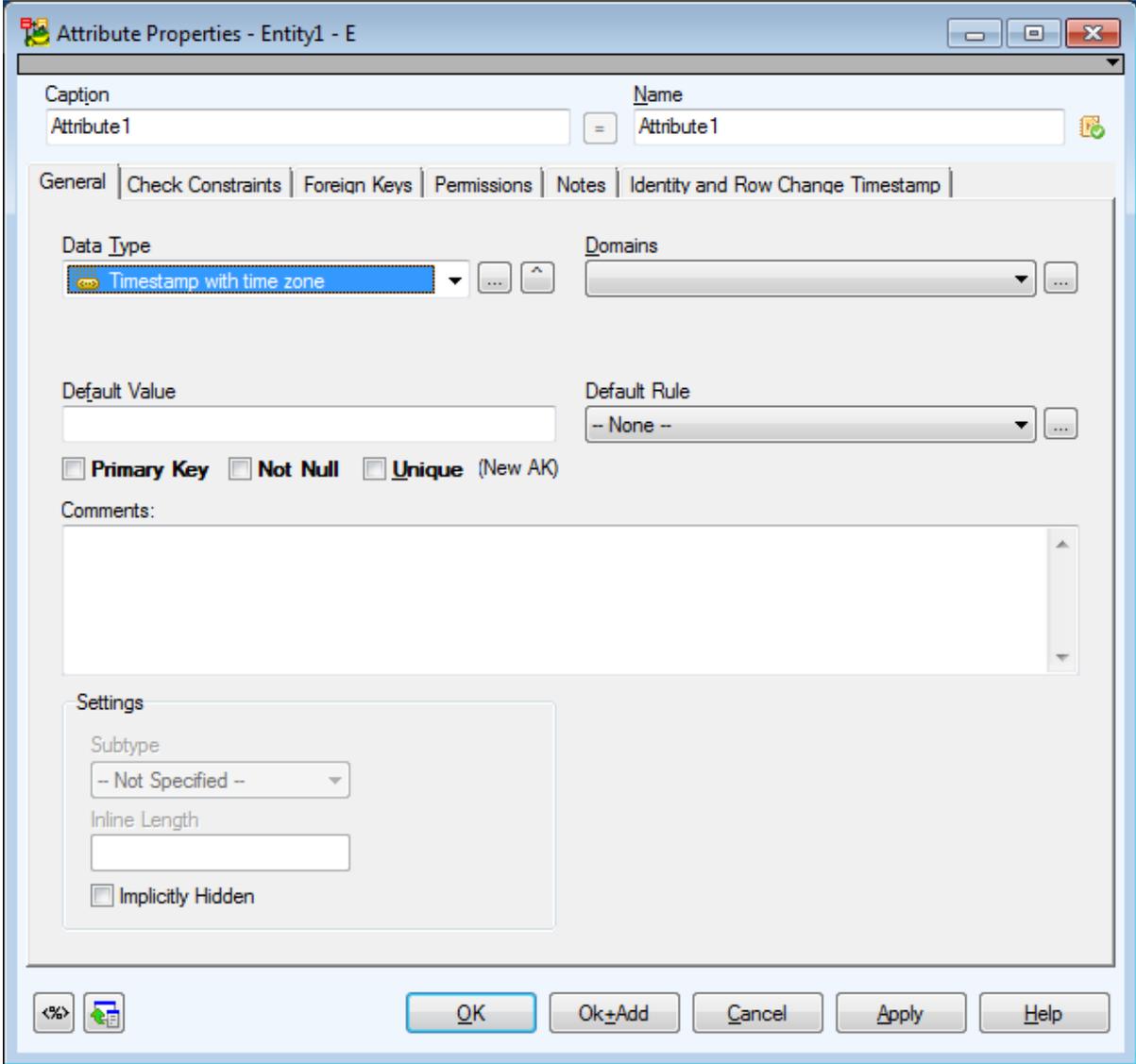
# Inline Length parameter

**Inline Length** parameter for BLOB, CLOB and DBCLOB user data types/dictionary types.



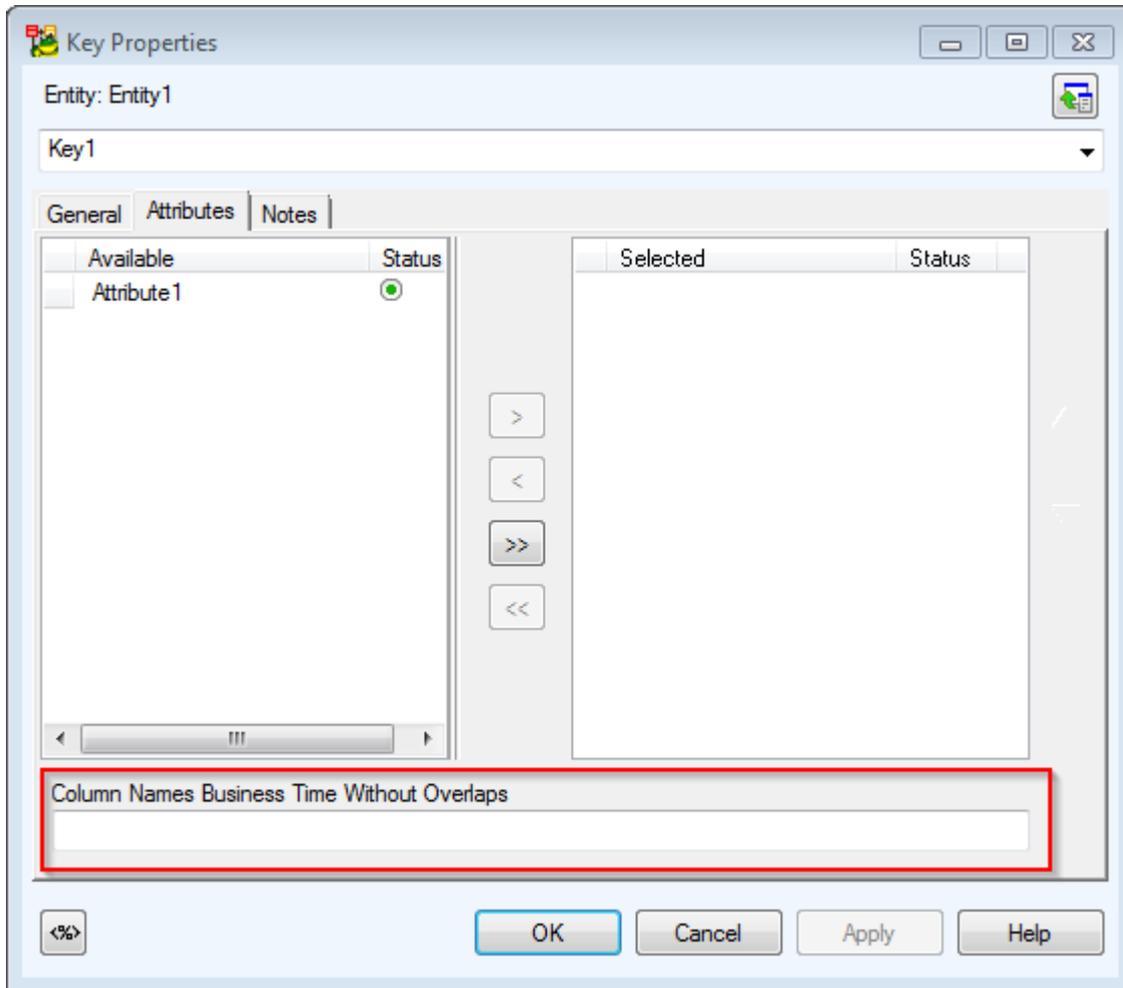
# Attribute

New data types *Timestamp(x)*, *Timestamp with time zone*, *Timestamp(x) with time zone*.

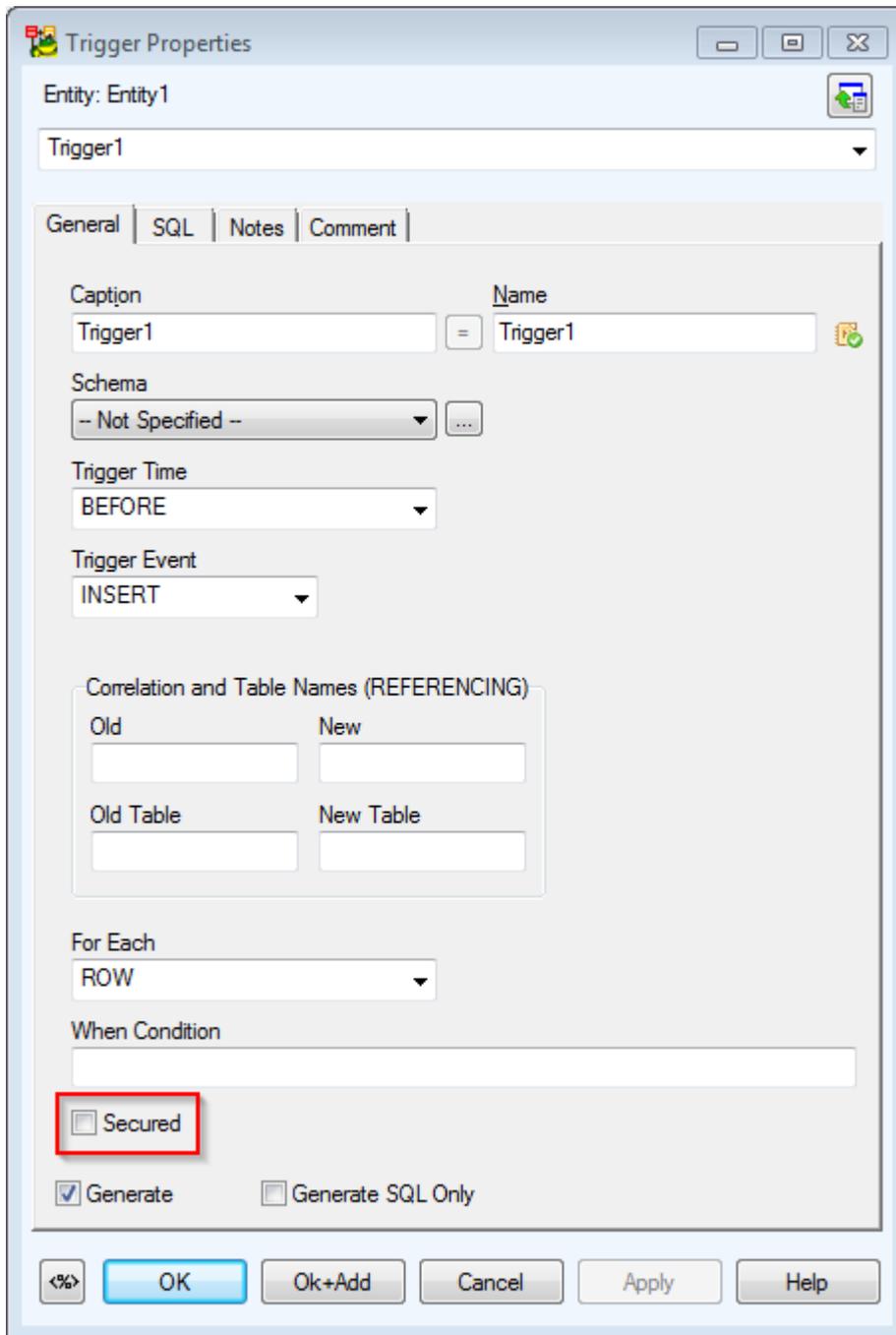


# Key

**Column Names Business Time Without Overlaps** box added in **Key Properties** dialog, tab **Attributes**, and **Index Properties** dialog, tab **Items**.

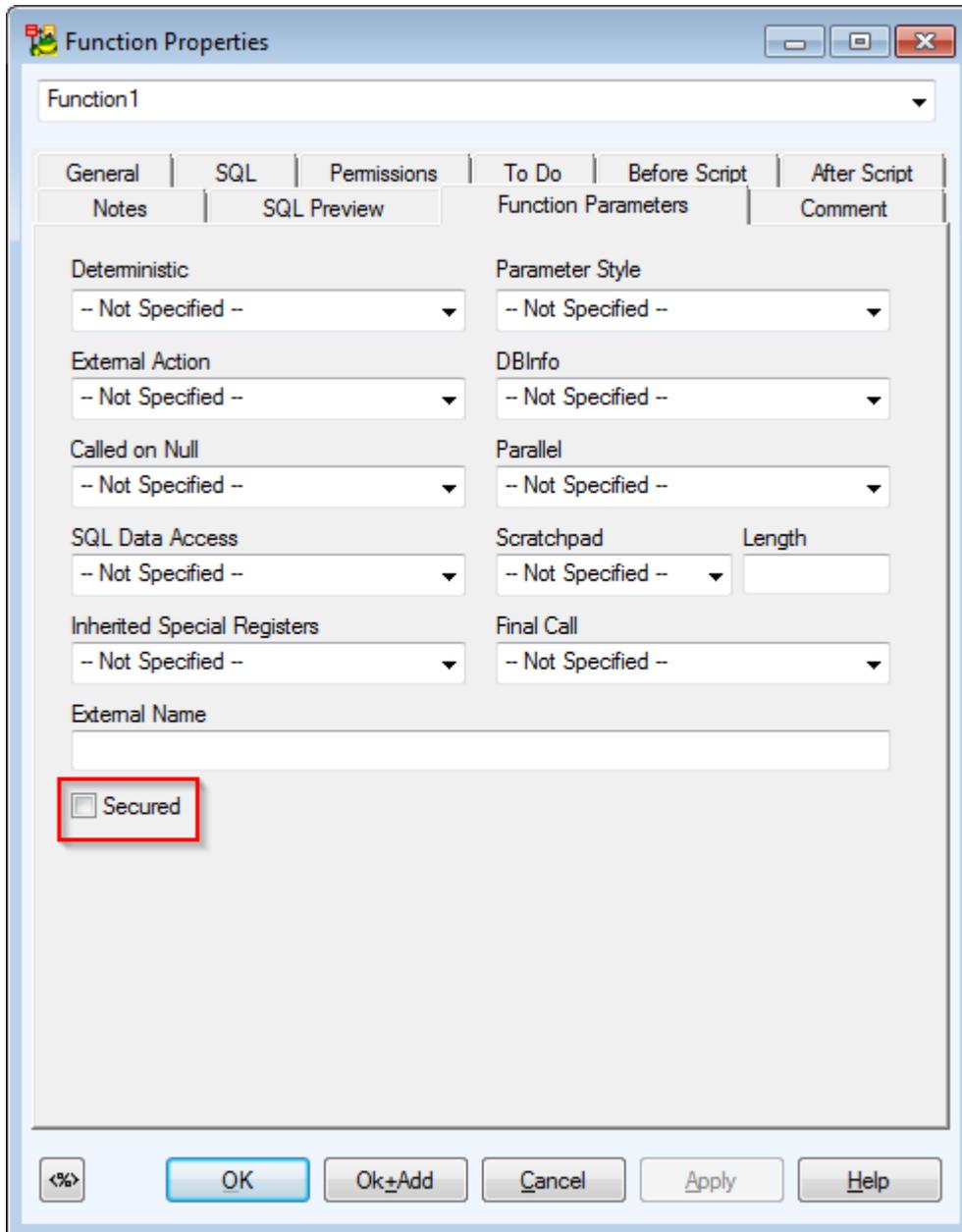


# Trigger



# Function

Checkbox **SECURED** in **Trigger Properties** dialog and **Function Properties** dialog for function types *External Scalar and Table* and *SQL*.

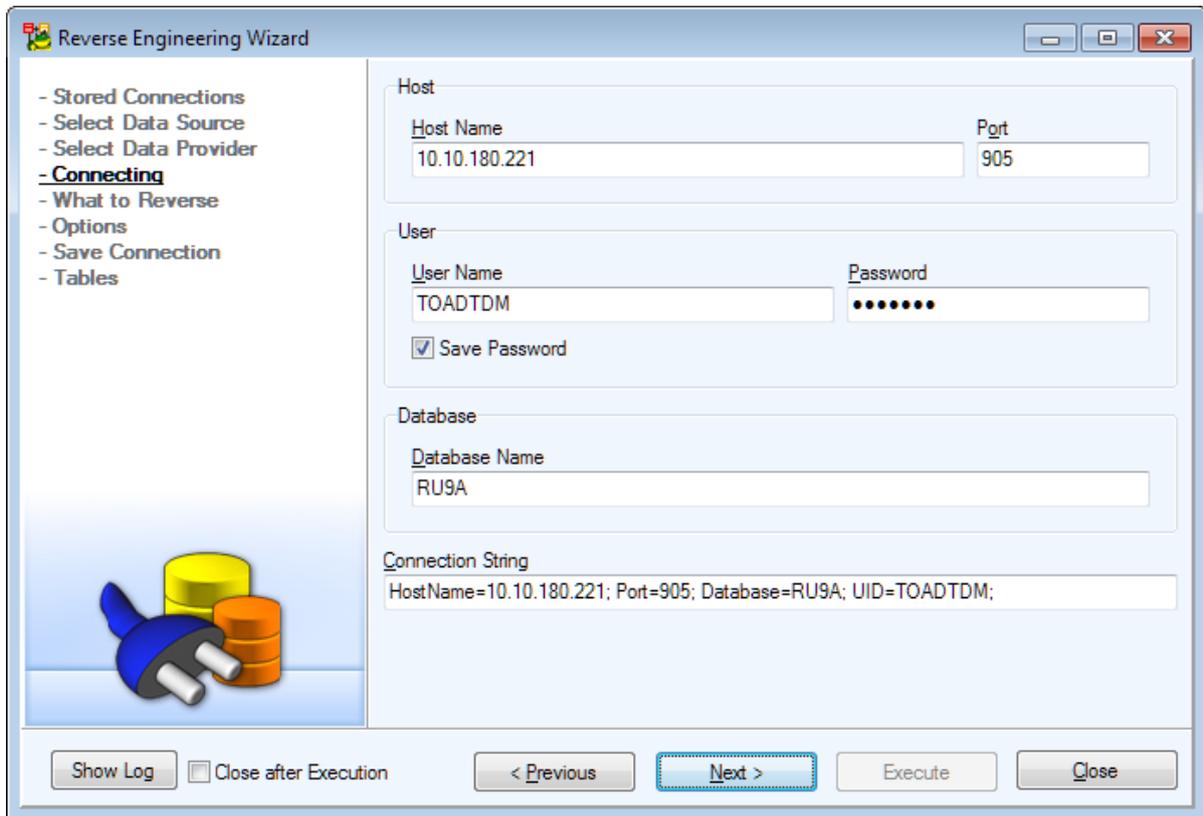


# Reverse Engineering - DB2 z/OS v. 11

Available **Data Providers** are:

- **Native Connection**
- **Connection via ODBC**

**Native Connection:**

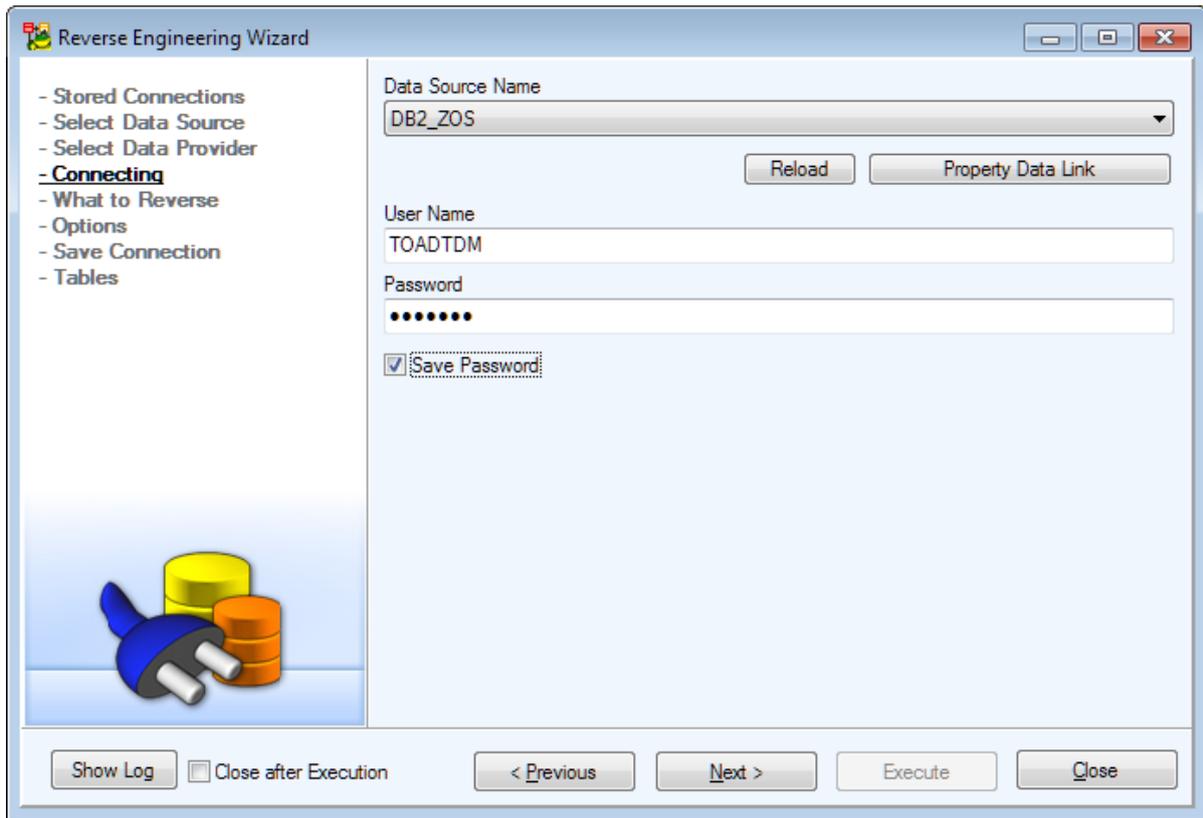


The screenshot shows the 'Reverse Engineering Wizard' dialog box. On the left, a navigation pane lists steps: '- Stored Connections', '- Select Data Source', '- Select Data Provider', '- **Connecting**', '- What to Reverse', '- Options', '- Save Connection', and '- Tables'. The 'Connecting' step is highlighted. Below the list is an icon of a blue plug and two database cylinders (one yellow, one orange). The main area contains the following fields:

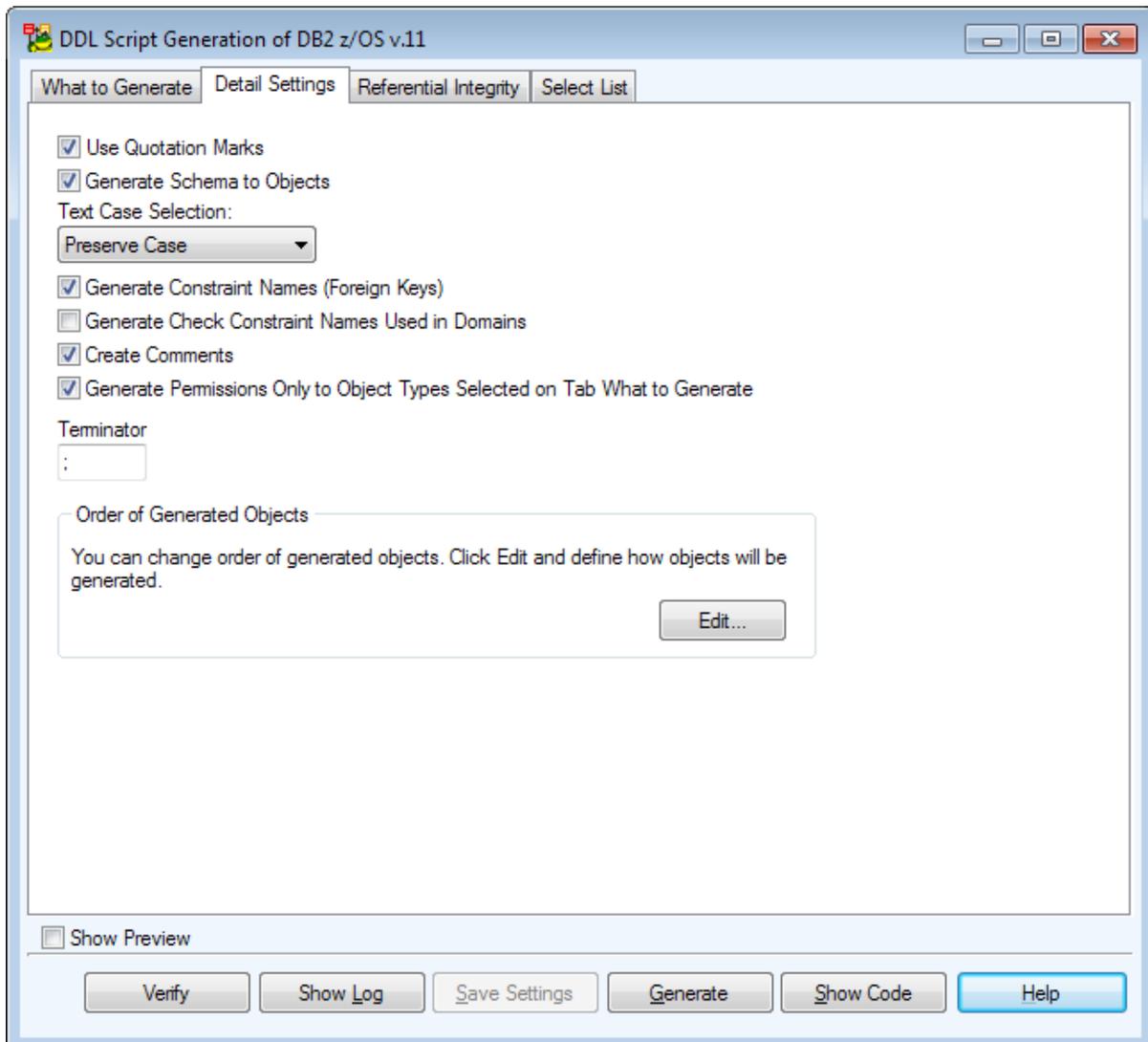
- Host:** Host Name: 10.10.180.221; Port: 905
- User:** User Name: TOADTDM; Password: [masked];  Save Password
- Database:** Database Name: RU9A
- Connection String:** HostName=10.10.180.221; Port=905; Database=RU9A; UID=TOADTDM;

At the bottom, there are buttons for 'Show Log', 'Close after Execution' (checkbox), '< Previous', 'Next >', 'Execute', and 'Close'.

**Connection via ODBC:**



# Script Generation - DB2 z/OS v. 11



# Specifics - Greenplum 4.1

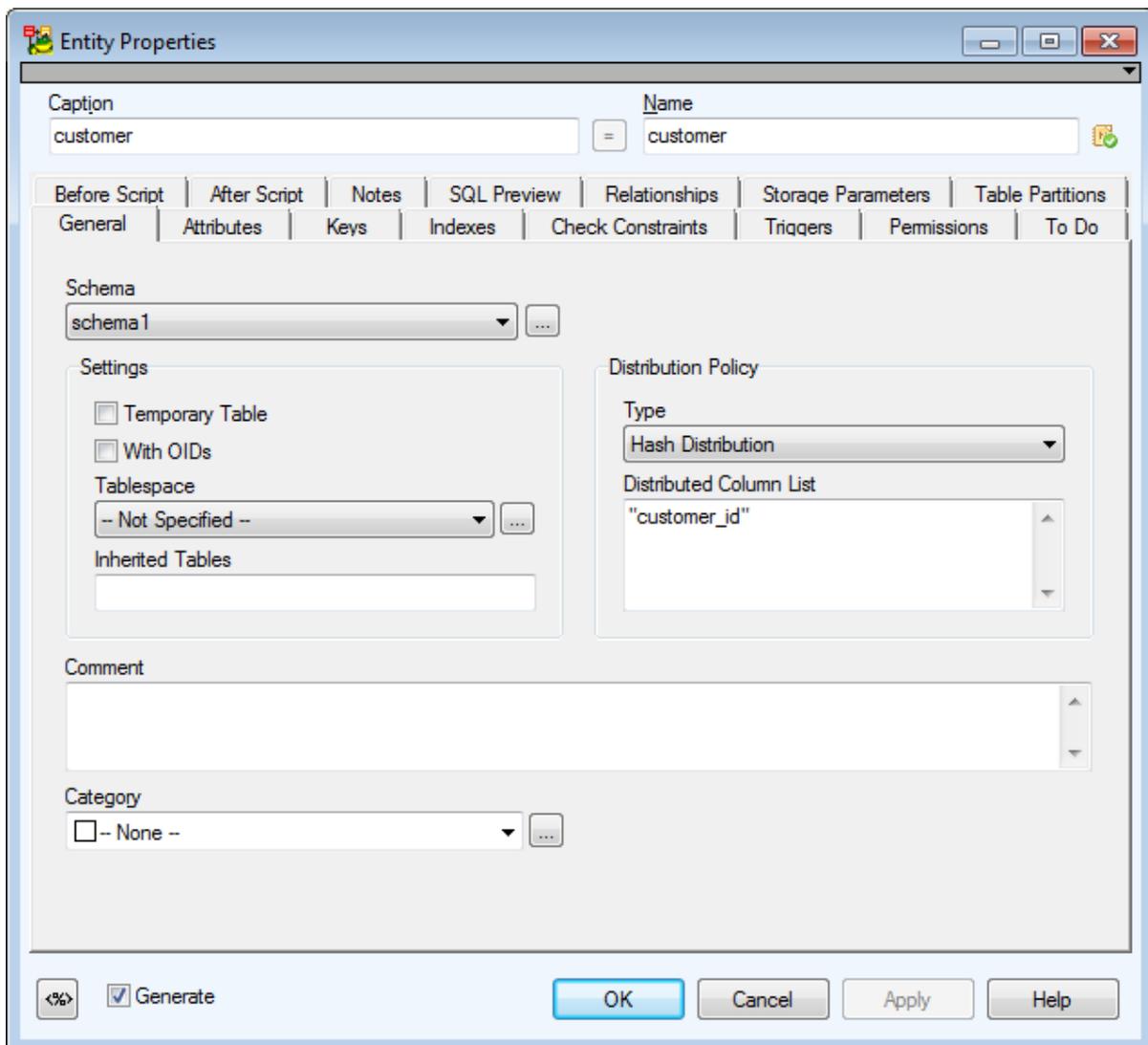
Greenplum database is based on PostgreSQL 8.2, which defines basic features and structure.

**Supported features** - SQL/DDL script generation, reverse engineering, HTML/RTF/PDF reports, change scripting, verification.

**Permissions** can be set for entities, views and functions.

## Entity

### General tab



## Storage Parameters tab

Fillfactor	<input type="text" value="20"/>
<input checked="" type="checkbox"/> Append-Only Table	
Orientation	<input type="text" value="-- Not Specified --"/>
Compress Type	<input type="text" value="NONE"/>
Compress Level	<input type="text" value="8"/>
Block Size (in bytes)	<input type="text" value="32768"/>

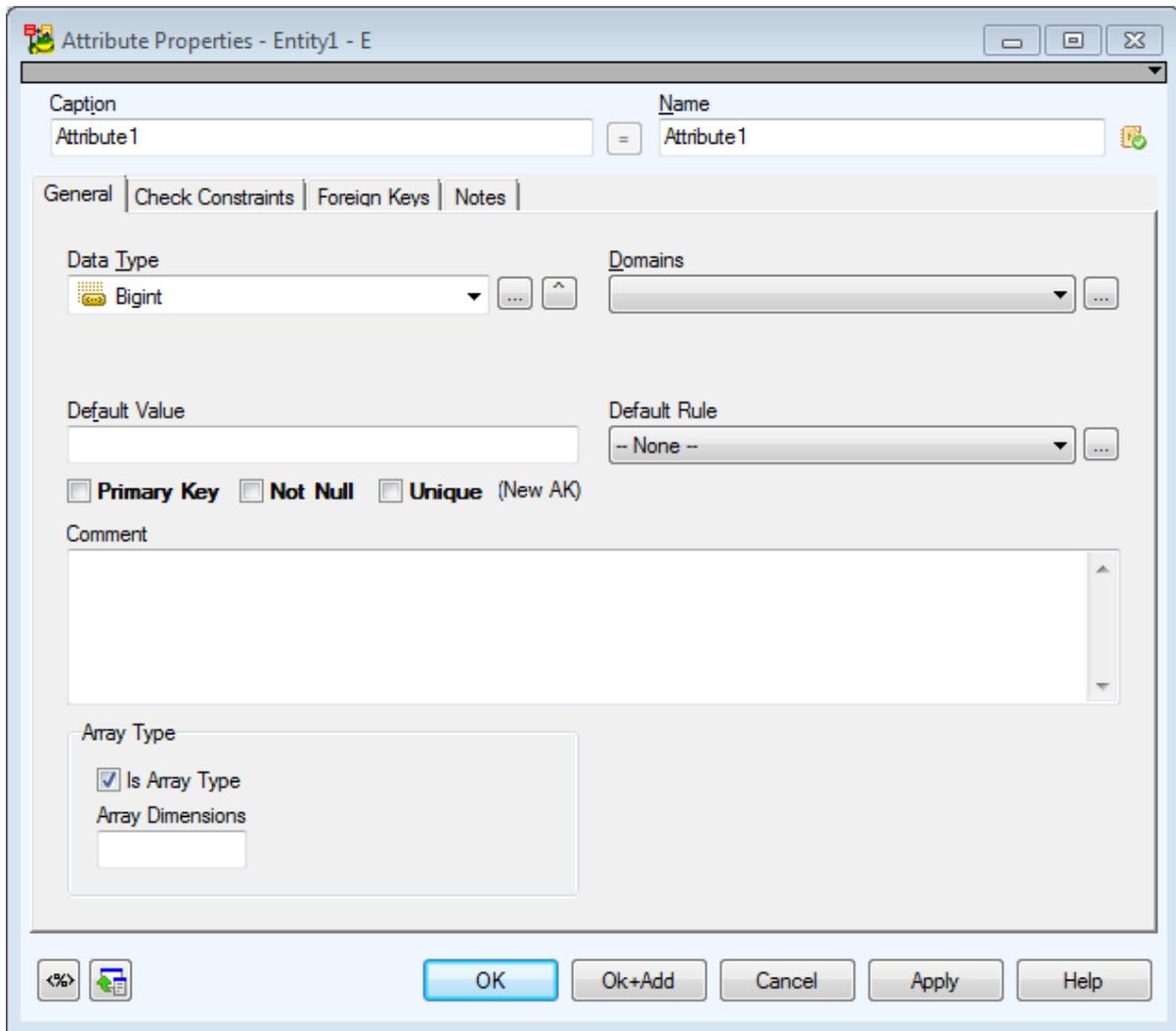
## Table Partitions tab

Define partitions/subpartitions textually.

Before Script	After Script	Notes	SQL Preview	Relationships	Storage Parameters	<b>Table Partitions</b>
<pre>PARTITION BY LIST(col1) (PARTITION part1 VALUES(1,2,3) WITH (fillfactor=11) TABLESPACE tablespace1, VALUES(4,5), DEFAULT PARTITION defpart)</pre>						

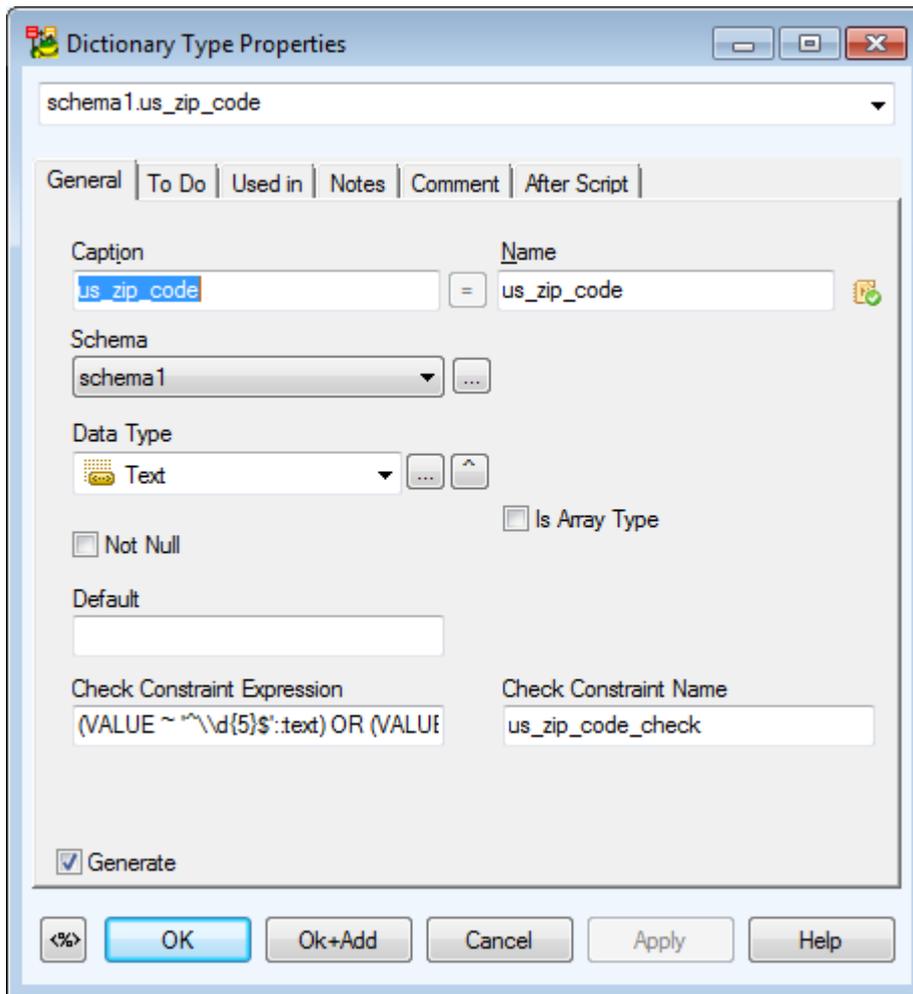
During reverse engineering, sub-tables of partitioned tables are filtered (the tables created because of partitioned tables). The behavior can be influenced in reverse engineering options, in checkbox **Load Sub-Tables of Partitioned Tables**.

# Attribute



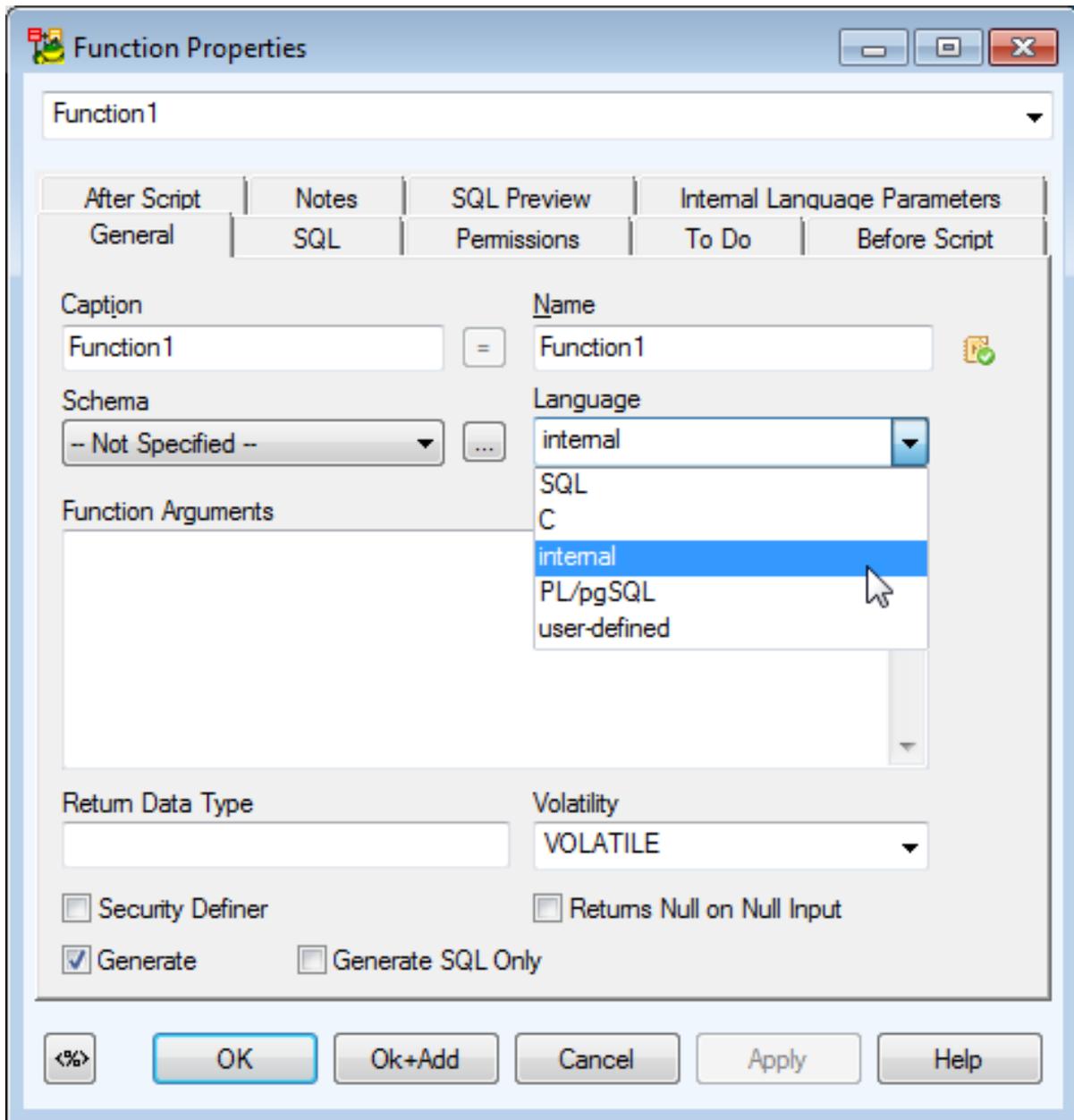
**Array Type** - Select the **Is Array Type** checkbox to enable the **Array Dimension** box.

# Dictionary Type/Domain



**Array Type** - Select the **Is Array Type** checkbox to enable the **Array Dimension** box.

# Function



If C is selected in **Language** combo box, **C Language Parameters** tab appears.

After Script	Notes	SQL Preview	C Language Parameters
File Name			
<input type="text" value="\$libdir/gpextprotocol.so"/>			
Link Symbol			
<input type="text" value="demoprot_import"/>			

If *internal* is selected in **Language** combo box, **Internal Language Parameters** tab appears.

After Script	Notes	SQL Preview	Internal Language Parameters
Internal Function Name			
<input type="text" value="\$libdir/gpextprotocol.so"/>			

## Foreign Key (Referential Integrity)

NO FUNCTIONALITY - Foreign key constraints are not supported in Greenplum database. They can be created but are not enforced.

## Trigger

Triggers can be created but the functionality is considerably limited.

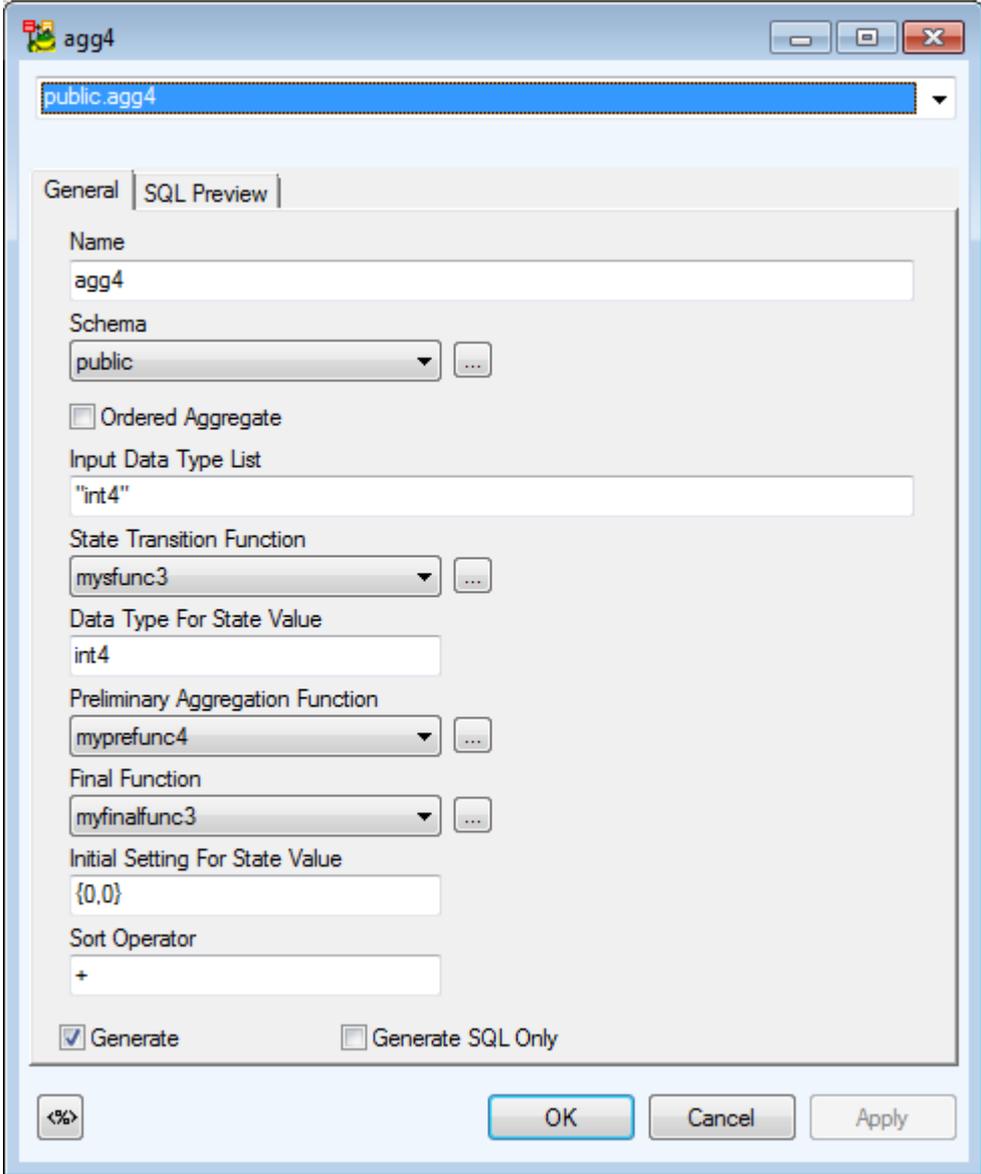
Other supported objects:

- Aggregate
- Sequences
- Rewrite Rules
- External Tables

Objects not supported in Toad Data Modeler:

- Resource Queue
- (Trusted) Protocol
- Filespace

# Aggregate



# Sequence

tab\_seq\_a\_ser\_seq

public.tab\_seq\_a\_ser\_seq

General | SQL Preview | Before Script | After Script | Comment

Name  
tab\_seq\_a\_ser\_seq

Schema  
public

Increment By  
1

Start With  
[Empty]

Maxvalue  
 NOMAXVALUE

Minvalue  
 NOMINVALUE

Cache  
1

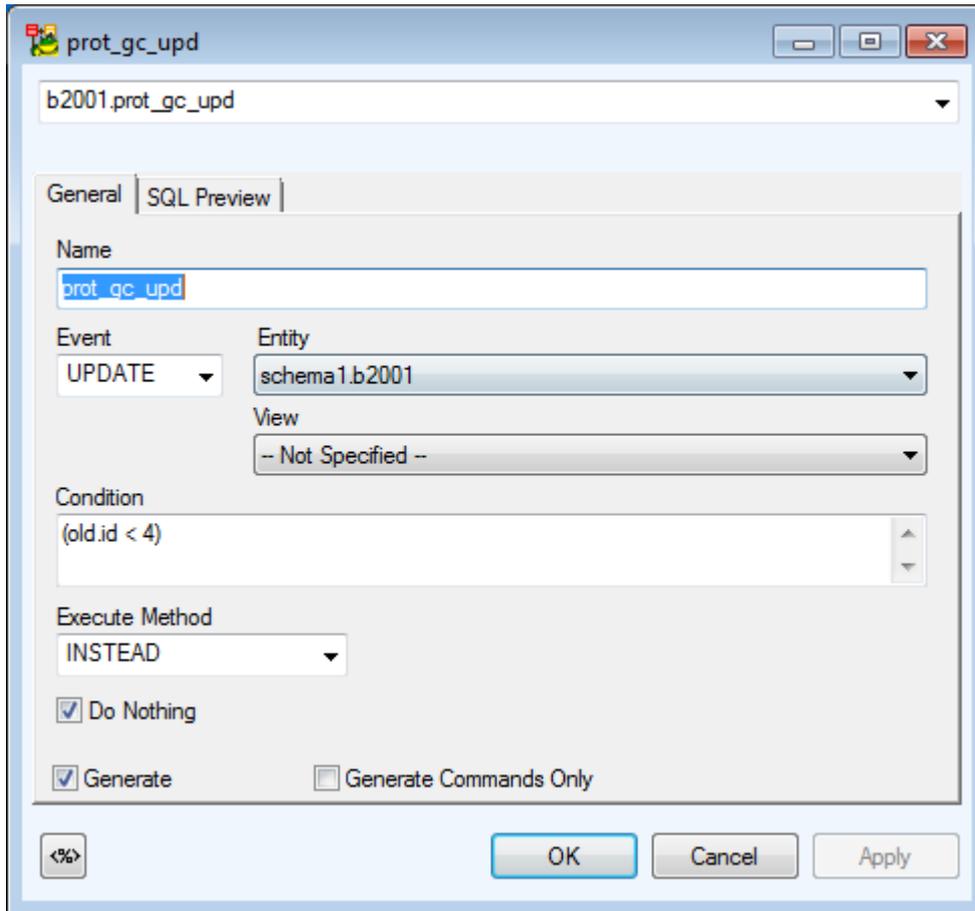
NOCYCLE

Owned By  
Table: tab\_seq  
Column: a\_ser

Generate  Generate SQL Only

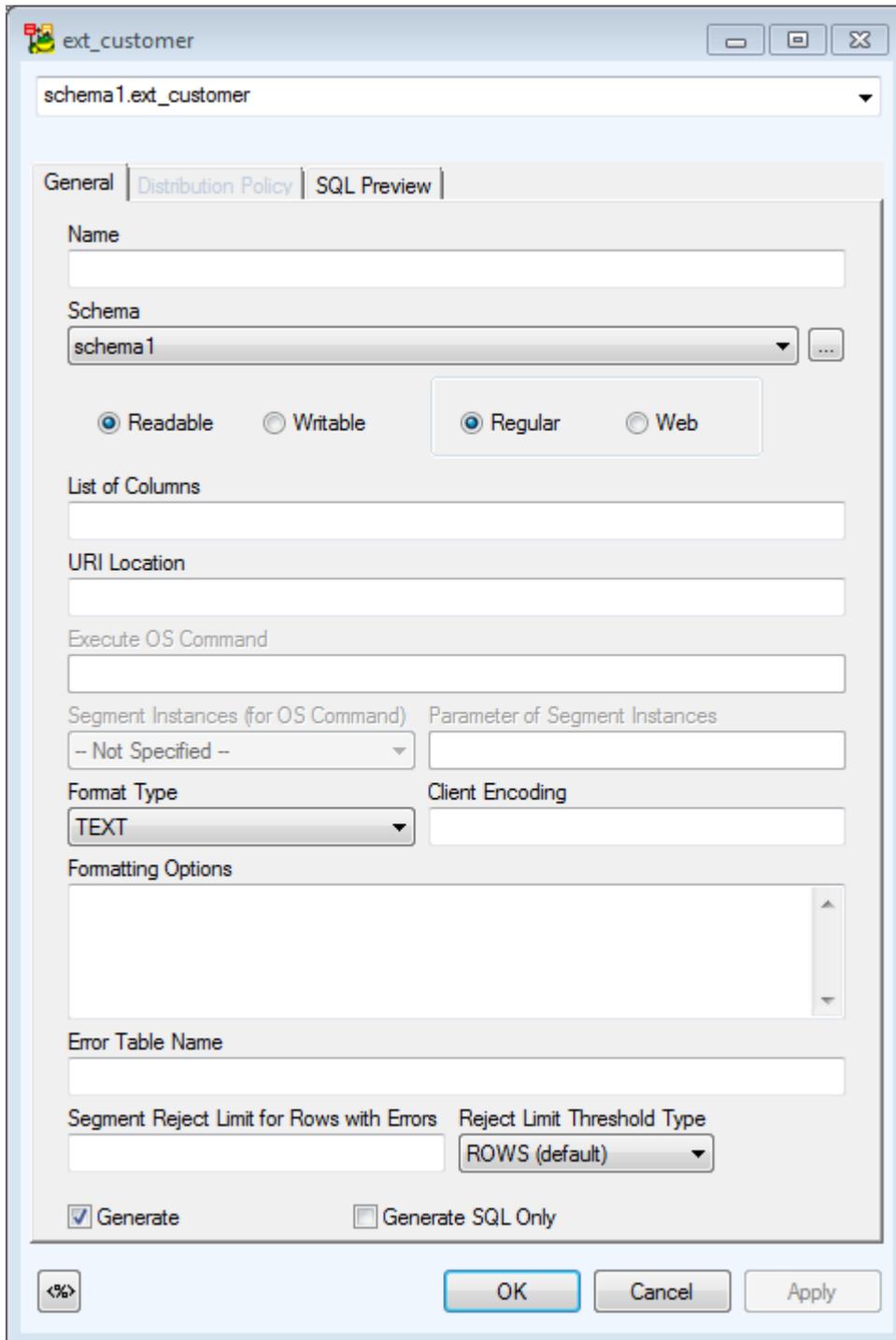
<%> OK Cancel Apply

# Rewrite Rules



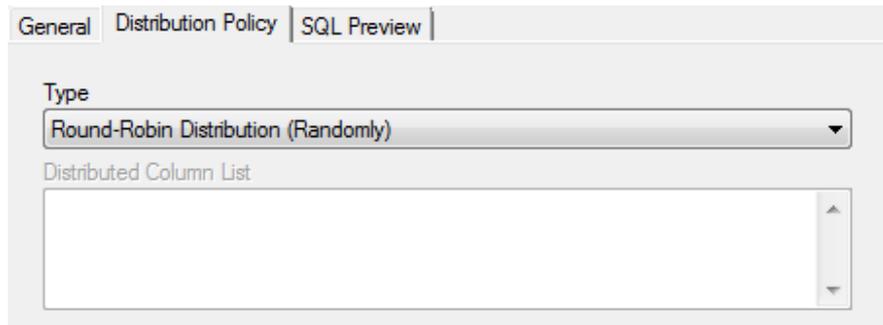
# External Tables

Object specifics for Greenplum db not present in PosgtreSQL 8.2.



- Set External Table to be **Readable/Writable**

**Writable** External Table - enables the **Distribution Policy** tab



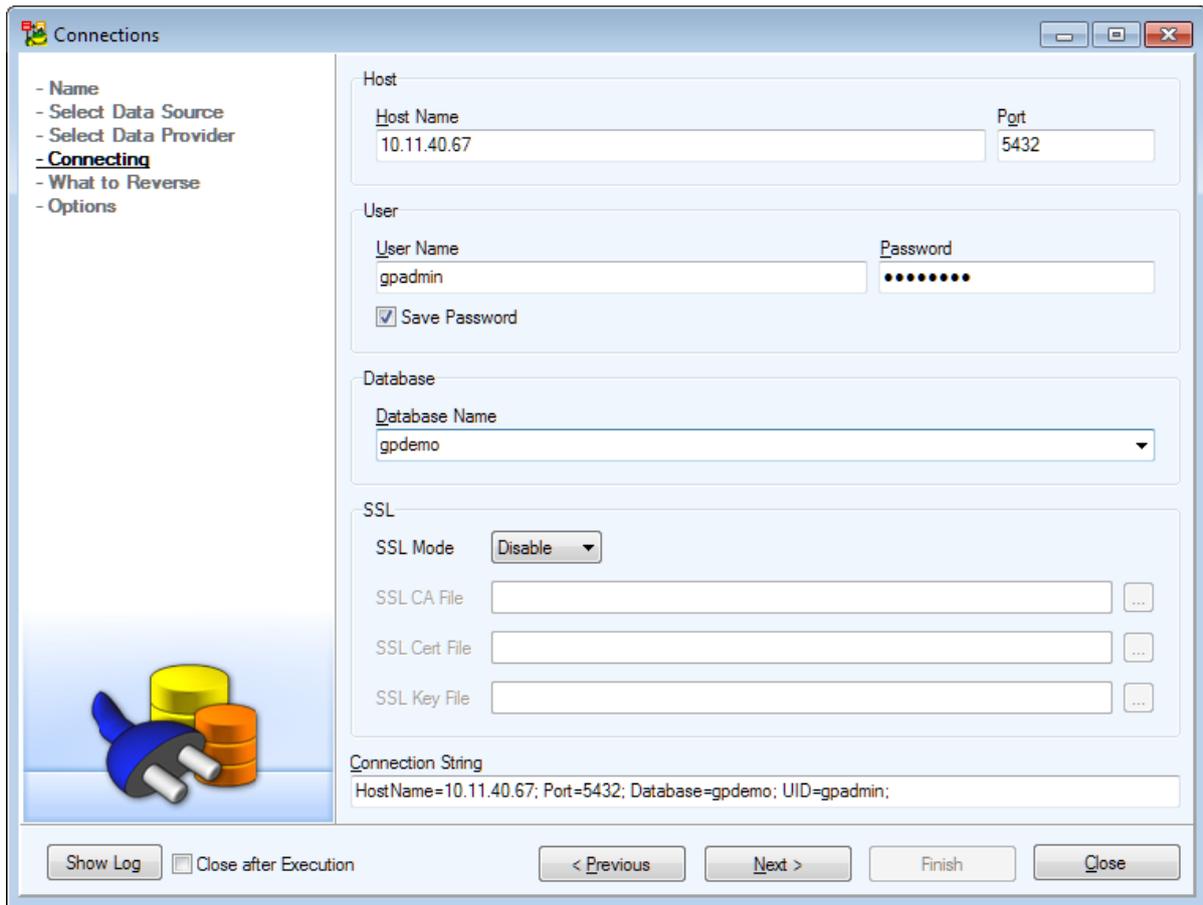
- Set External Table to be **Web** or **Regular** type External Table
  - Web** External Table - set **Execute OS Command**, **Segment Instances**, **Parameter of Segment Instances**
- Error Table Name - allows full table name, e.g. "schema1"."tablename"

## Reverse Engineering - Greenplum 4.1

Available **Data Providers** are:

- **Native Connection**

**Native Connection:**

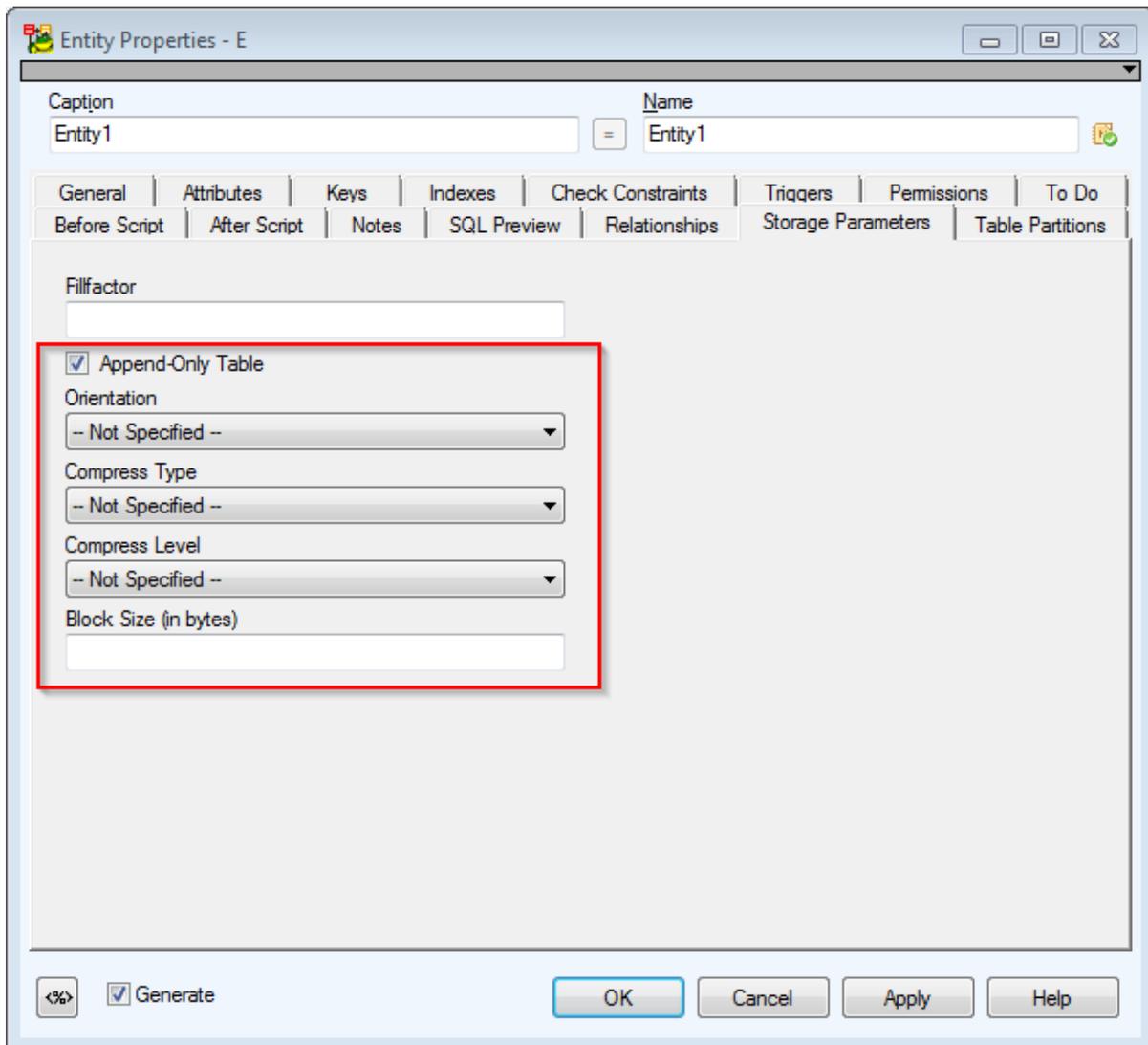


# Specifics - Greenplum 4.2

**Supported features** - SQL/DDDL script generation, reverse engineering, HTML/RTF/PDF reports, change scripting, verification.

**Permissions** can be set for entities, views and functions.

## Entity



**Storage Parameters** tab- Select the **Append-Only Table** checkbox to enable the **Block Size (in bytes)** box.

**Compress Type** combo box - new options added:

- ZLIB (default)
- QUICKLY
- RLE\_TYPE
- NONE

## Attribute

The screenshot shows the 'Attribute Properties - Entity1' dialog box. The 'Caption' and 'Name' fields are both set to 'Attribute 1'. The 'General' tab is active, showing the following settings:

- Data Type:** Bigint
- Domains:** (empty)
- Default Value:** (empty)
- Default Rule:** -- None --
- Primary Key:**
- Not Null:**
- Unique:**  (New AK)
- Comment:** (empty text area)
- Array Type:**
  - Is Array Type:**
  - Array Dimensions:** (empty)
- Column Encoding (highlighted in red):**
  - Compress Type:** z window from list
  - Compress Level:** -- Not Specified --
  - Block Size (in bytes):** (empty)

Buttons at the bottom include '<%', a refresh icon, 'OK', 'Ok+Add', 'Cancel', 'Apply', and 'Help'.

**Column Encoding** - only in Column Oriented tables, cannot be combined with table's compression parameters

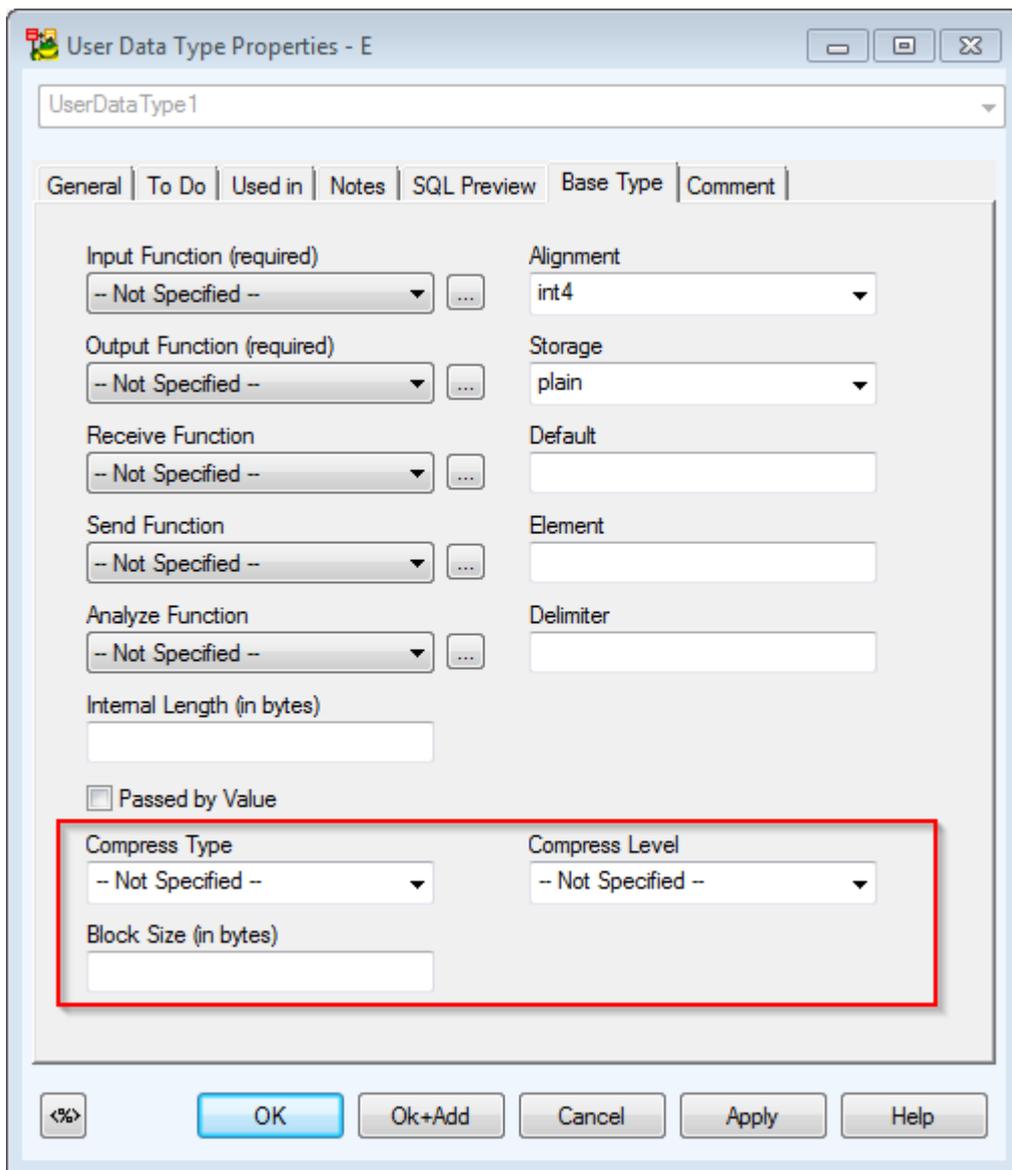
- Compress Type (ZLIB, QUICKLY, RLE\_TYPE, NONE)
- Compress Level (0 to 9)

- Block Size (in bytes)

## External Table

General tab, Format Typebox - new option *CUSTOM*.

## User Data Type



New options added on tab **Base Type**:

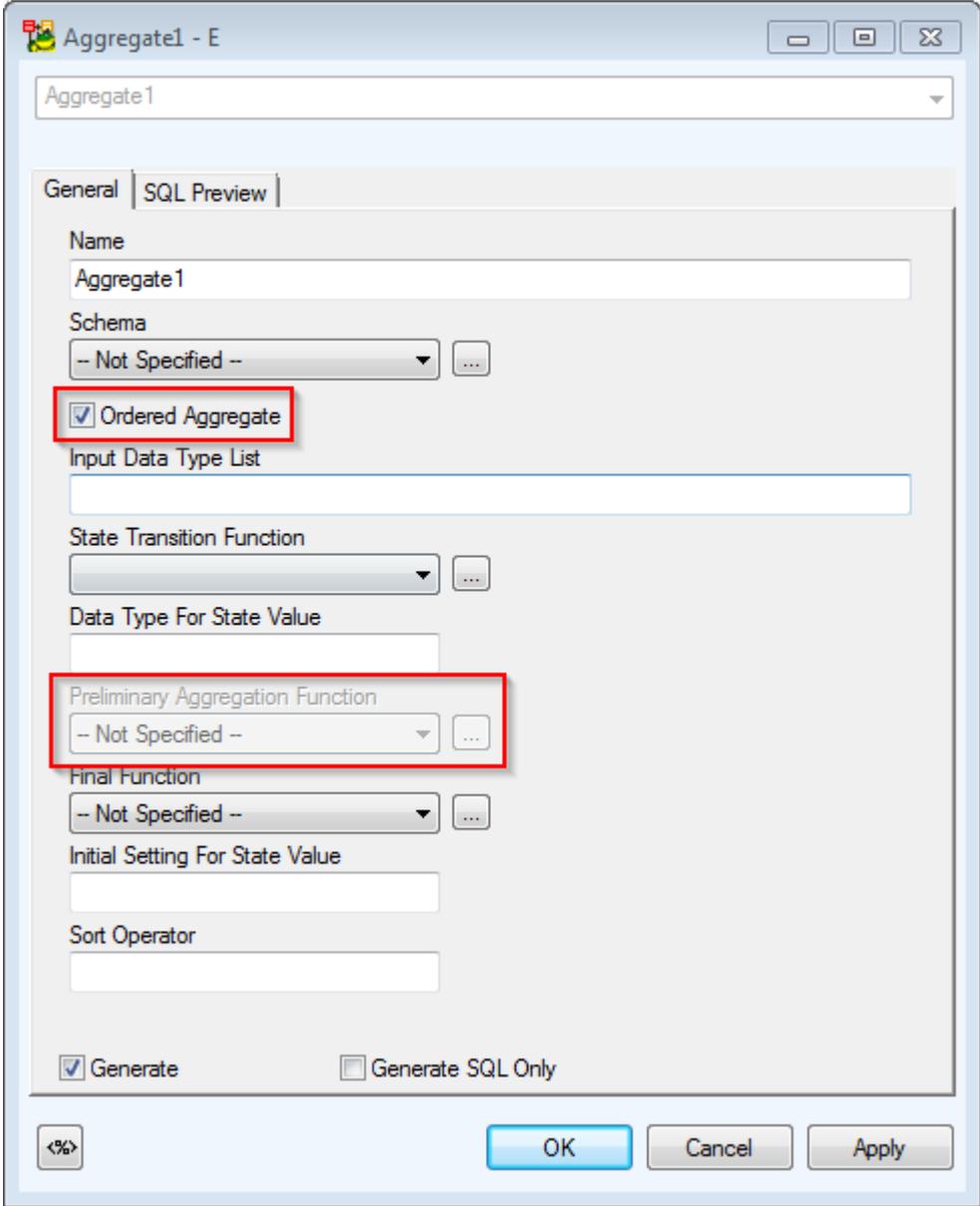
- **Compress Type** (ZLIB, QUICKLY, RLE\_TYPE, NONE options)
- **Compress Level** (0 to 9)
- **Block Size** (in bytes)

## Function

General pane - new **With Parameter (Describe Functions)** option, only for generating and reporting functions.

**Other objects:**

# Aggregate Function

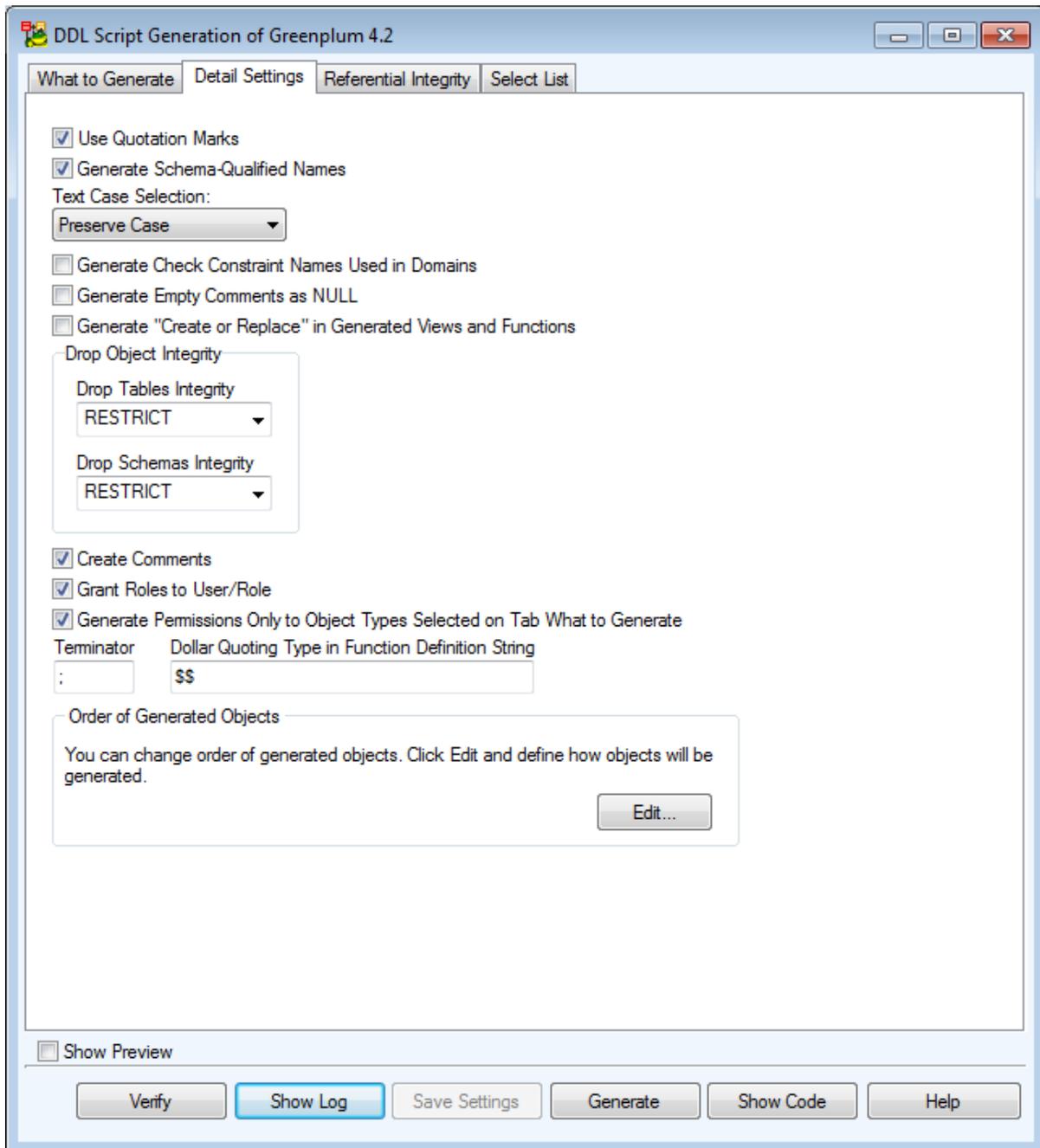


**Ordered Aggregate** option added. If selected, **Preliminary Aggregation Function** is disabled and when applied (OK/Apply), the combo box is set to value *-- Not Specified --*.

## Reverse Engineering - Greenplum 4.2

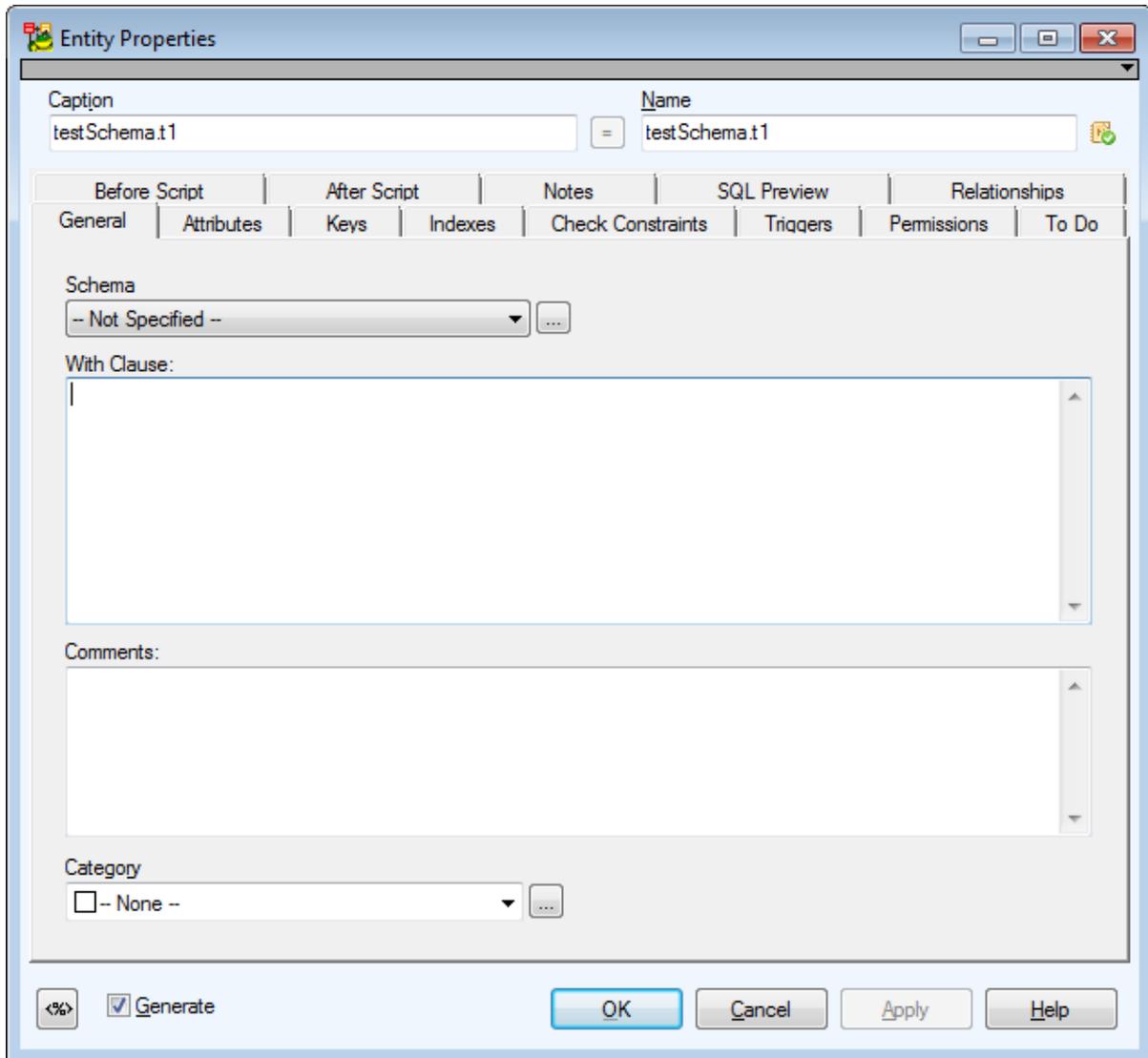
See [Reverse Engineering - Greenplum 4.1](#) for more information.

# Script Generation - Greenplum 4.2

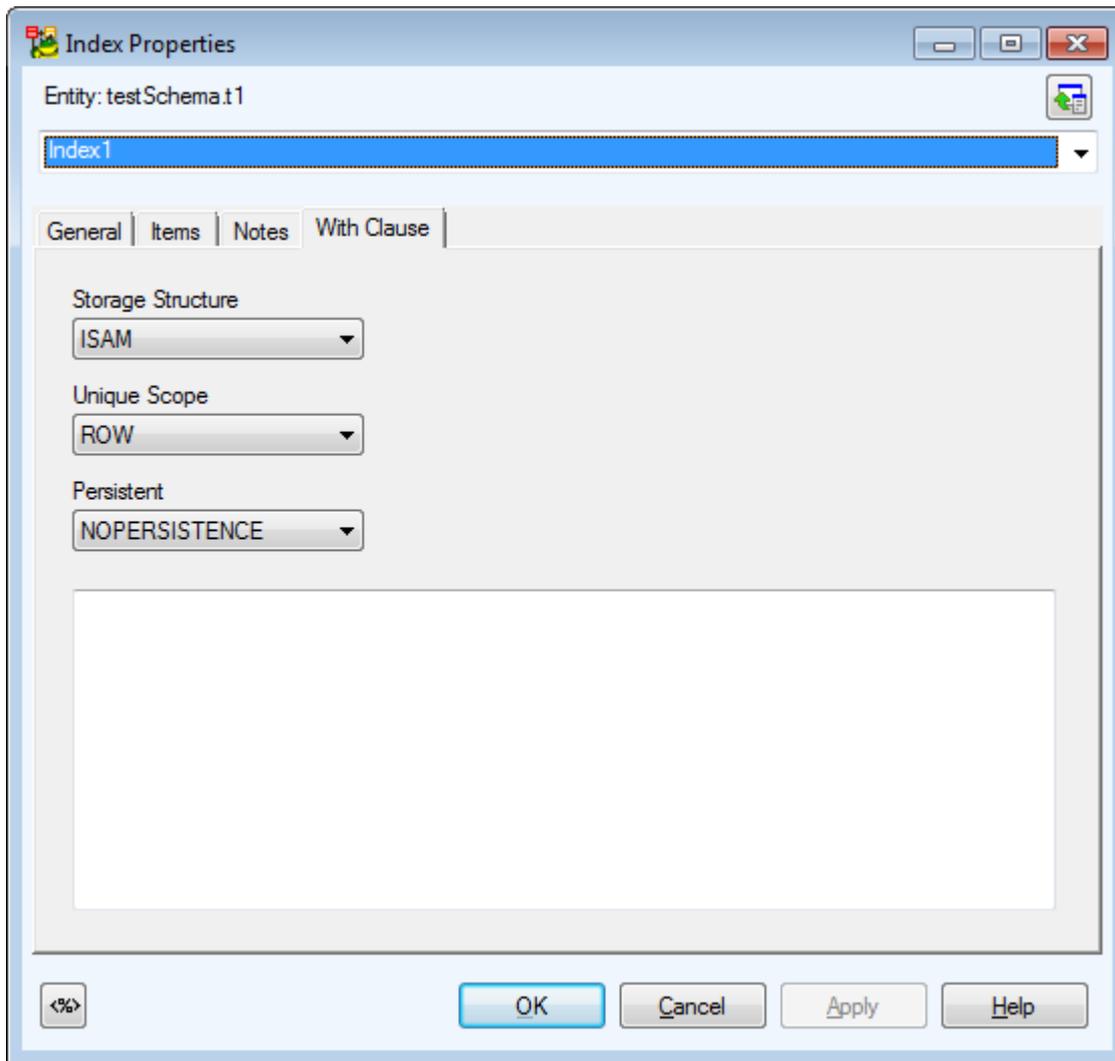


# Specifics - Ingres 9.3

## Entity

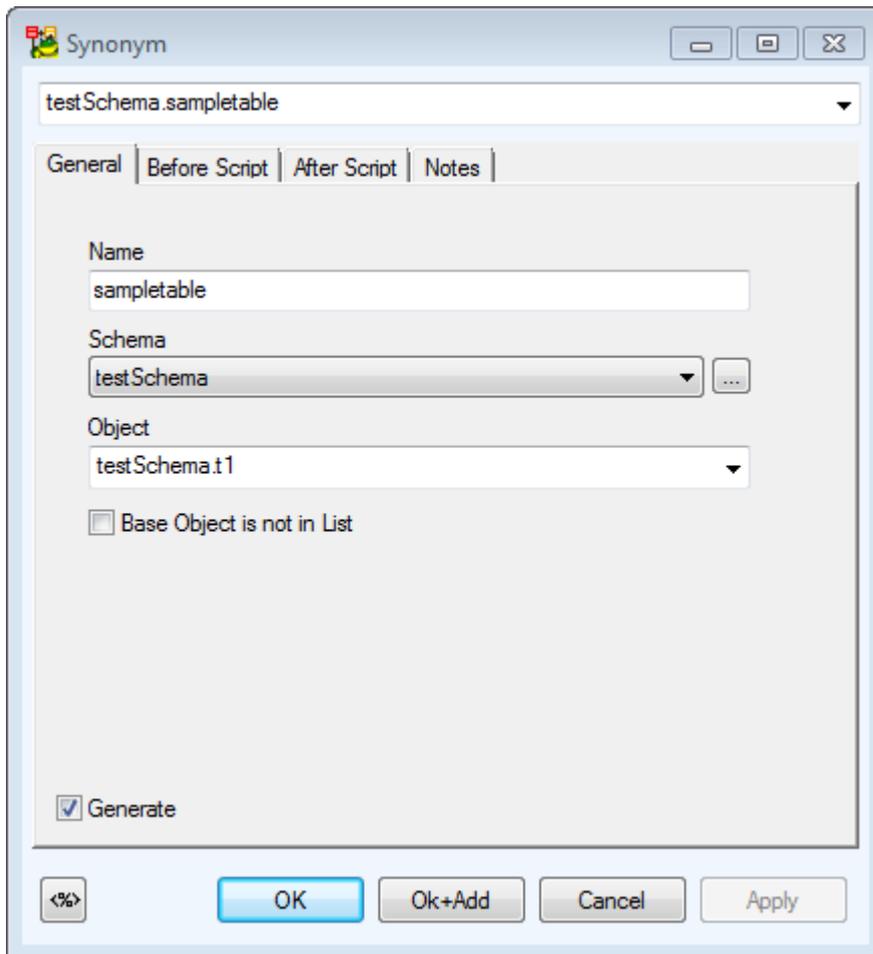


# Index

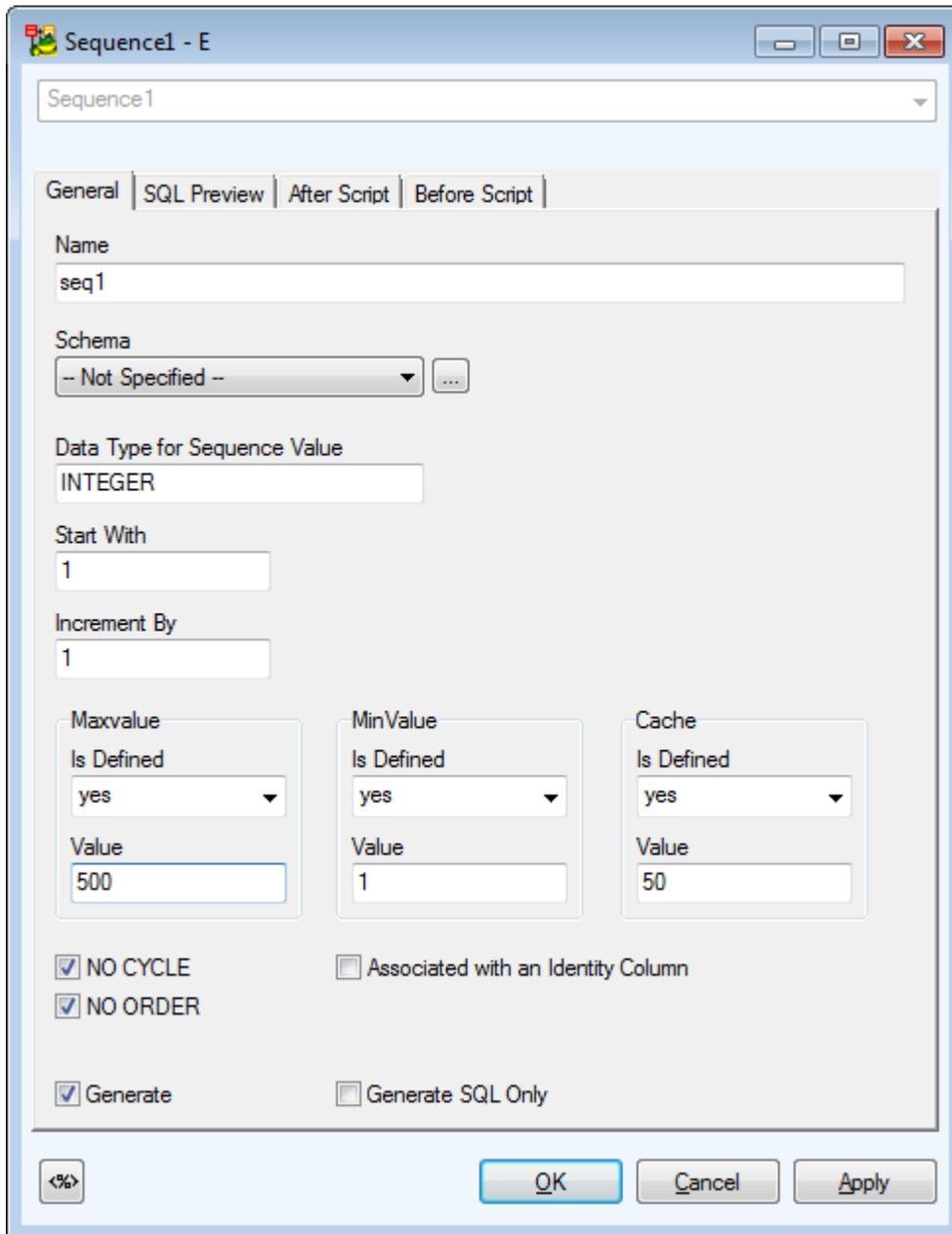


**Entity** and **Index** dialogs contain **With Clause**.

# Synonyms



Other objects:

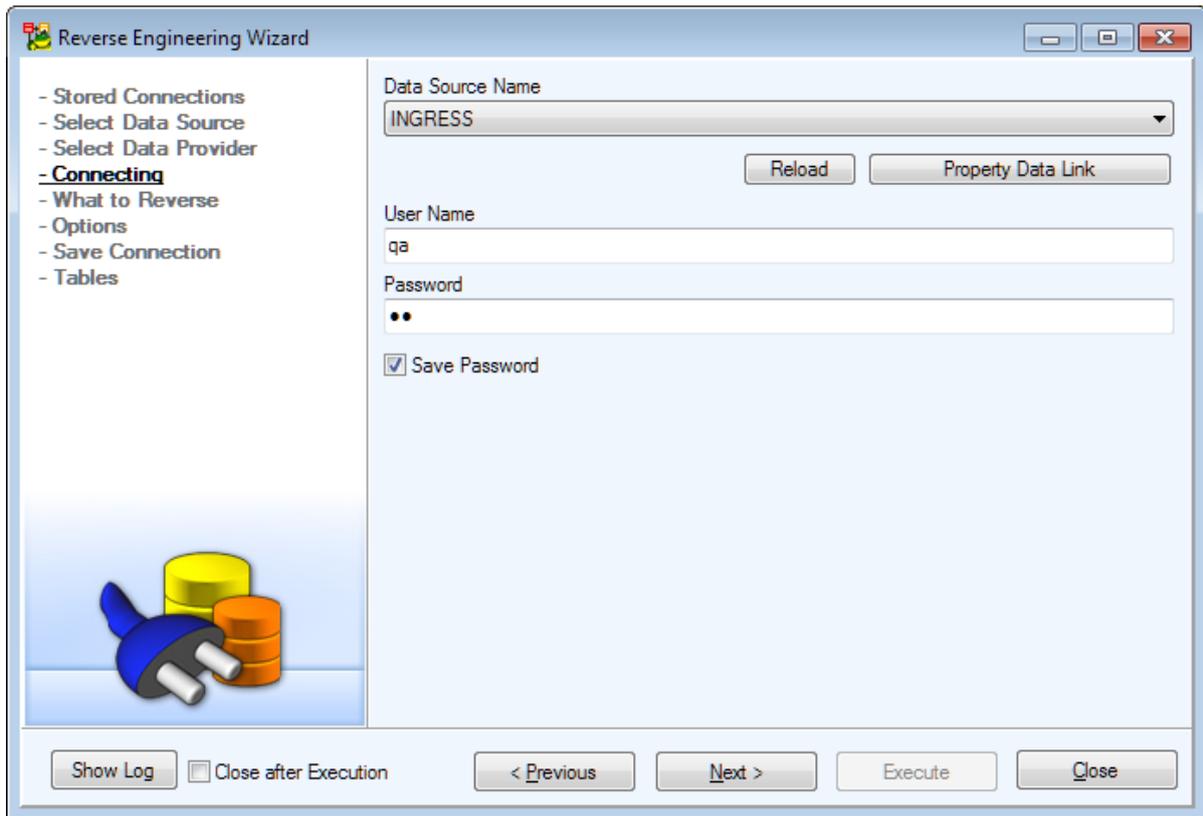


## Reverse Engineering - Ingres 9.3

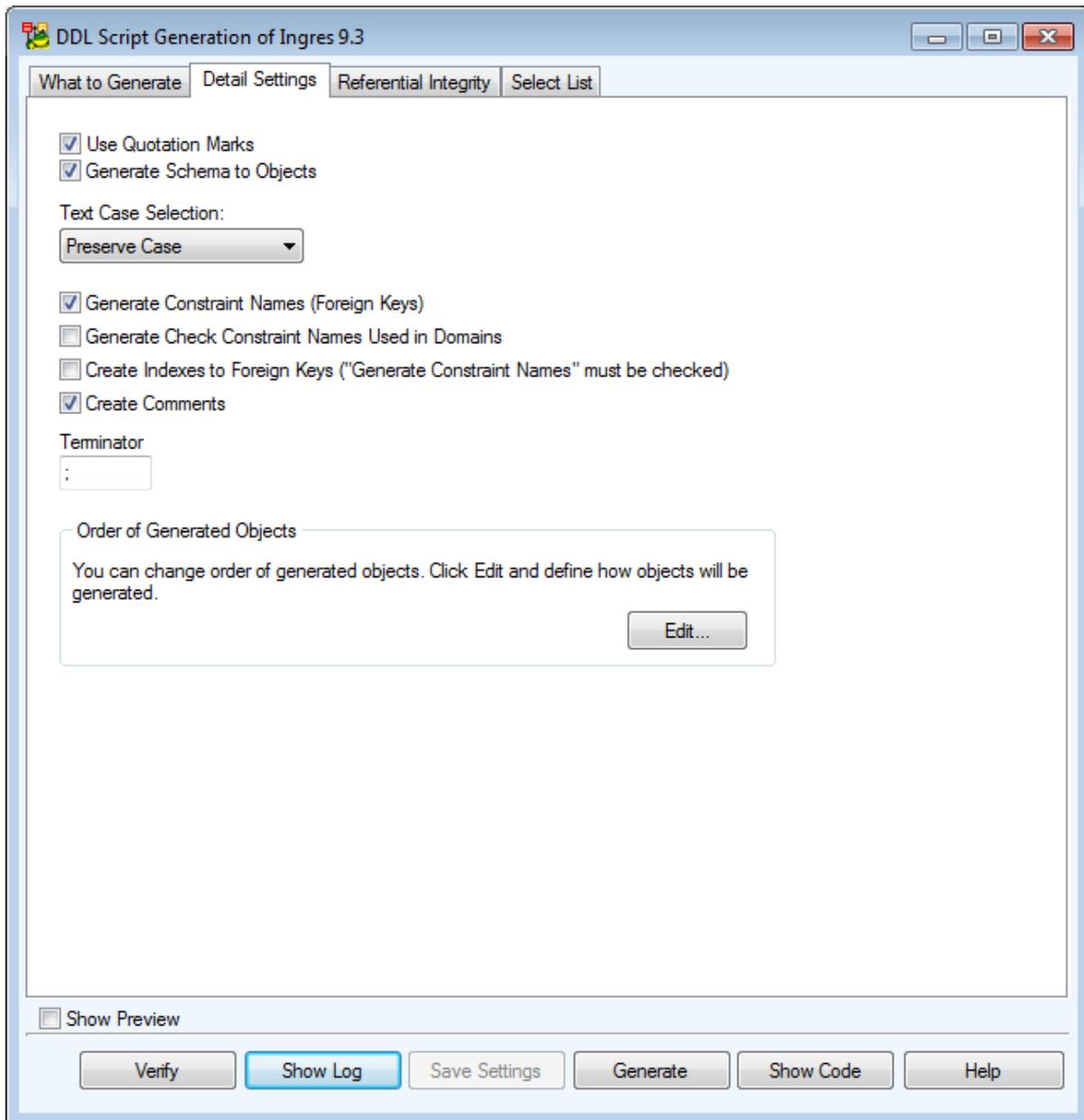
Available **Data Providers** are:

- **Connection via ODBC**

**Connection via ODBC:**



# Script Generation - Ingres 9.3



# Specifics - Ingres 10.0

See [Specifics - Ingres 9.3](#) for more information.

# Reverse Engineering - Ingres 10.0

See [Reverse Engineering - Ingres 9.3](#) for more information.

# Specifics - EDB Postgres Advanced Server 10

EDB Advanced Server is closely based on PostgreSQL 10.

EDB 10 also offers support for packages and synonyms. Packages and synonyms are supported in Change Script Generation, Reverse Engineering and in Reports.

Change Script Generation is based on PostgreSQL 10 and differences between EDB and Postgres might not be generated correctly.

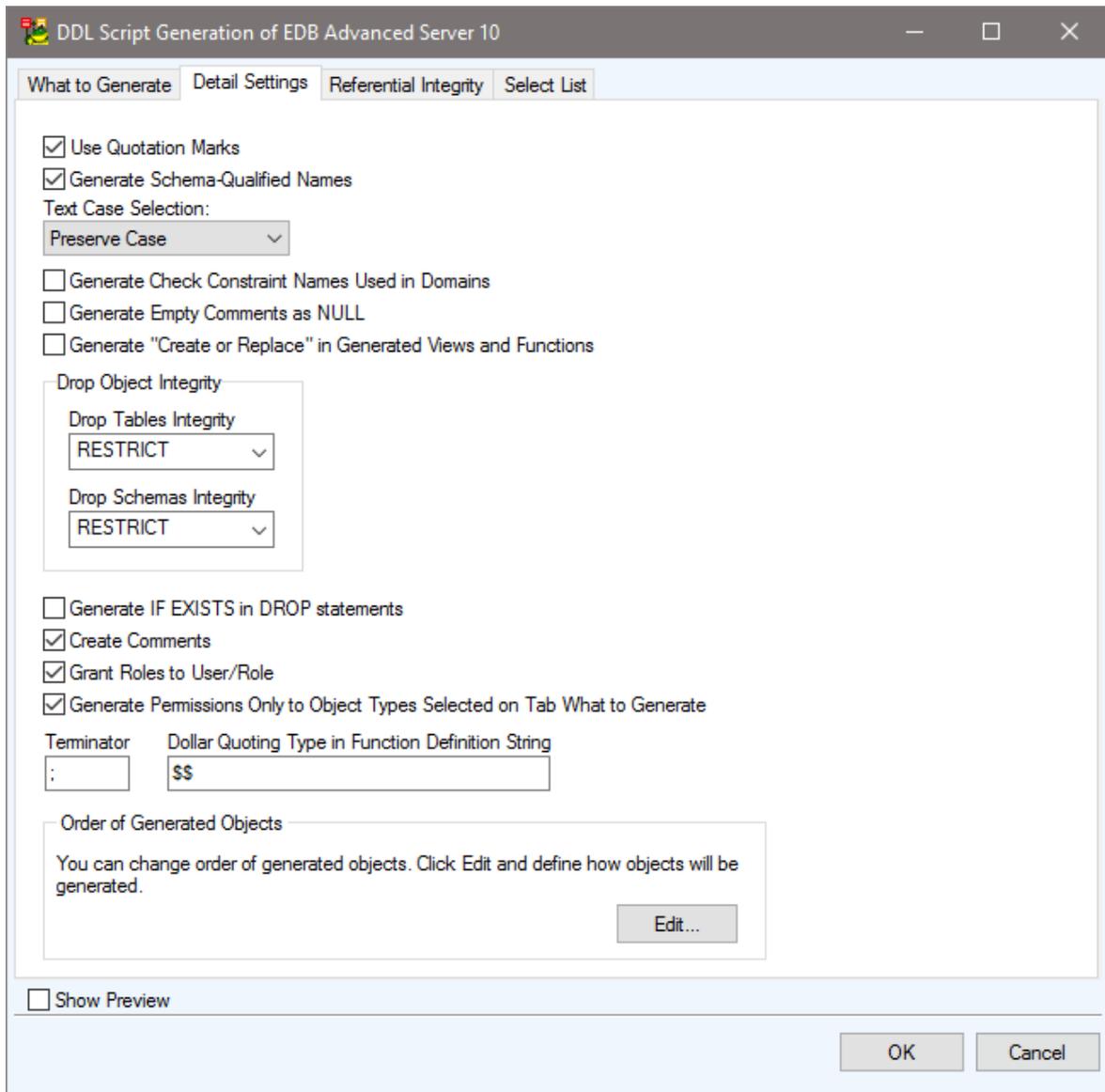
## Stored procedures

COST in stored procedures is rounded to 3 decimals during Reverse Engineering.

SET in stored procedures is loaded into AfterScript. Default value is 100.

By default procedures will execute with the privileges of the user that created them (SECURITY DEFINER). By default VOLATILE is set for procedures. By default procedures cannot be executed in parallel mode (PARALLEL UNSAFE). Default values are not generated in TDM.

# Script Generation - EDB Advanced Server 10



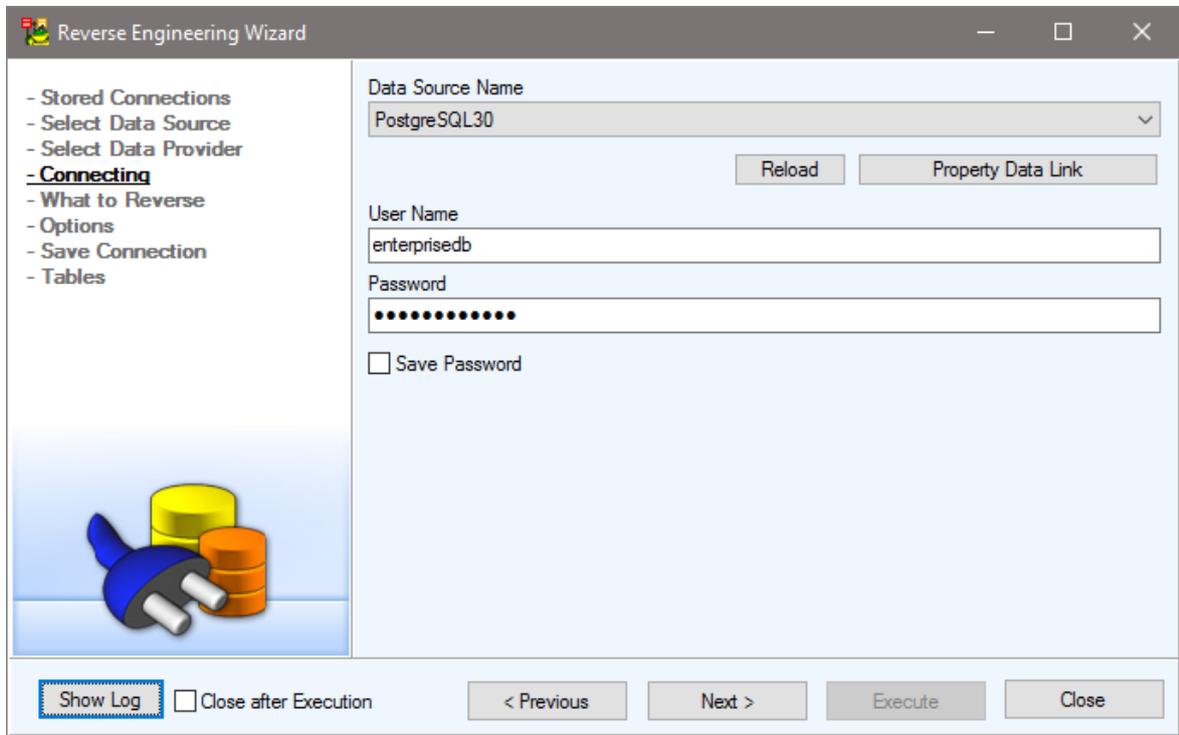
## Reverse Engineering - EDB Advanced Server

Available **Data Providers** are:

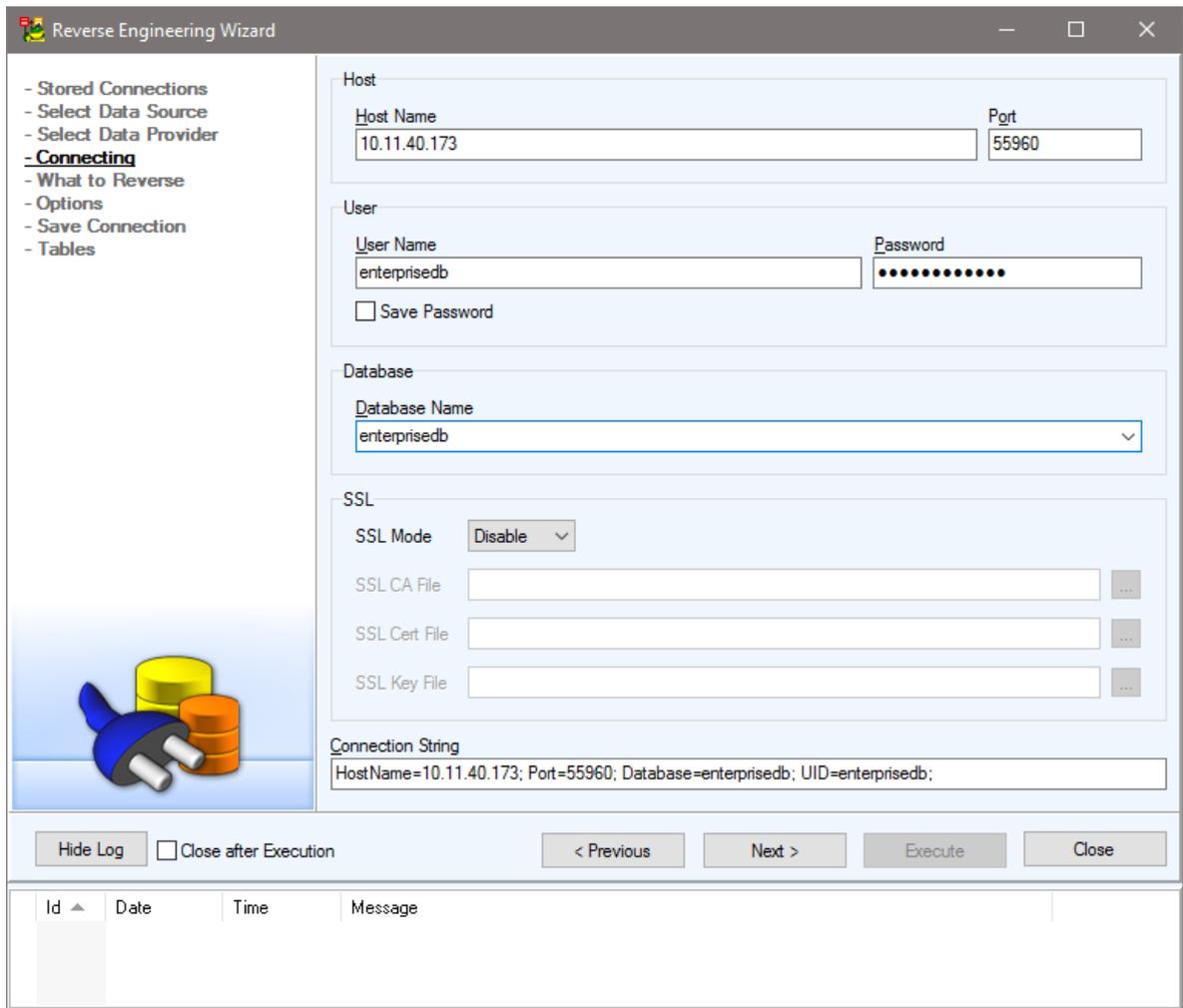
- **Connection via ODBC**
- **Native Connection**

**Connection via ODBC:**

1. Install the correct ODBC driver for your database
2. In **Reverse Engineering Wizard | Connecting** select **Property Data Link**
3. Click **Add** and select the driver
4. Enter your connection details, click **Test** and **Save**
5. Click Reload to refresh the list of data sources and select your desired ODBC connection

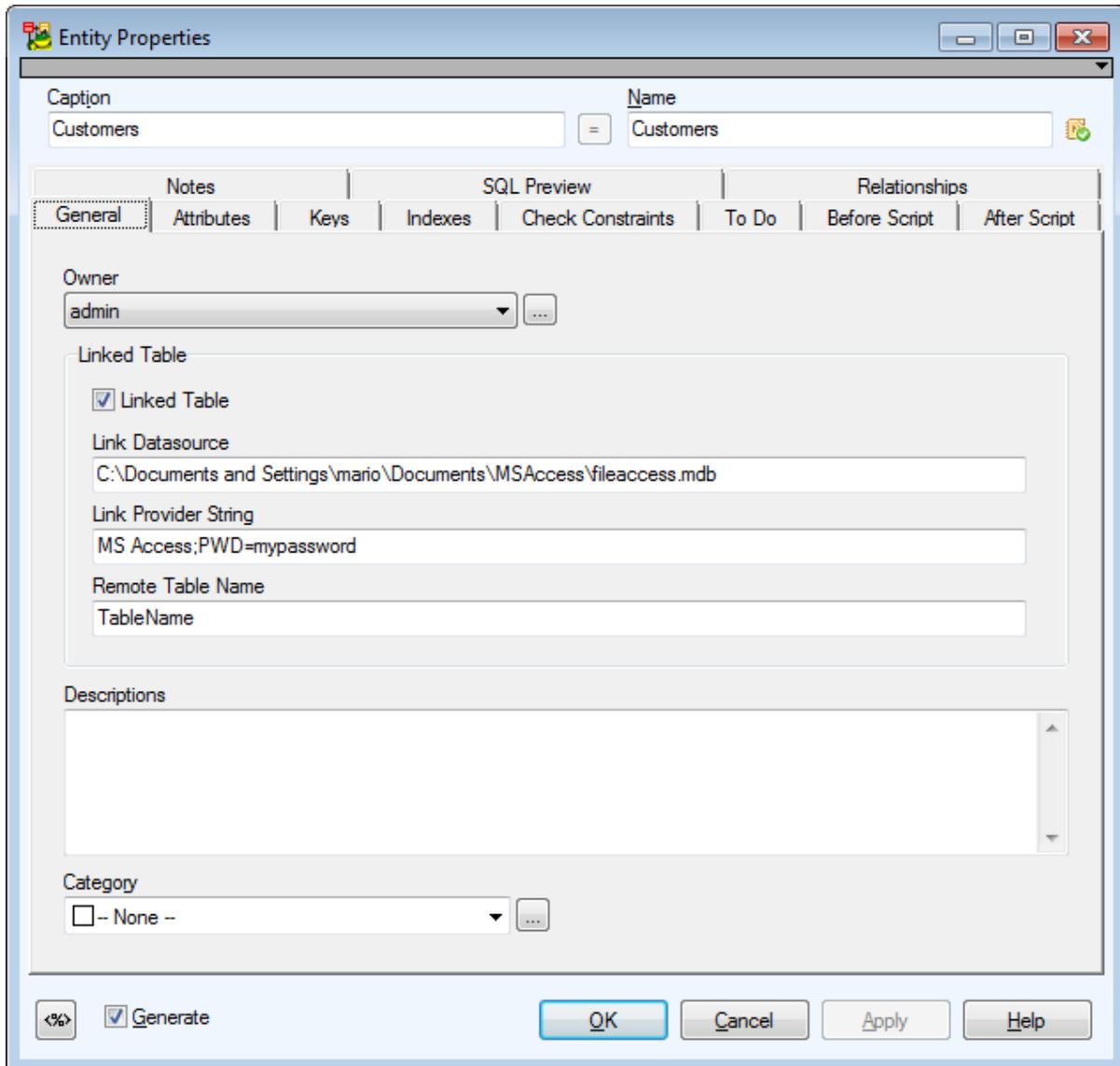


**Native Connection:**



# Specifics - Microsoft Access 2010

## Entity



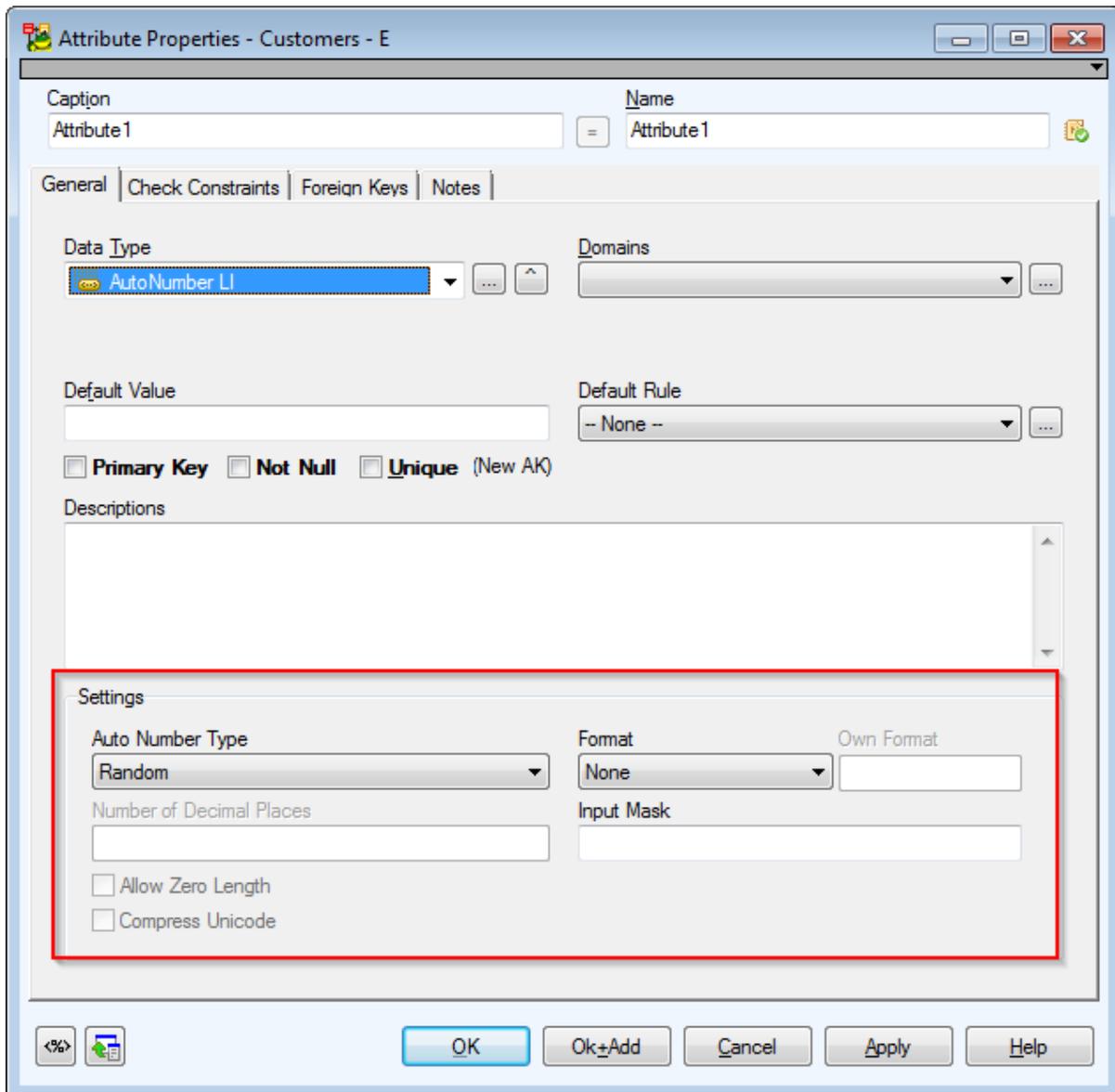
Select the **Linked Table** checkbox and fill out the following boxes (name: *filled\_value*):

Link Datasource: C:\Documents and Settings\mario\Documents\MSAccess\fileaccess.mdb

Link Provider String: *Microsoft Access;PWD=mypassword*

Remote Table Name: *TableName*

# Attribute



Select the AutoNumber LI data type to display the **Auto Number Type** box. From this box, you can select **Increment** or **Random**.

Select the *Byte*, *Currency*, *Decimal*, *Double*, *Integer*, *Long Integer*, *Single* data type to display the **Number of Decimal Places** box.

Select the *Hyperlink*, *Memo*, *Text* data type to display the **Allow Zero Length** and **Compress Unicode** checkboxes.

## User Data Types in the Model menu

Microsoft Access database does not have user data types. Nevertheless, these user data types in Toad Data Modeler work only as an additional structure that can be used for a data type that Microsoft Access database supports but that is not available in Toad Data Modeler.

### Generation of names in brackets

- Check **Delimited Identifiers** in **DDL Script Generation | Detail Settings** to generate all names in brackets
- If unchecked, names will be generated without brackets with exception of names that include spaces

### Generation of captions

- Check **Generate Captions** in **DDL Script Generation | Detail Settings** for MS Access models

# Specifics - Microsoft Access 2010

## Entity

The screenshot shows the 'Entity Properties' dialog box with the following fields and values:

- Caption:** Customers
- Name:** Customers
- Owner:** admin
- Linked Table:**  Linked Table
- Link Datasource:** C:\Documents and Settings\mario\Documents\MSAccess\fileaccess.mdb
- Link Provider String:** MS Access;PWD=mypassword
- Remote Table Name:** TableName
- Category:** -- None --

Buttons at the bottom: <%, Generate, OK, Cancel, Apply, Help.

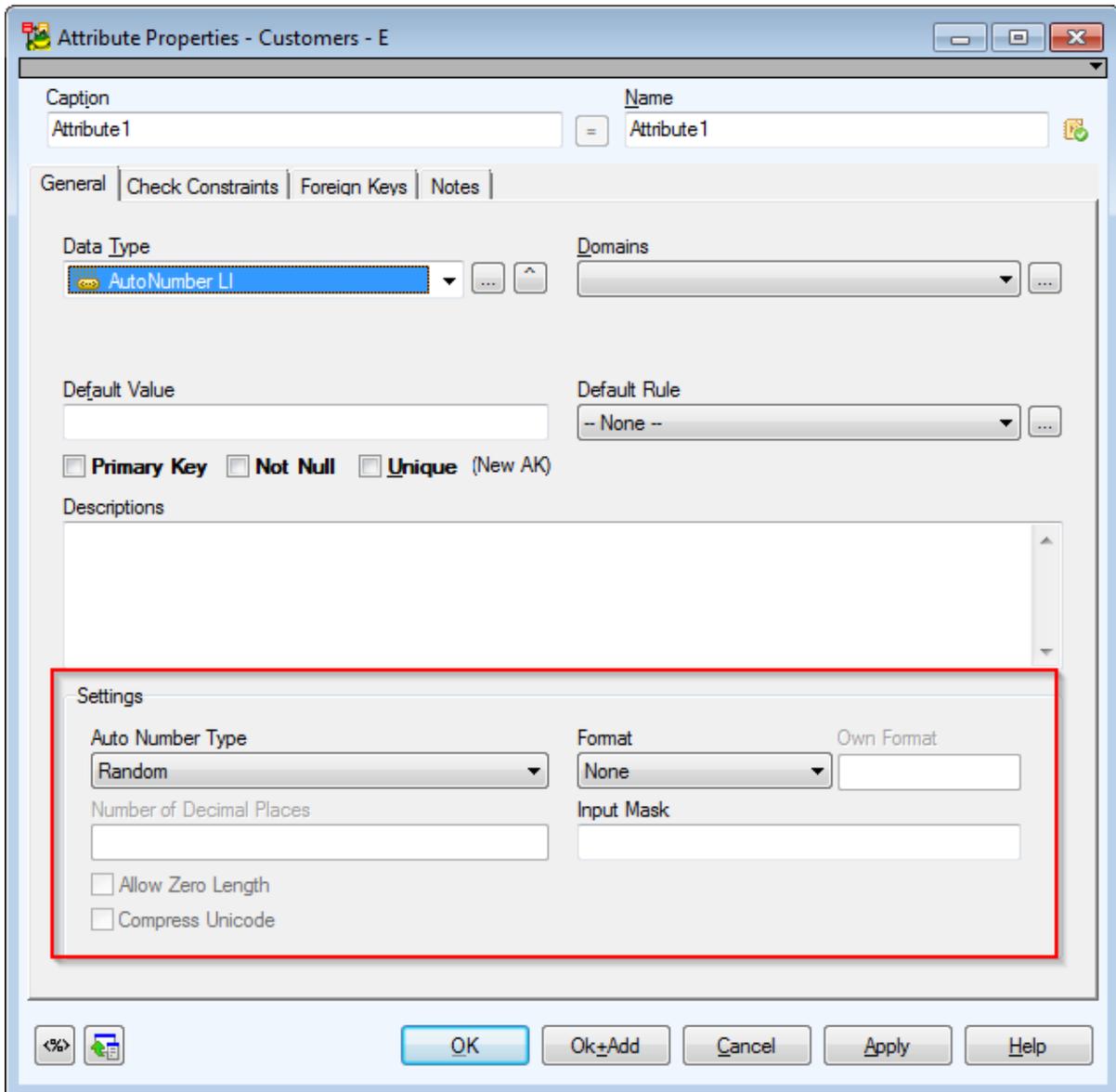
Select the **Linked Table** checkbox and fill out the following boxes (name: *filled\_value*):

Link Datasource: C:\Documents and Settings\mario\Documents\MSAccess\fileaccess.mdb

Link Provider String: *Microsoft Access;PWD=mypassword*

Remote Table Name: *TableName*

# Attribute



Select the AutoNumber LI data type to display the **Auto Number Type** box. From this box, you can select **Increment** or **Random**.

Select the *Byte*, *Currency*, *Decimal*, *Double*, *Integer*, *Long Integer*, *Single* data type to display the **Number of Decimal Places** box.

Select the *Hyperlink*, *Memo*, *Text* data type to display the **Allow Zero Length** and **Compress Unicode** checkboxes.

## User Data Types in the Model menu

Microsoft Access database does not have user data types. Nevertheless, these user data types in Toad Data Modeler work only as an additional structure that can be used for a data type that Microsoft Access database

supports but that is not available in Toad Data Modeler.

### Generation of names in brackets

- Check **Delimited Identifiers** in **DDL Script Generation | Detail Settings** to generate all names in brackets
- If unchecked, names will be generated without brackets with exception of names that include spaces

### Generation of captions

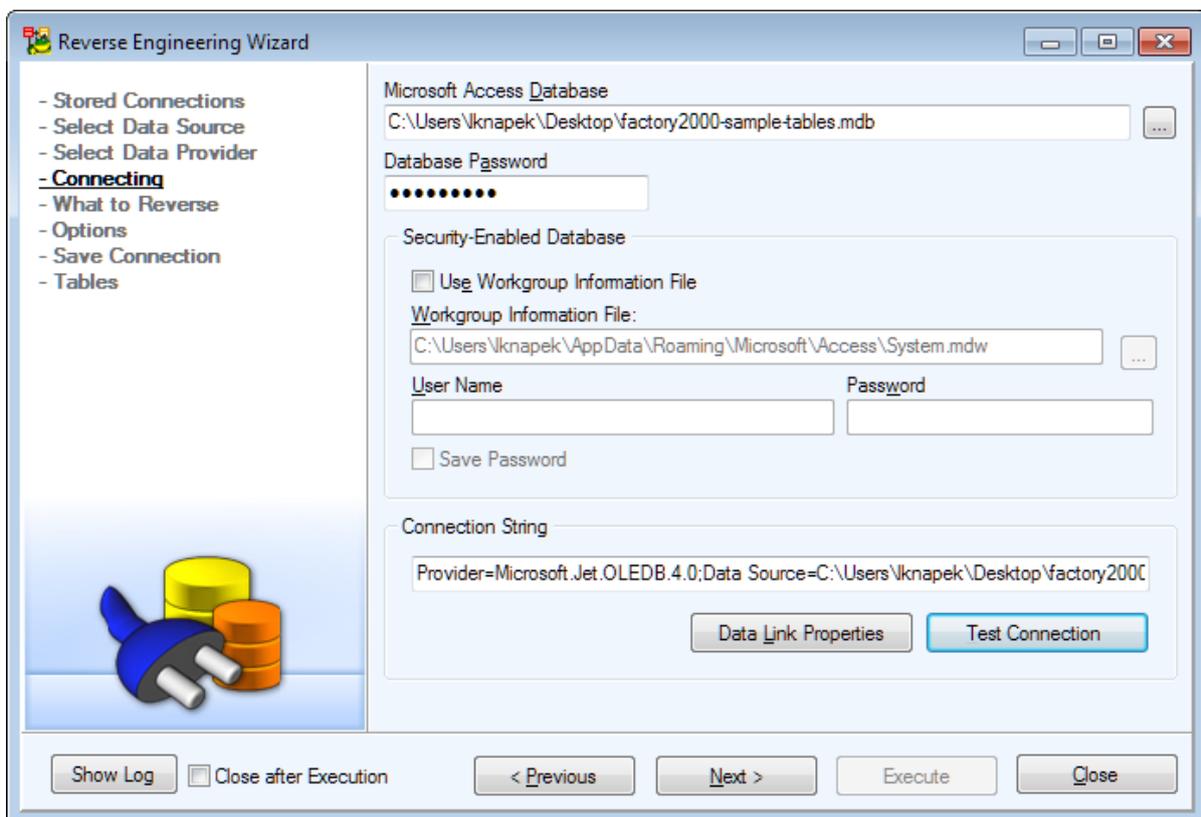
- Check **Generate Captions** in **DDL Script Generation | Detail Settings** for MS Access models

## Reverse Engineering - Microsoft Access 2007/2010

Available **Data Providers** are:

- **Connection via ADO and DAO**

**Connection via ADO and DAO:**



## Script Generation - Microsoft Access 2007/2010

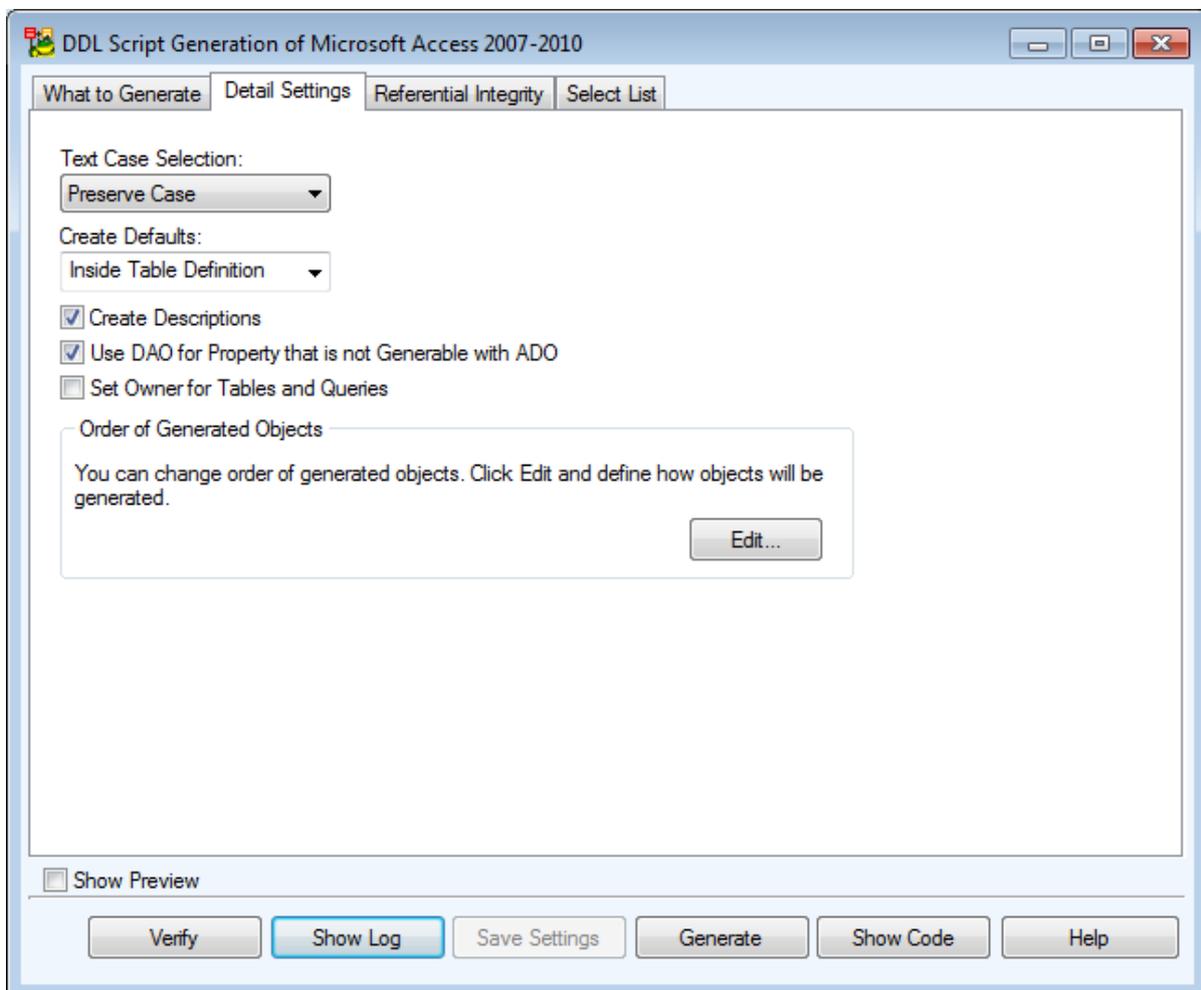
Toad Data Modeler doesn't generate an SQL script for the Microsoft Access database, but does generate a fully functional source code in the VBA language. The source code can be executed directly in the Microsoft Access as

Module. Appropriate libraries in Microsoft Access are necessary - "Microsoft ActiveX Data Objects 2.x Library" and "Microsoft ADO Ext. 2.x for DDL and Security" for ADO, or "Microsoft DAO 3.6 Object Library" for DAO.

To select a suitable library, click the **Tools menu | References** (in Microsoft Access main menu).

In the beginning of every script generated for Microsoft Access, you will see the following order:

```
'=====
'== Microsoft Access 2000/2002/2003 database creation method
'=====
'== 1. Create a new database in the Microsoft Access
'== 2. Create a new module
'== 3. Copy the TDM3 output SQL script into the new Microsoft Access module
'== 4. Select from main menu "Tools" item "References..." and check
'== the "Microsoft ActiveX Data Objects 2.x Library"
'== and "Microsoft ADO Ext. 2.x for DDL and Security"
'== and "Microsoft DAO 3.6 Object Library"
'== 5. Place your mouse cursor somewhere in the main procedure Main()
'== 6. Run the module code (Click the "Run Sub/UserForm" button or press F5)
'=====
```



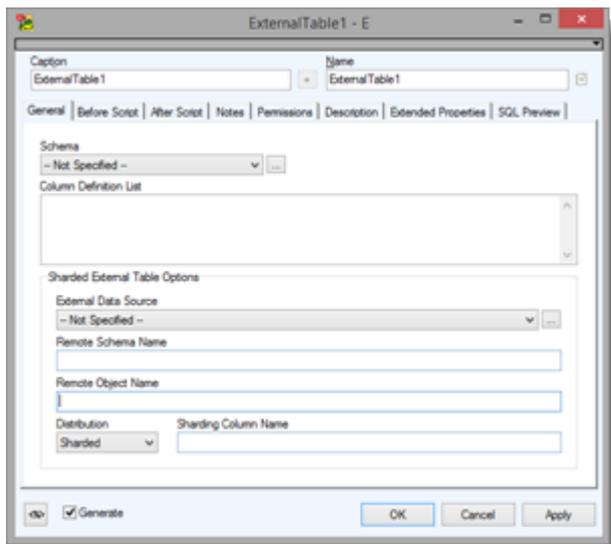
# Specifics - Microsoft Azure SQL Database V12

## Specifics of Toad Data Modeler support

- It is not possible to specify a Filegroup in **Entity Properties | General | Data Space**. When you enable Partition Scheme you can select from available schemes



- Definitions of User and Schema objects are not loaded
- There are differences between Azure SQL Database and SQL Server 2016 in External Table syntax
- Right-click External Tables in the **Physical Model Explorer** and select Add to create external tables.



**NOTE:** Rule, Default - Settings of Attributes are loaded into AfterScript during Reverse Engineering

## New Objects in Azure SQL Database V12

- Partition Scheme and Partition Function
- Fulltext Catalog, Fulltext Index and Fulltext Stoplist

- Extended Property
- XML Schema Collection
- Primary, Secondary, Selective and Secondary Selective XML Index
- Spatial Index, Columnstore Index
- Sequence
- Column Encryption Key
- External Table
- External Data Source
- Security Policy

### **SQL Server unsupported features in Azure SQL Database**

These objects are not supported by Microsoft Azure in comparison with Microsoft SQL Server:

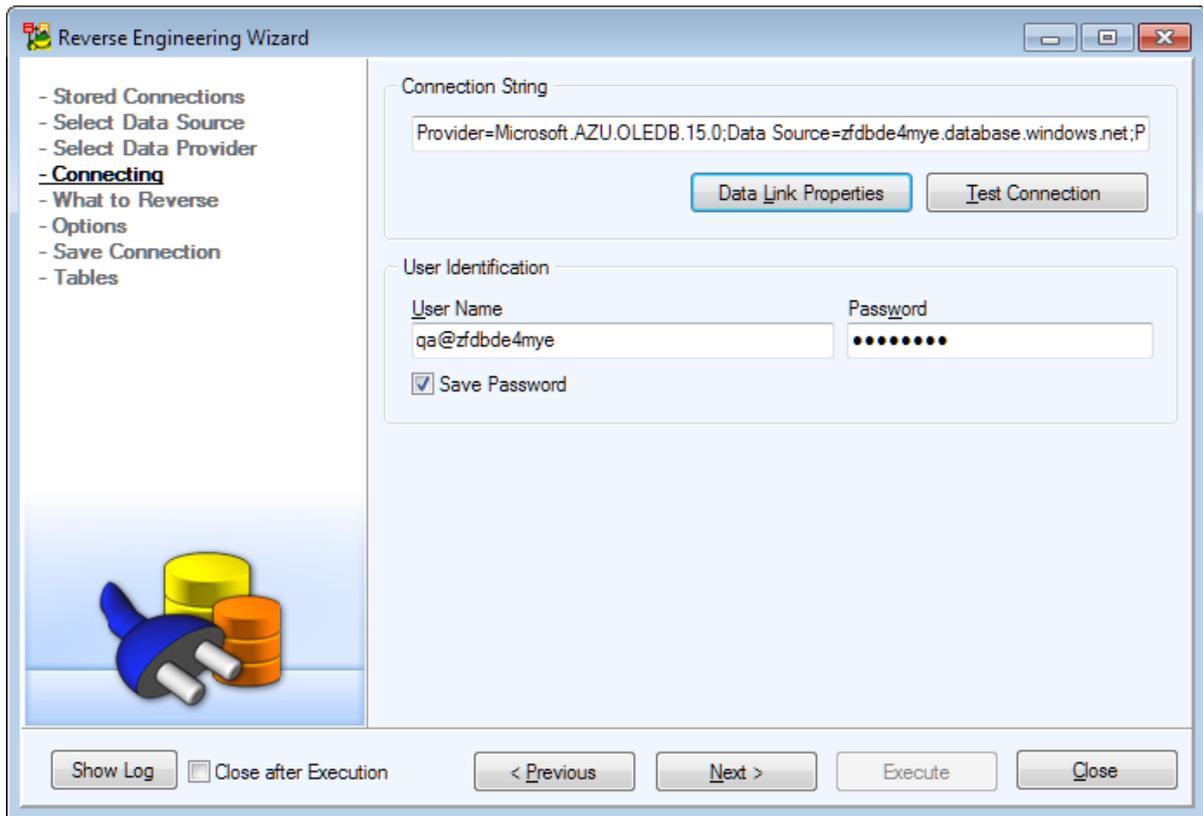
- Assembly
- CLR features such as (CLR procedures, functions, triggers, user-defined type, ORDER property in functions)
- Aggregate Function
- Extended Stored Procedure
- Filestream and its settings
- Semantic search (STATISTICAL\_SEMANTICS in CREATE FULLTEXT INDEX)
- FileTable
- Search Property List and its application in Fulltext Index
- STATISTICS\_INCREMENTAL property of keys and indexes
- External File Format
- Stretch database (REMOTE\_DATA\_ARCHIVE v CREATE TABLE)

## **Reverse Engineering - Microsoft Azure SQL Database V12**

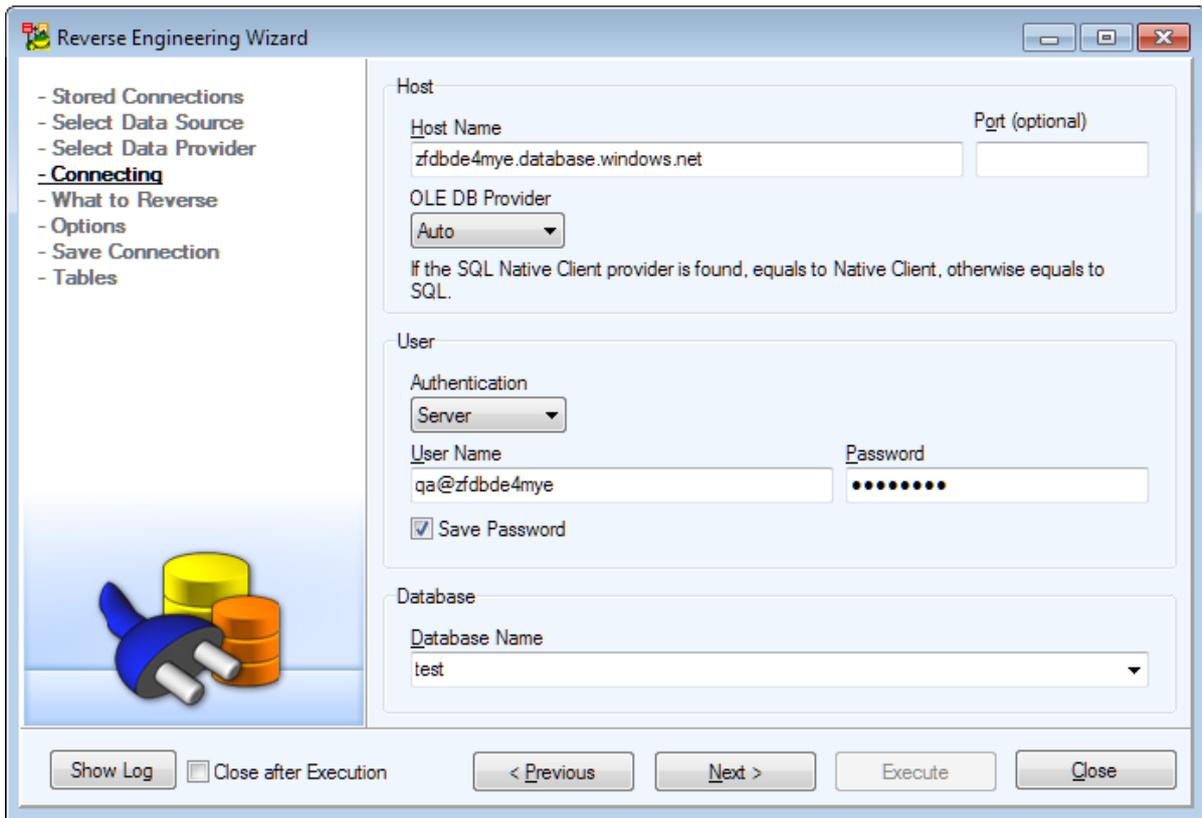
Available **Data Providers** are:

- **Connection via ADO**
- **Native Connection**

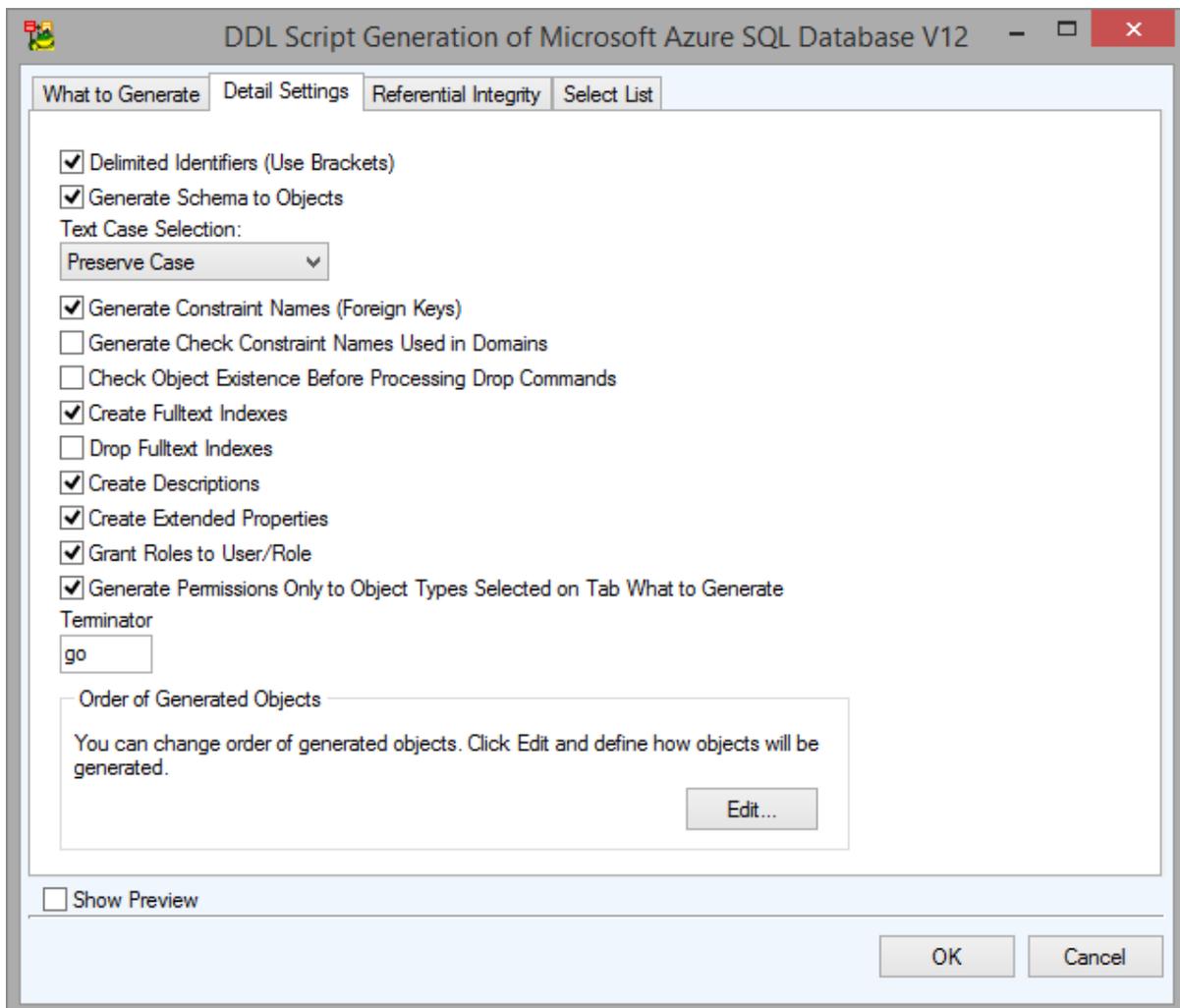
**Connection via ADO:**



**Native Connection:**



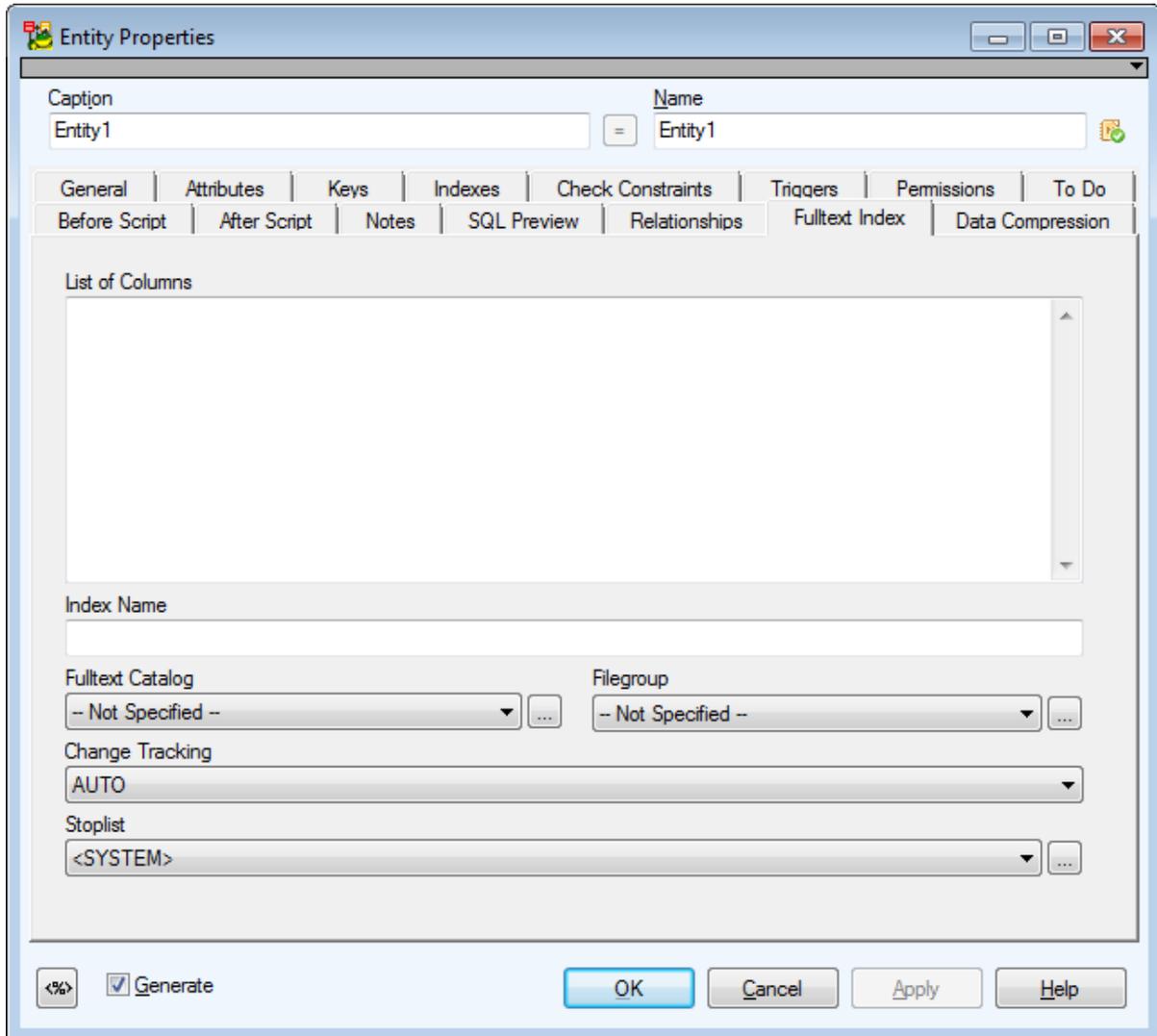
# Script Generation - Microsoft Azure SQL Database V12



# Specifics - Microsoft SQL Server 2012

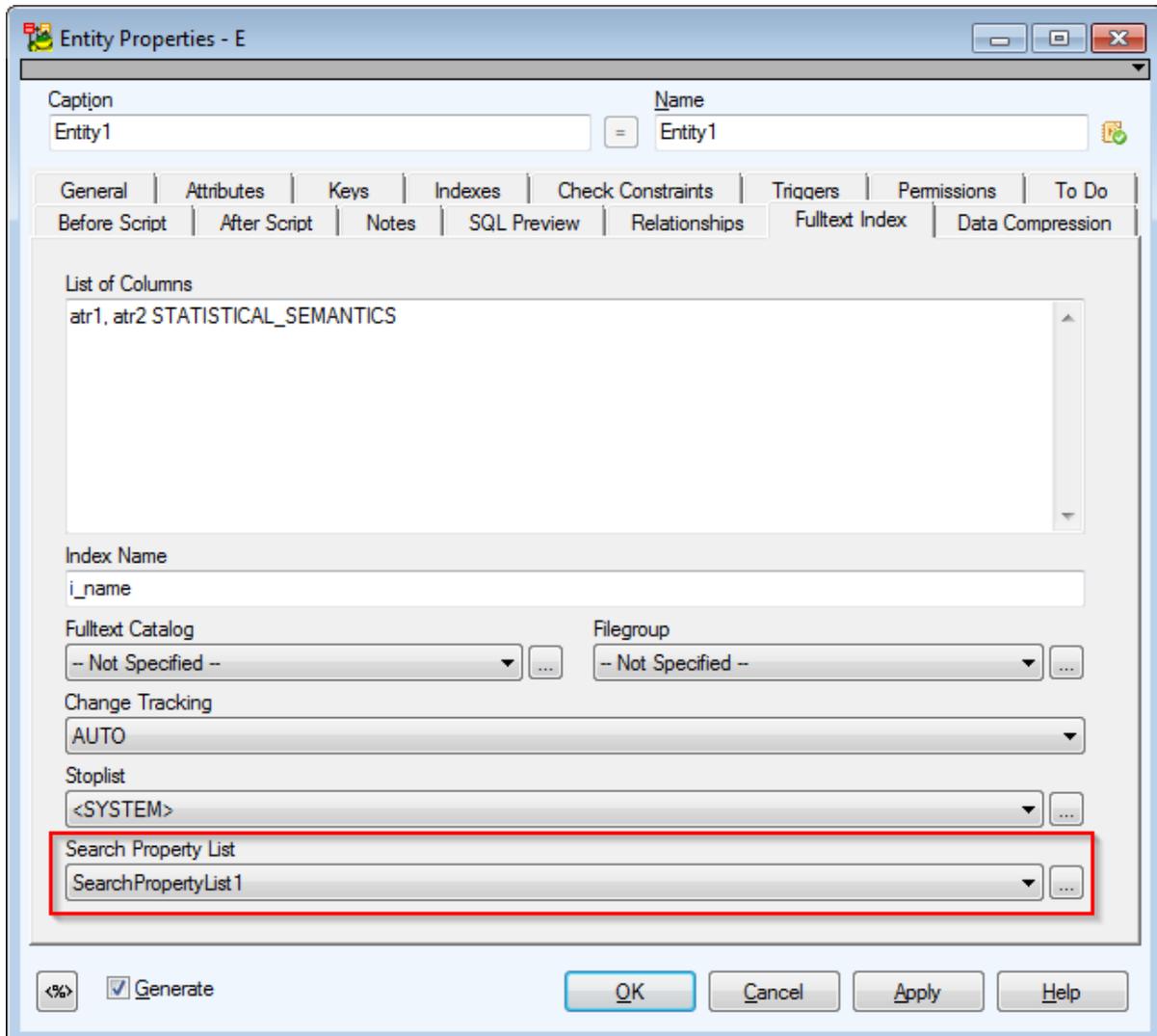
## Entity

### Fulltext Index tab

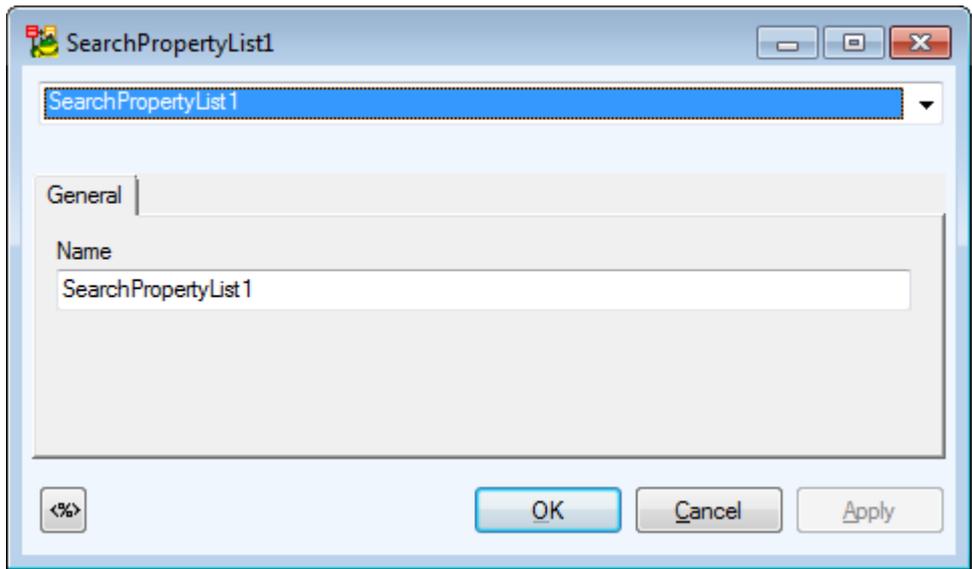


# Search Property List

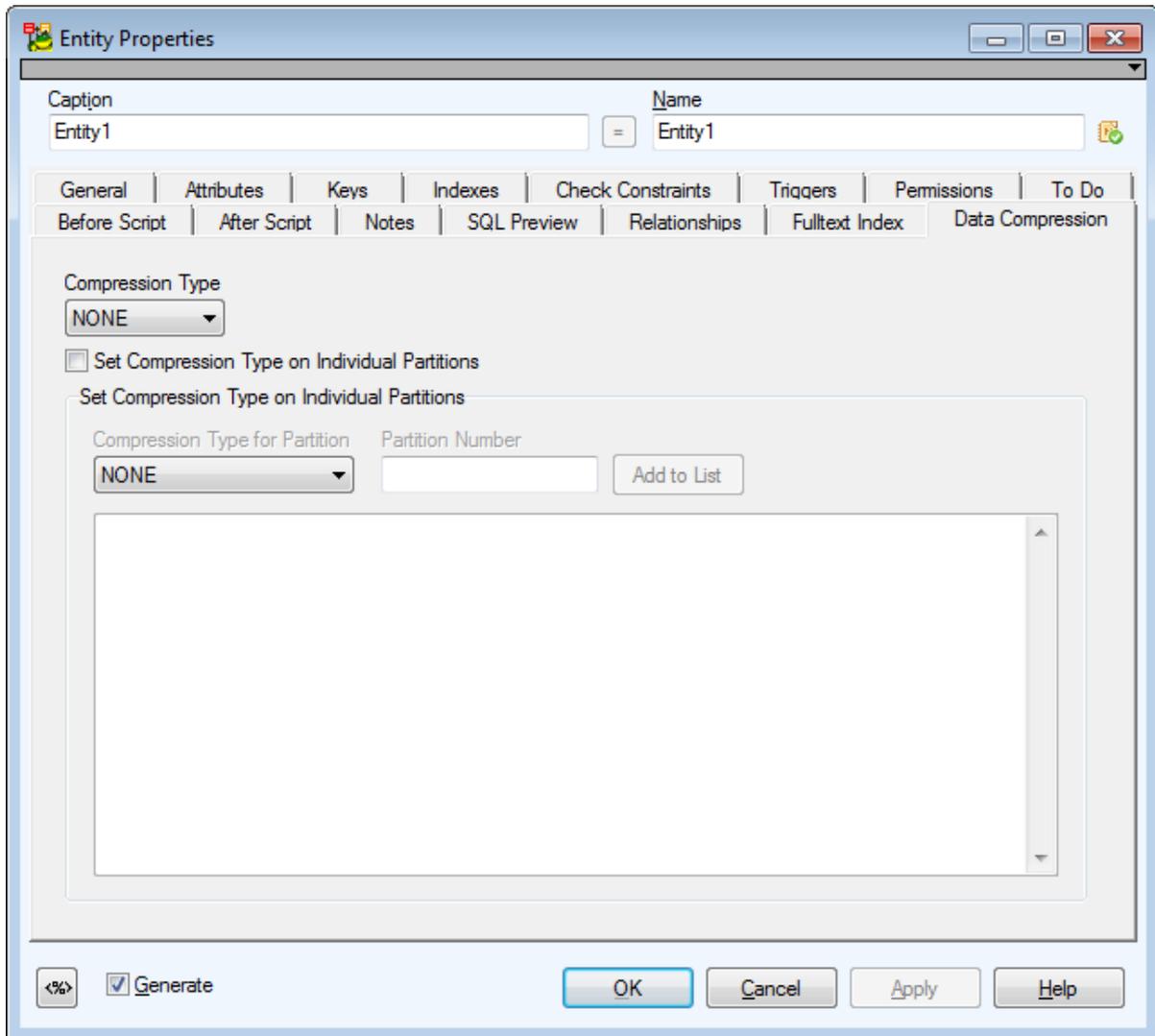
Fulltext Index tab, new combo box **Search Property List**.



SearchPropertyList object has only listing function. It is not possible to define it (CREATE/DROP/ALTER not supported).



## Data Compression Tab



## Description Column

The screenshot shows the 'Entity Properties - E' dialog box. At the top, there are fields for 'Caption' (Entity1) and 'Name' (Entity1). Below these are several tabs: 'Before Script', 'After Script', 'Notes', 'SQL Preview', 'Relationships', 'Fulltext Index', 'General', 'Attributes', 'Keys', 'Indexes', 'Check Constraints', 'Triggers', 'Permissions', and 'To Do'. The 'Data Space' section is highlighted with a red box and contains two radio buttons: 'Filegroup' (selected) and 'Partition Scheme'. Under 'Filegroup', there are two dropdown menus, both set to '-- Not Specified --'. Under 'Partition Scheme', there is one dropdown menu set to '-- Not Specified --' and a text field for 'Partition Column Name'. Below the 'Data Space' section, the 'Description' text box is highlighted with a red box. At the bottom, there is a 'Category' dropdown set to '-- None --', a checked 'Generate' checkbox, and buttons for '<%', 'OK', 'Cancel', 'Apply', and 'Help'.

Take notice of the **Data Space** area. If you click **Filegroup**, list of file groups will become available. If you select **Partition Scheme**, the options will change.

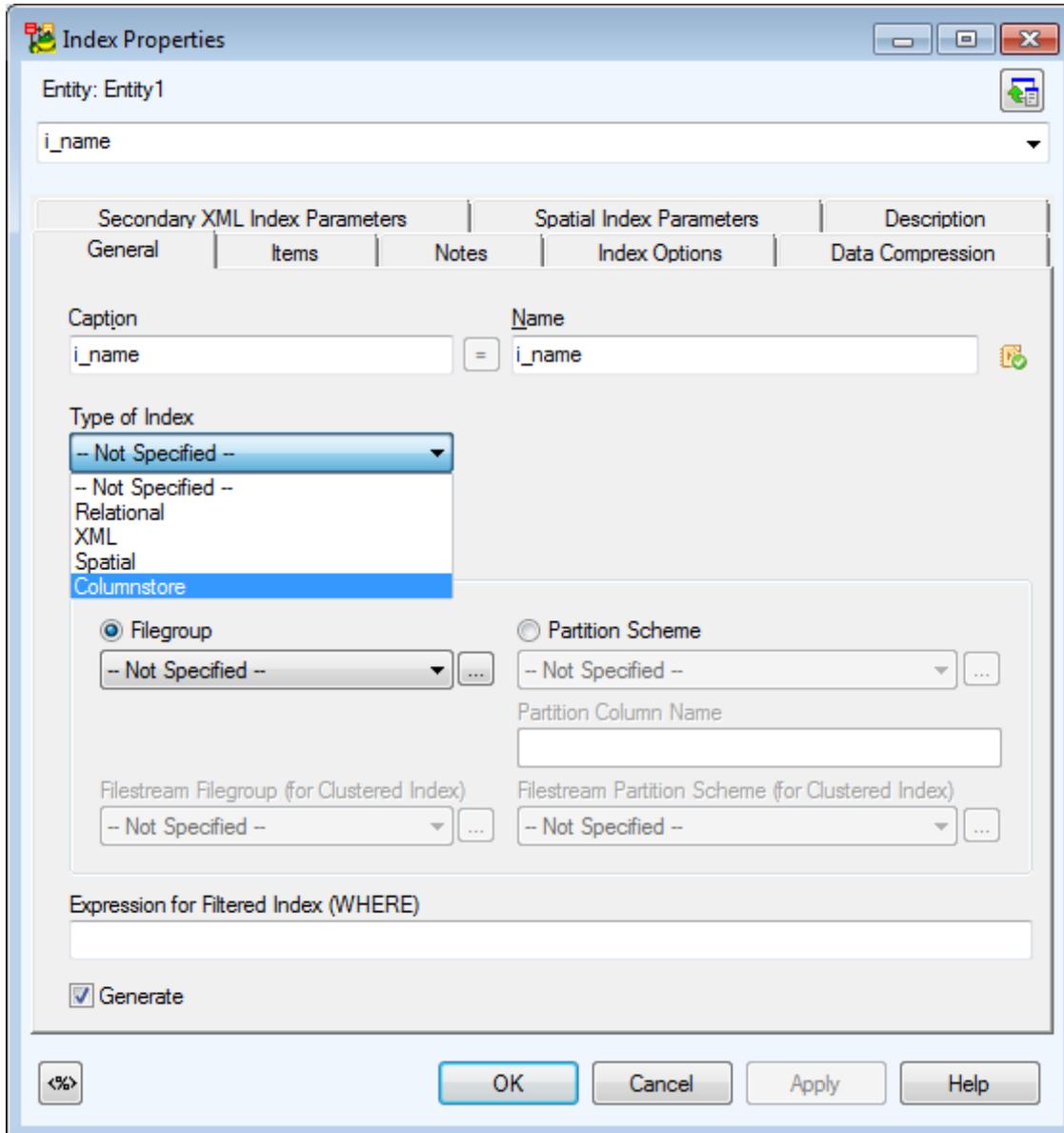
**Description** text box - The description text is generated in final DDL script.

**Note:** To generate the table, column descriptions for SQL Server db in Toad Data Modeler, **schema/owner has to be defined**. Please see the **Entity Properties** form | **General** tab and from the **Schema** box select a schema or click the icon on the right to open the **Schema** dialog and define a new schema, then select it for the entity.

# Index

## Columnstore option

**Type of Index** box - new **Columnstore** index type. For Columnstore indexes, many options are available, though they are not valid and they do not get generated.



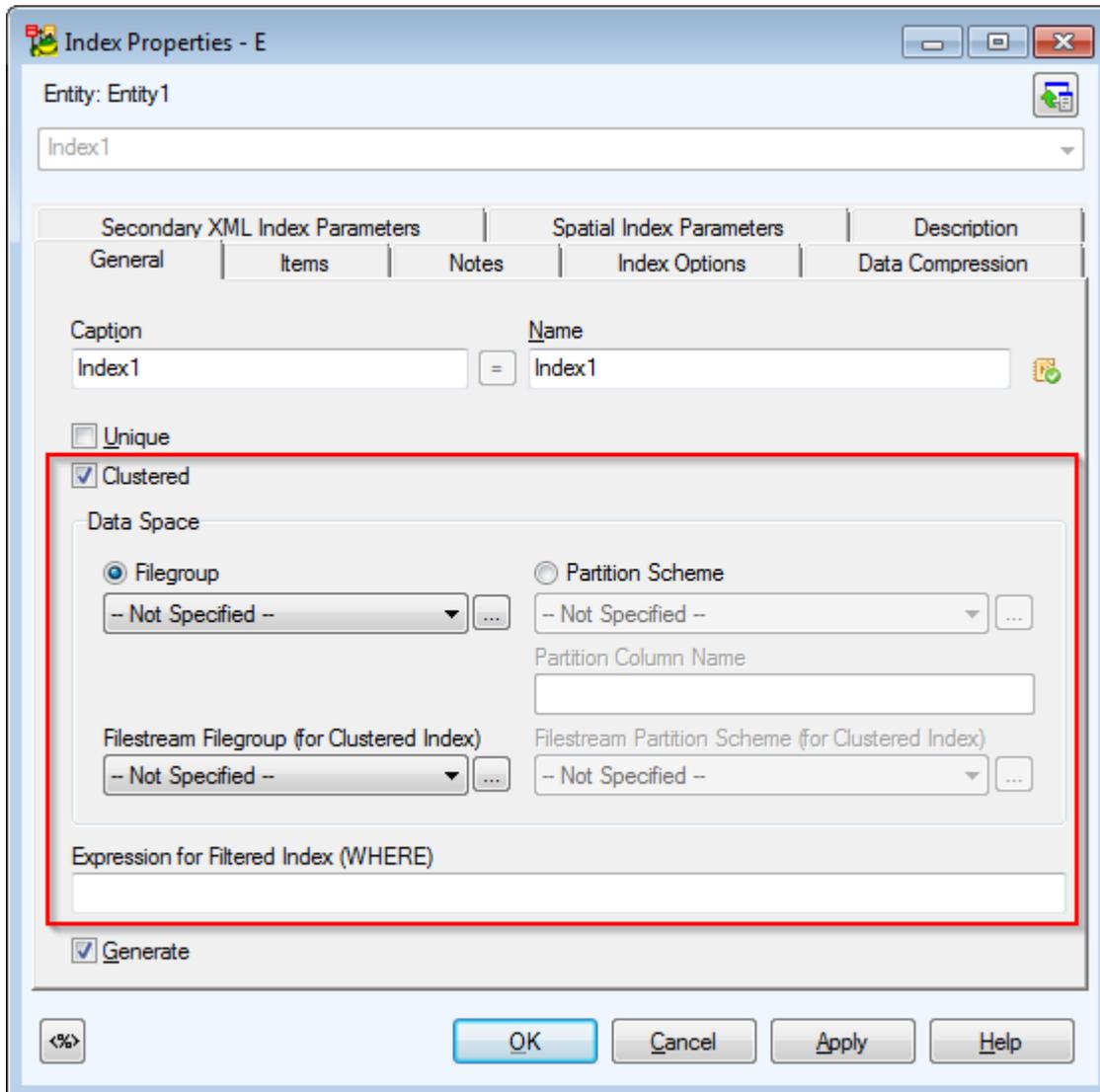
## Spatial Index Parameters Tab

The screenshot shows the 'Index Properties - E' dialog box with the 'Spatial Index Parameters' tab selected. The 'Entity' is 'Entity 1' and the 'i\_name' field is empty. The 'Spatial Tesselation Scheme' dropdown is highlighted with a red box and shows 'GEOMETRY\_GRID' selected. Below it are the 'Bounding Box (for Geometric Spatial Index)' fields for X-min, Y-min, X-max, and Y-max. There are also 'Grids' dropdowns for Level 1, Level 2, Level 3, and Level 4, all set to '<NONE>'. At the bottom is a 'Cells Per Object' field.

New **Spatial Tesselation Scheme** box with new options *GEOMETRY\_AUTO\_GRID* and *GEOGRAPHY\_AUTO\_GRID*. Based on selection, further **Bounding Box** and **Grids** options are enabled or disabled.

**Cells Per Object**— Default value changed to empty from 16.

## The FilestreamFilegroup box

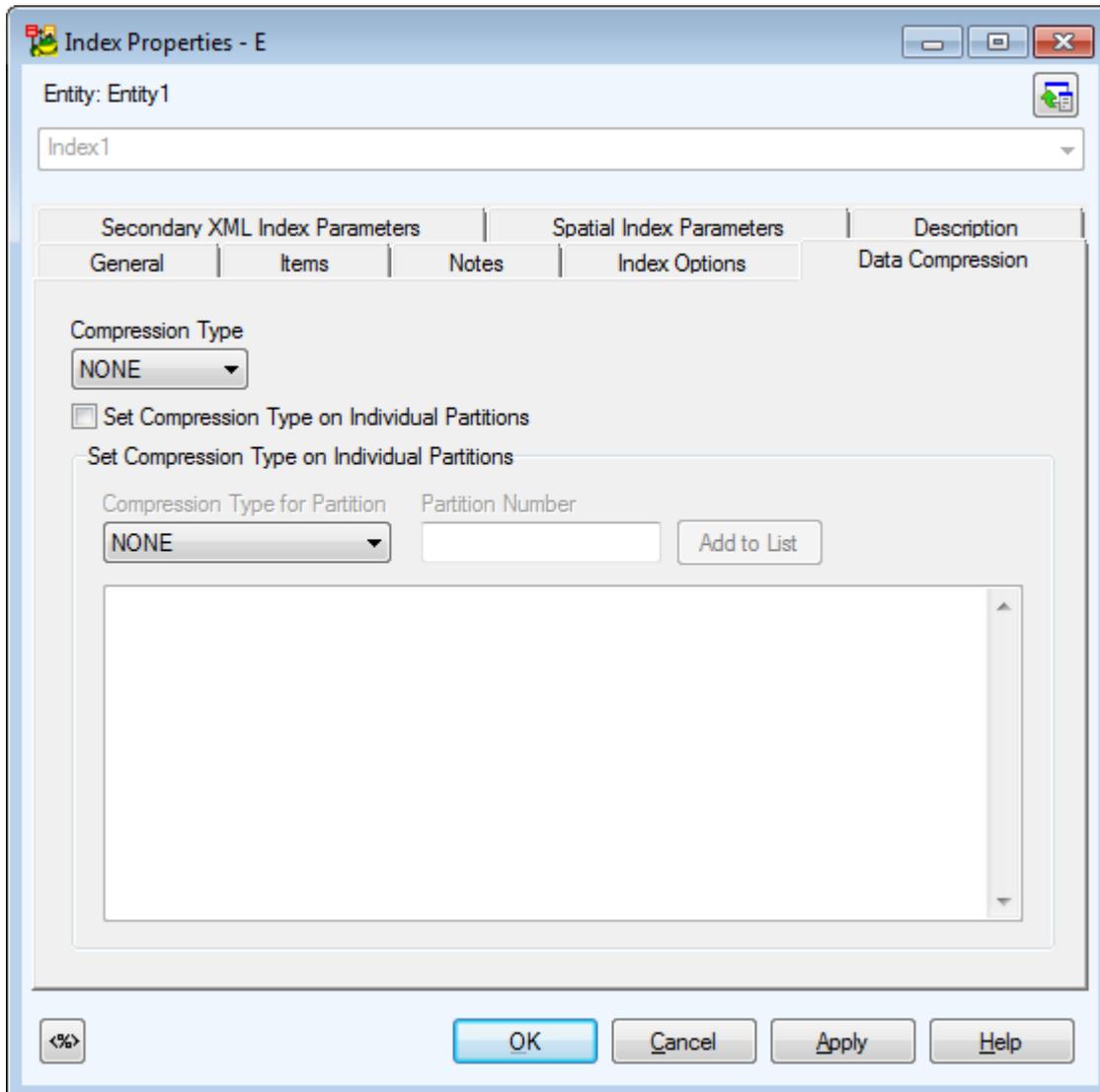


The **FilestreamFilegroup** box is available only for Clustered indexes - see the **Data Space** area. In tables that are not partitioned, you can make selection from a new list **Filestream Filegroup**. In partitioned tables, you can make selection from a new list **Partition Scheme**.

**i** Note: To be able to define Filestream Filegroup, the database requires columns with the Filestream property in a table.

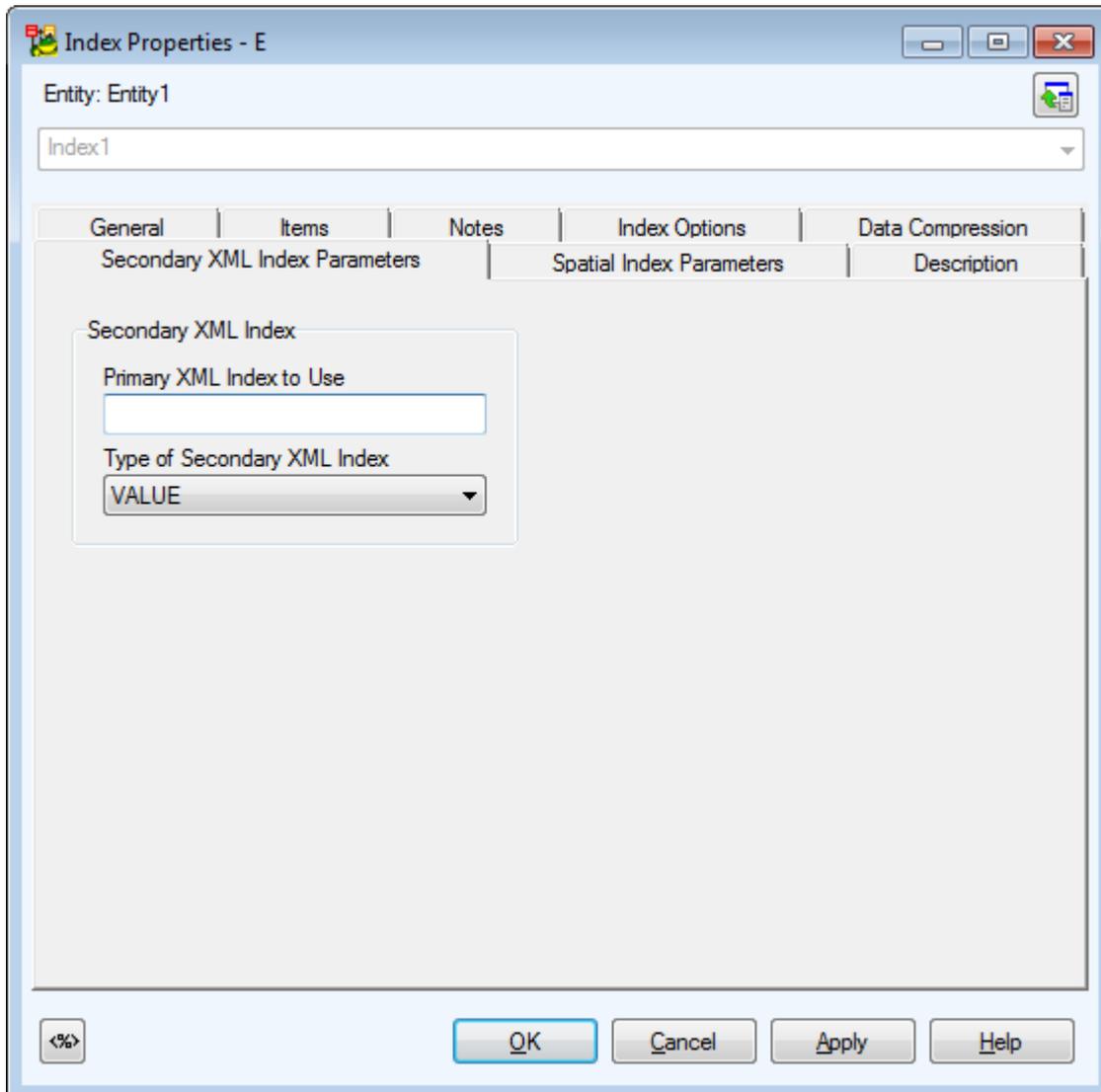
Possibility to define Filtered Index in the **Expression for Filtered Index (WHERE)** box.

## Data Compression Tab



On this tab, you can set Data Compression either for all indexes (combo box "Compression Type") or for particular partitions. (To accomplish this, it is necessary to select checkbox **Set Compression Type on Individual Partitions** and fill out the box in the **Set Compression Type on Individual Partitions** area. See the auxiliary combo box **Compression Type for Partition**, box **Partition Number** and button **Add to List**).

## Secondary XML Index Parameters Tab



## Spatial Index Parameters Tab

Entity: Entity1

Index1

General | Items | Notes | Index Options | Data Compression

Secondary XML Index Parameters | Spatial Index Parameters | Description

Bounding Box (for Geometric Spatial Index)

X-min:  Y-min:

X-max:  Y-max:

Grids

Level 1:  Level 2:  Level 3:  Level 4:

Cells Per Object:

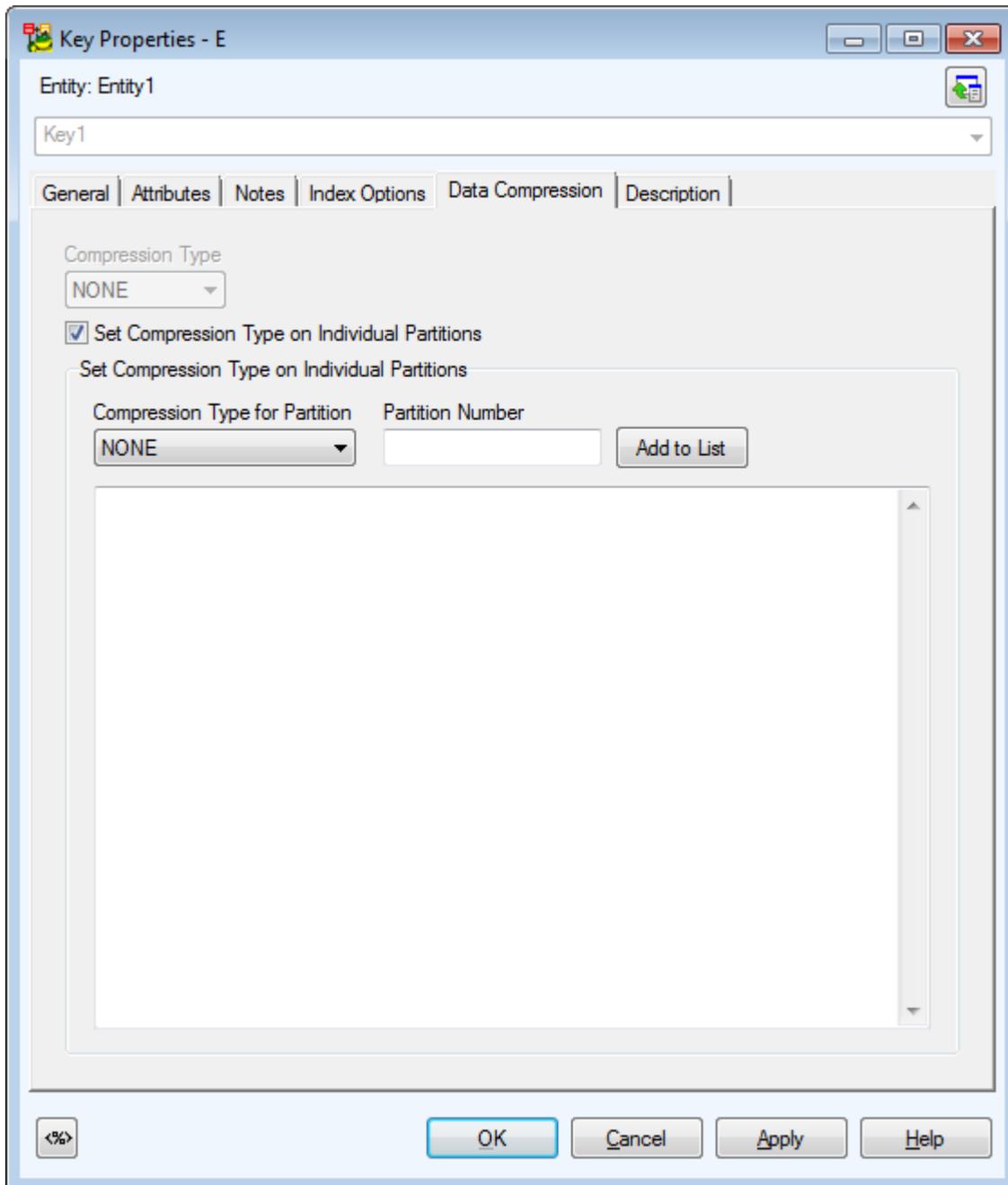
<%> OK Cancel Apply Help

**i** Note: Now index can be generated as normal (relational) index, primary XML index, secondary XML index and Spatial index. What index will be generated is selected automatically by the following conditions (particularly by data type of particular attribute):

- If index has only one column and this column is of XML type, XML index will be generated. For the XML index it is searched whether the box "Primary XML index..." is filled out. If it is filled out, then it is a secondary XML index. Otherwise it is a primary XML index.
- If index has only one column and this column is of Geometry or Geography type, Spatial index will be generated then.
- Database requires a primary key in a table if Spatial Index exists. User has to arrange it on his own.

# Key

## Data Compression Tab



On the **Data Compression** tab, you can set Data Compression either for all key (combo box **Compression Type**) or for particular partitions. (To accomplish this, it is necessary to select checkbox **Set Compression Type on**

**Individual Partitions** and fill out the box in the **Set Compression Type on Individual Partitions** area. See the auxiliary combo box **Compression Type for Partition**, box **Partition Number** and button **Add to List**).

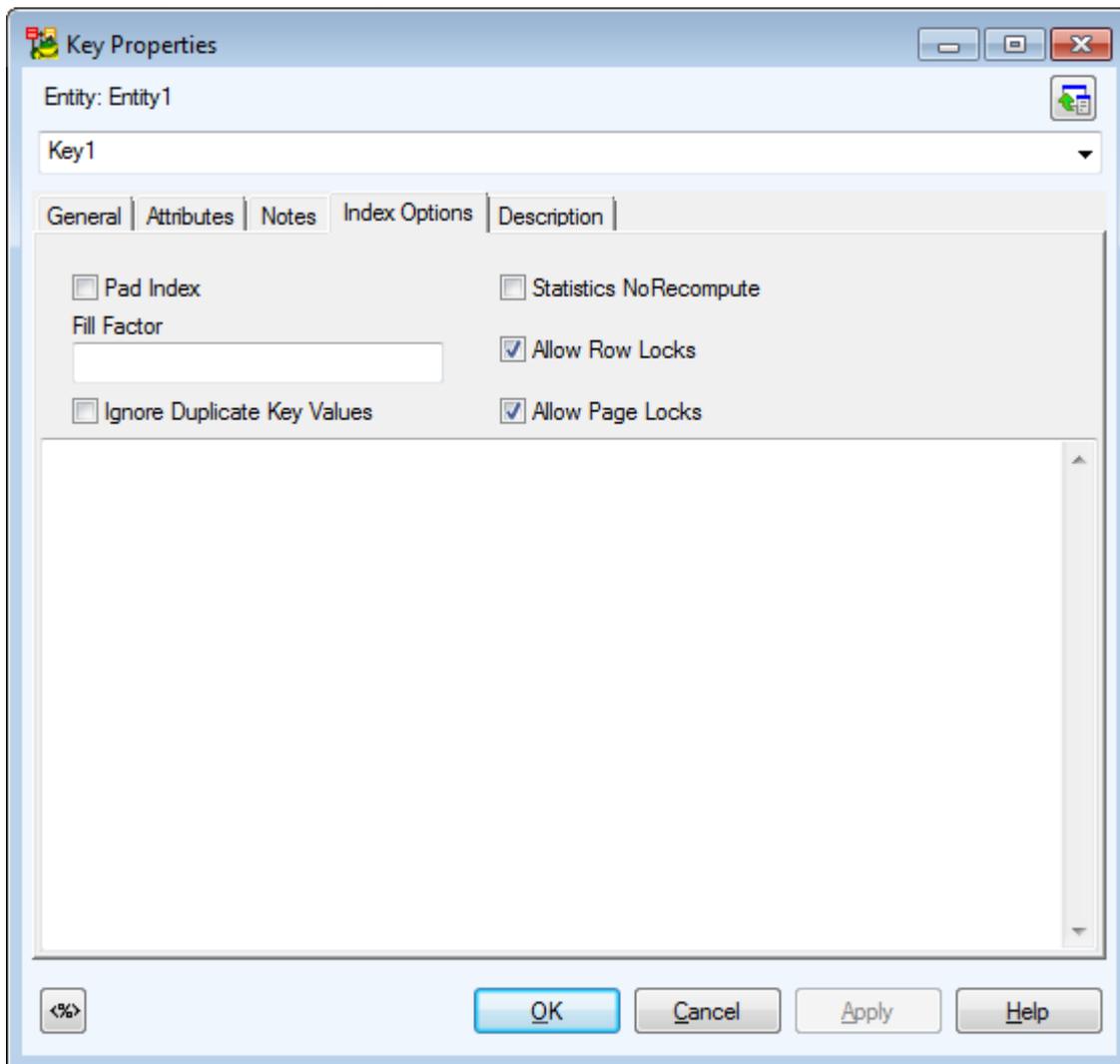
**i** Note: To load Data Compression defined in index during reverse engineering, it is necessary to select the **Load Index Options** checkbox in the Reverse Engineering Wizard (it is selected by default).

**General** tab - take notice of the **Data Space** area. If you click **Filegroup**, list of file groups will become available. If you select **Partition Scheme**, schemes will be available.

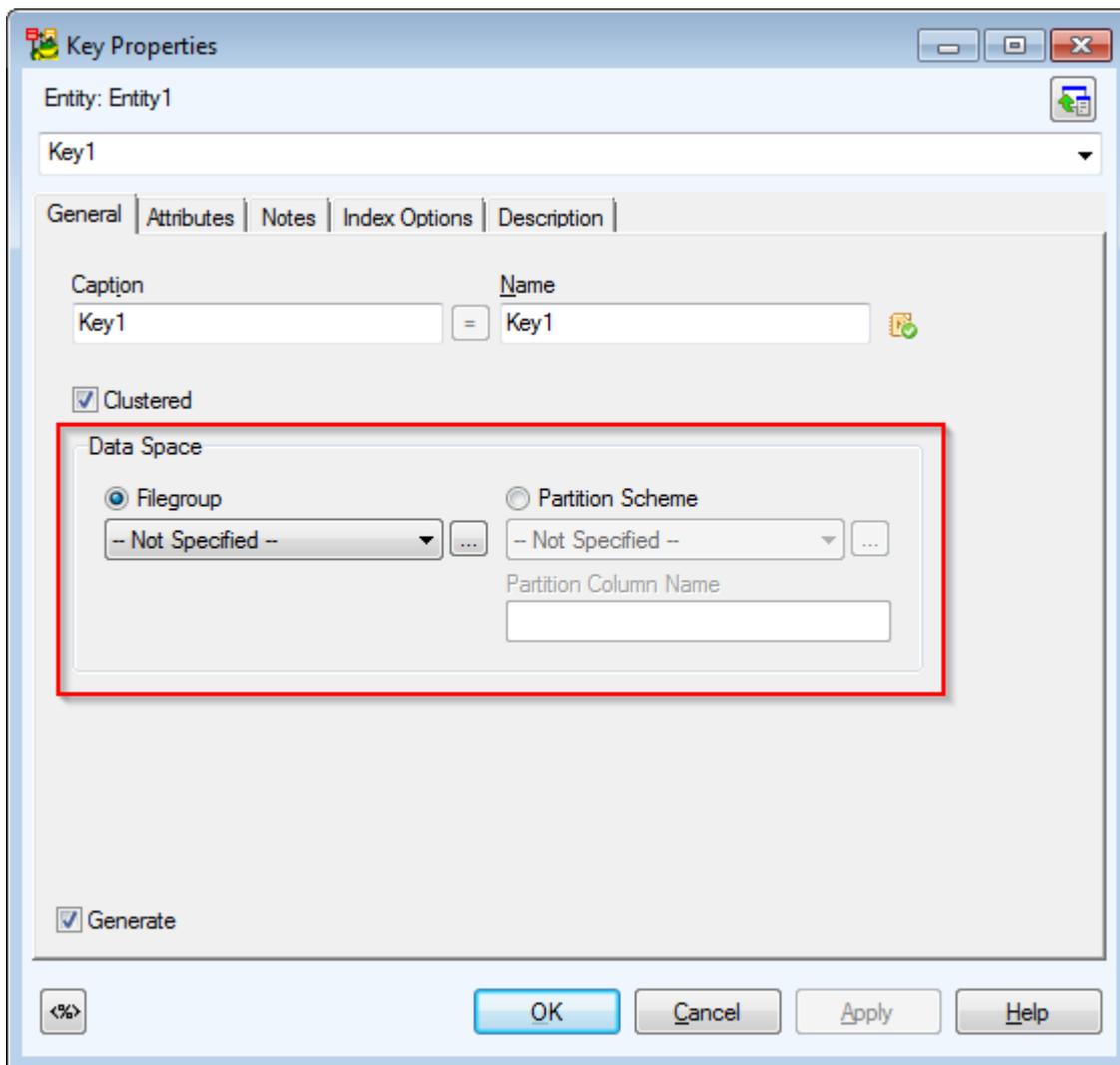
**Index Options** tab - Detailed settings for index of the key should be defined here.

## Index Options Tab

Detailed settings for index of the key should be defined here.

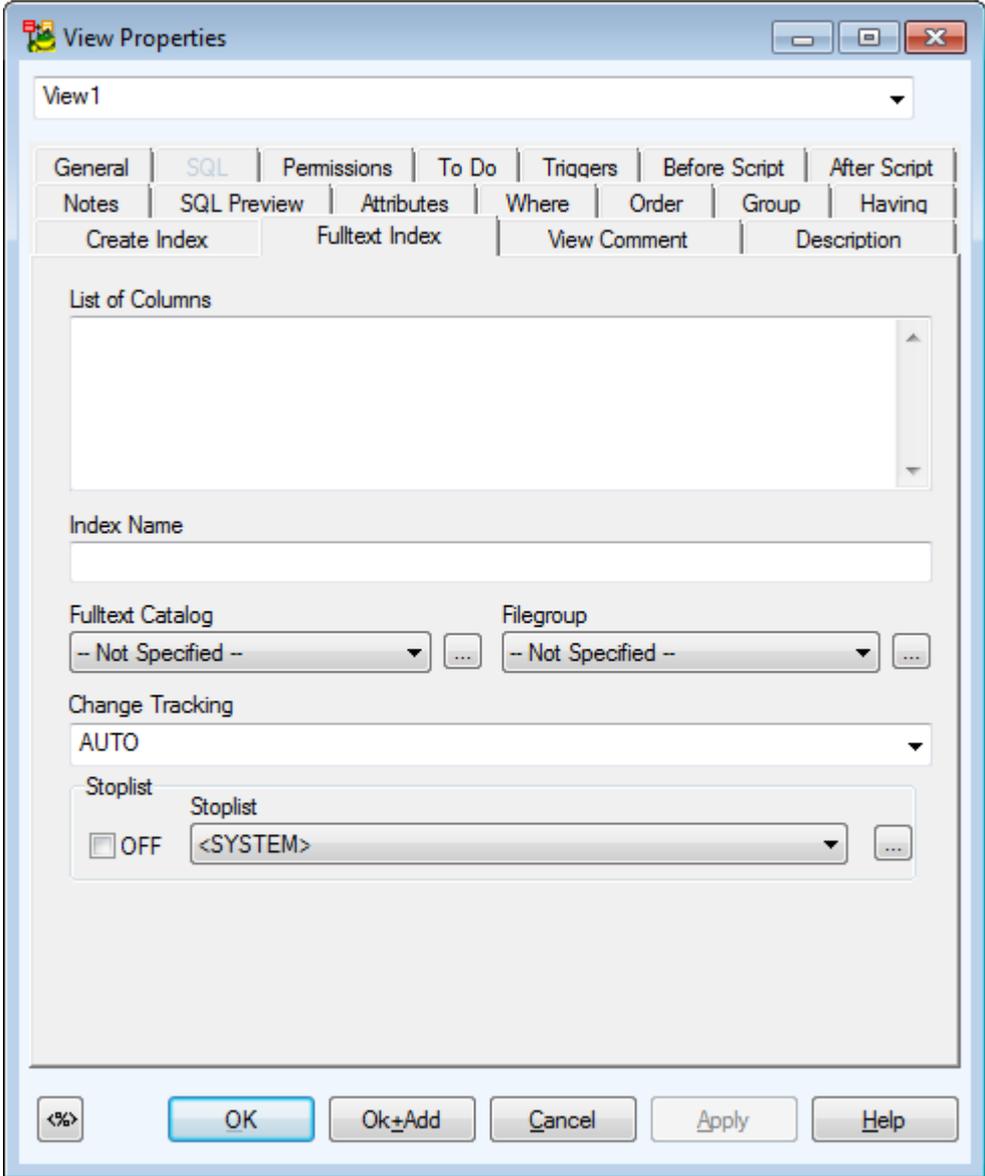


# Data Space

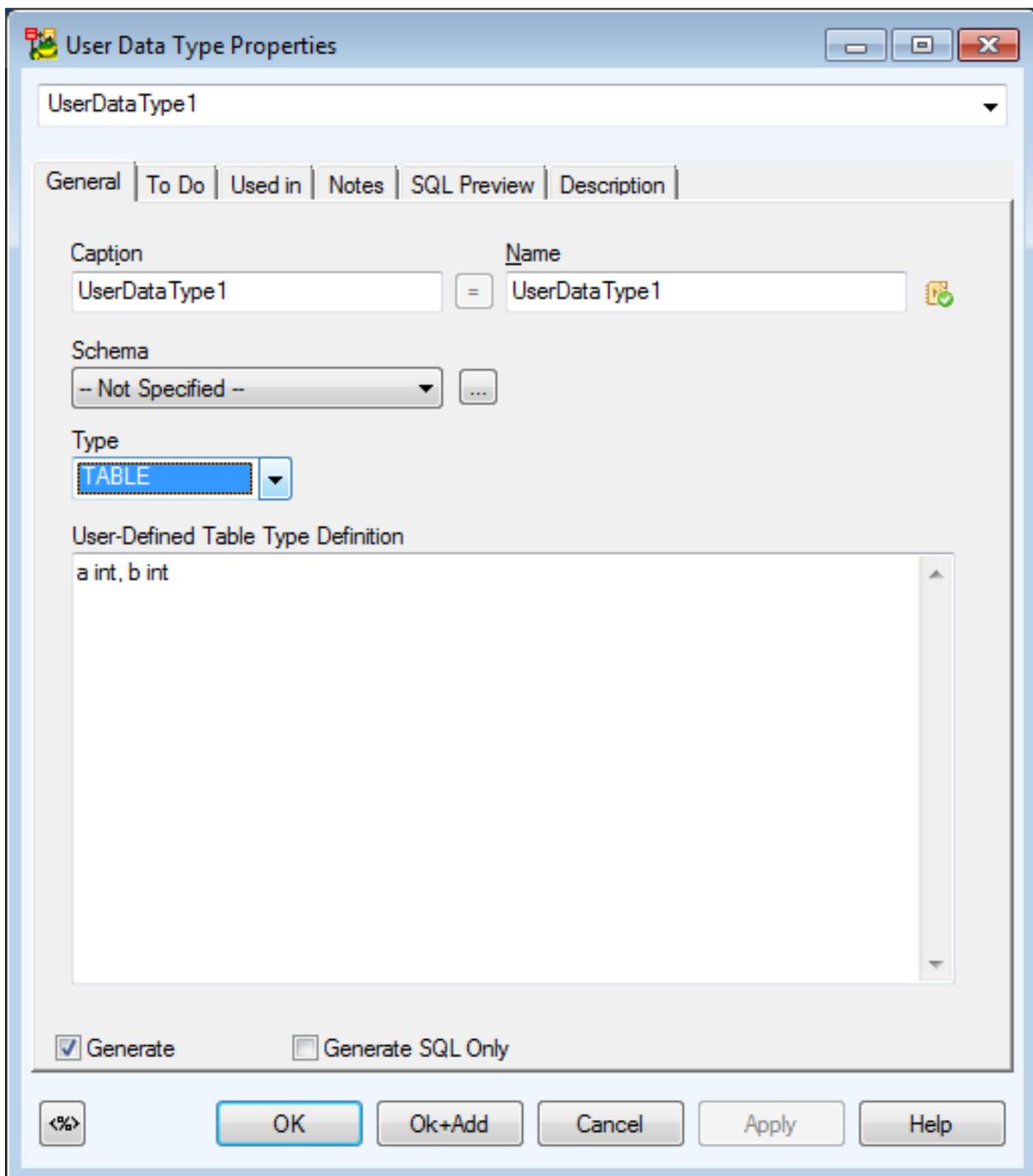


Take notice of the **Data Space** area. If you click **Filegroup**, list of file groups will become available. If you select **Partition Scheme**, schemes will be available.

# View



# User Data Type

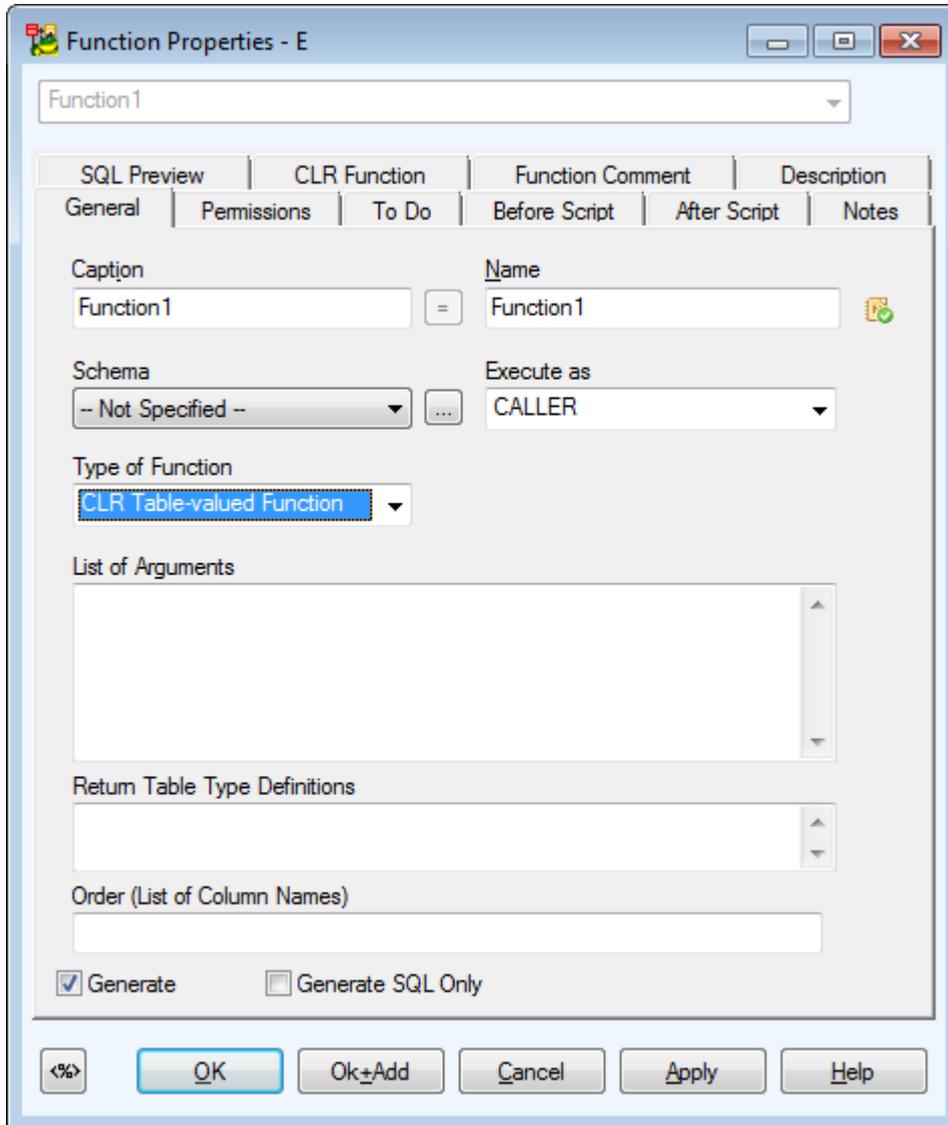


When you select **TABLE** from the **Type** box, the **User-Defined Table Type Definition** window will display. Here, write the text that is defined in syntax in brackets after AS TABLE, for example "*a int, b int*" (without the quotation marks).

Although this type is available in attributes, it is not possible to use it there.

When you click the **Generate SQL Only** checkbox, the **SQL** tab will display.

## Function



When you select the **CLR Table-valuedFunction**, the **Order (List of Column Names)** box will become available. Here, you can define order (column names should be separated by commas).

When you select the **Aggregate Function**, you can write more input parameters separated by comma to the **List of Arguments** box.

# Extended Properties

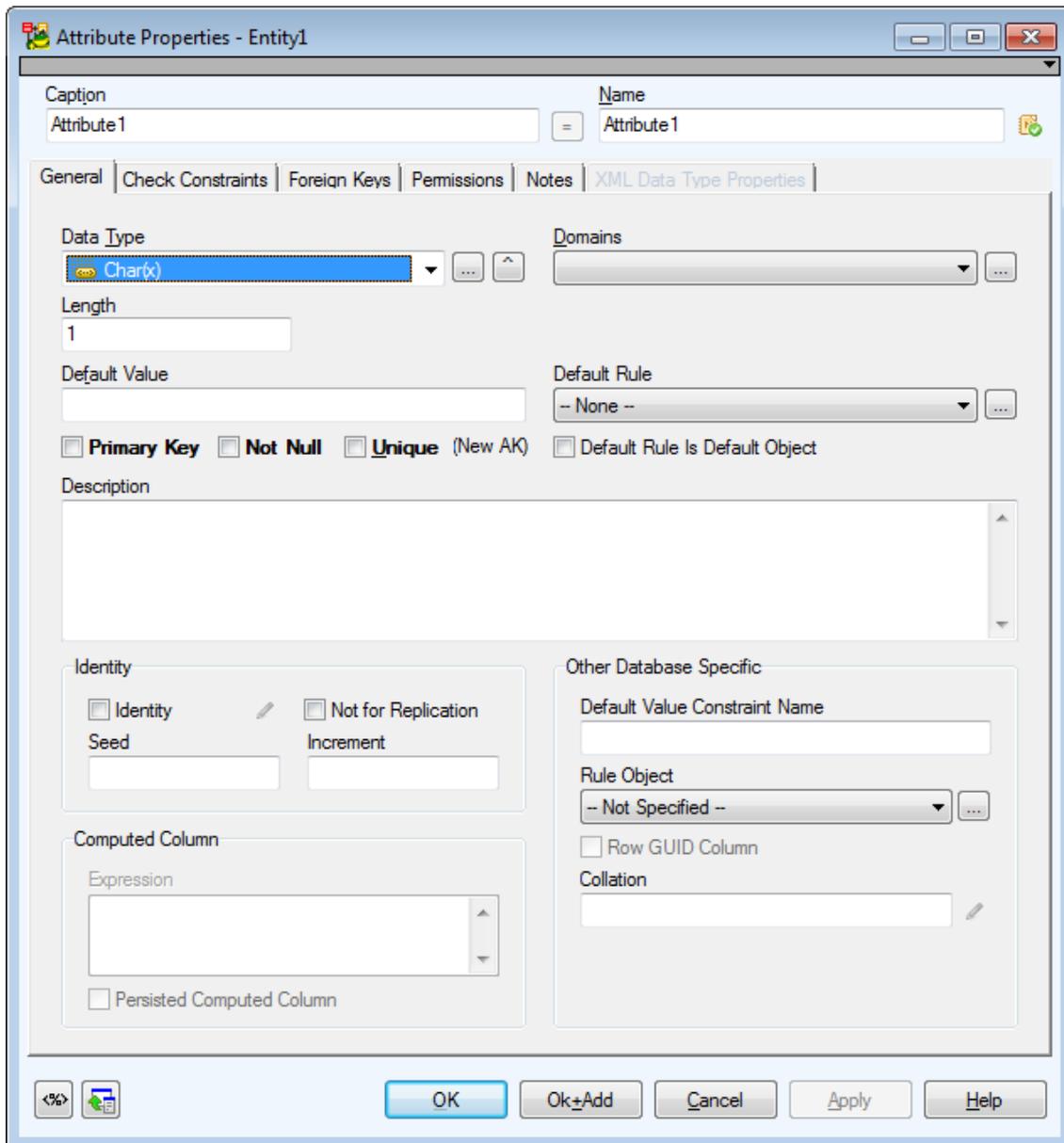
Extended Properties are supported for the following objects (generally where comments are supported):

1. Objects with new tab **Extended Properties**.
2. Objects for which Extended Properties are loaded during reverse engineering as a text as for example After Script, function parameters, procedure parameters, view columns, key/index/foreign key for Filetable.

See other objects in Model Explorer:

- **Filegroups**
- **Partition Schemes**
- **FilestreamFilegroups** - Similarly to Filegroups, FilestreamFilegroups work in Toad Data Modeler only as a list. During script generation, the FilestreamFilegroups definition is not generated and during reverse engineering the definition is not loaded either (only names are loaded).
- **Stoplists** - Similarly to Filegroups, Stoplists work in Toad Data Modeler only as a list. During script generation, the Stoplism definition is not generated and during reverse engineering the definition is not loaded either (only names are loaded).
- **Partition Functions**
- **Fulltext Catalogs**
- **XML Schema Collections**
- **Assemblies**

# Attribute



## Data Types

*Char, NChar, NText, NVarChar, NVarChar(max), Text, VarChar, VarChar(max)* - the **Collation** box available.

*Unique identifier* - Select this item to display the **Row GUID Column** checkbox.

*XML* - You can define details on xml attribute on tab **XML Data Type Properties**.

*Computed Column* - It is not a data type. Select this item if you want to set column as computed.

**Identity** - Define Identity properties.

**Default rule Is Default Object** - Select this checkbox to bind default rule (known as default object in Microsoft SQL Server) to attribute.

**Rule Object** - Select a rule object that will be bound to attribute.

Data types of type DATE: *date*, *datetime2*, *datetime2(x)*, *datetimeoffset*, *datetimeoffset(x)*, *time*, *time(x)*.

Spatial data types: *geometry*, *geography*.

Other data types: *hierarchyid*.

**i** Note: Rowversion is generated instead of data type timestamp (they are synonyms and it is recommended to use rowversion).

The **Filestream** checkbox is displayed only for data type *varbinary(max)*.

The **Sparse** checkbox is not available for the following data types: *geography*, *geometry*, *image*, *ntext*, *text*, *timestamp*, *user-defined data types*, *computed column*.

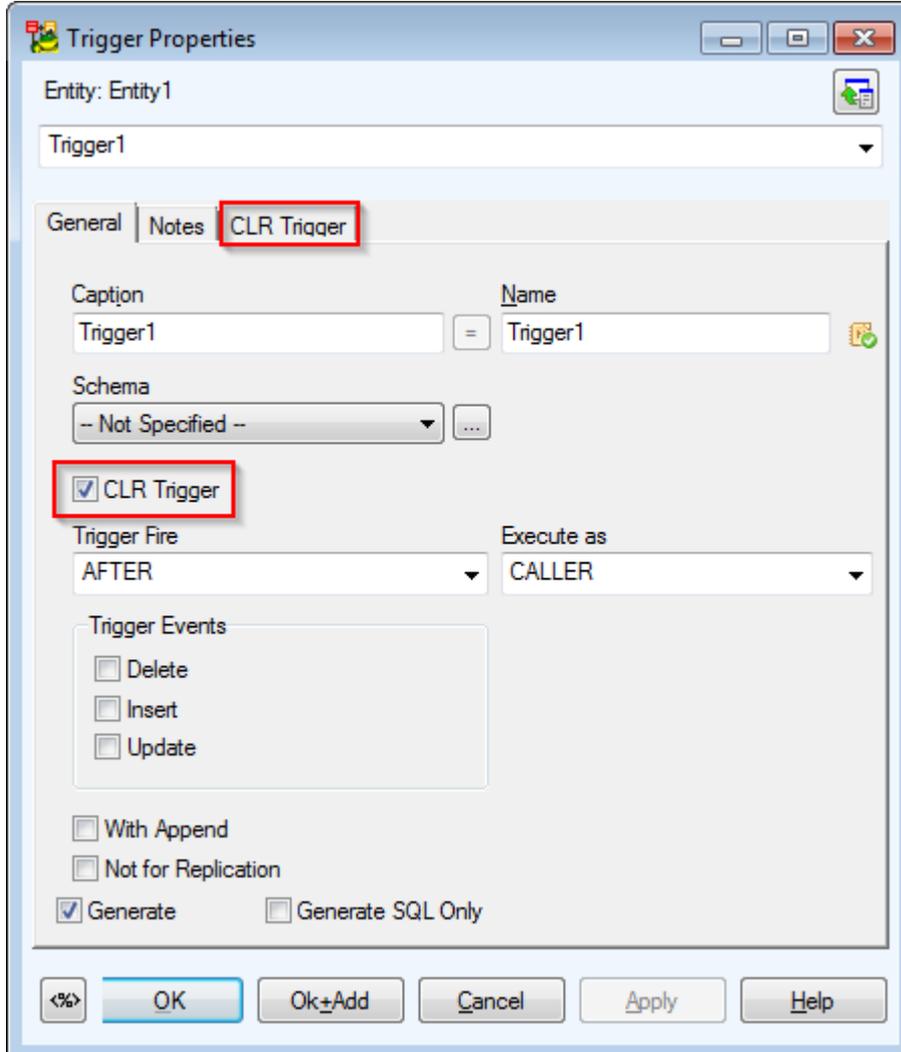
**Default Rule Is Default Object** - Select this checkbox if you want to bind default rule (known as default object in Microsoft SQL Server) to attribute.

**Rule Object** - Select a rule object that will be bound to attribute.

# Trigger

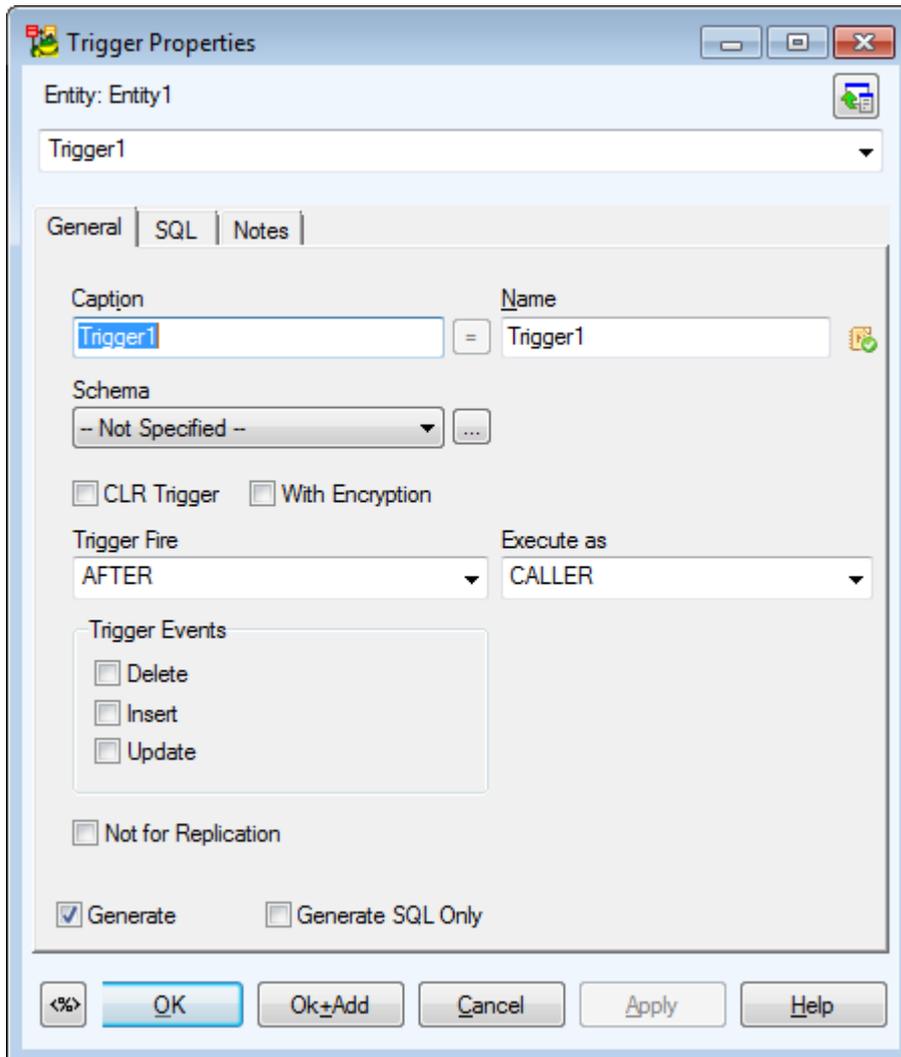
## CLR Trigger

**CLR Trigger** - If you select this checkbox, you can define CLR trigger on tab **CLR Trigger**.



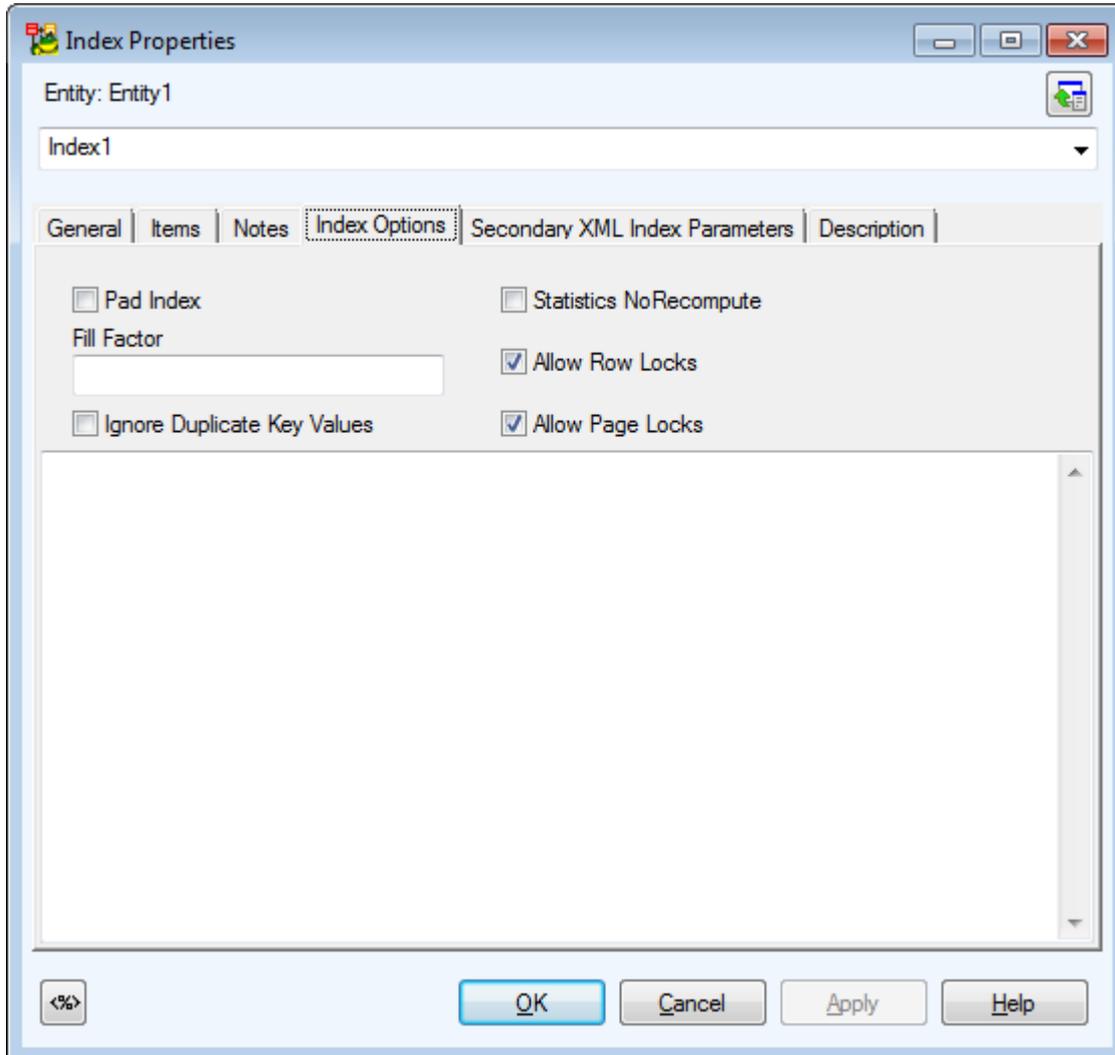
## Removal of the With Append option

The **With Append** option is not available now and the checkbox was removed from trigger.

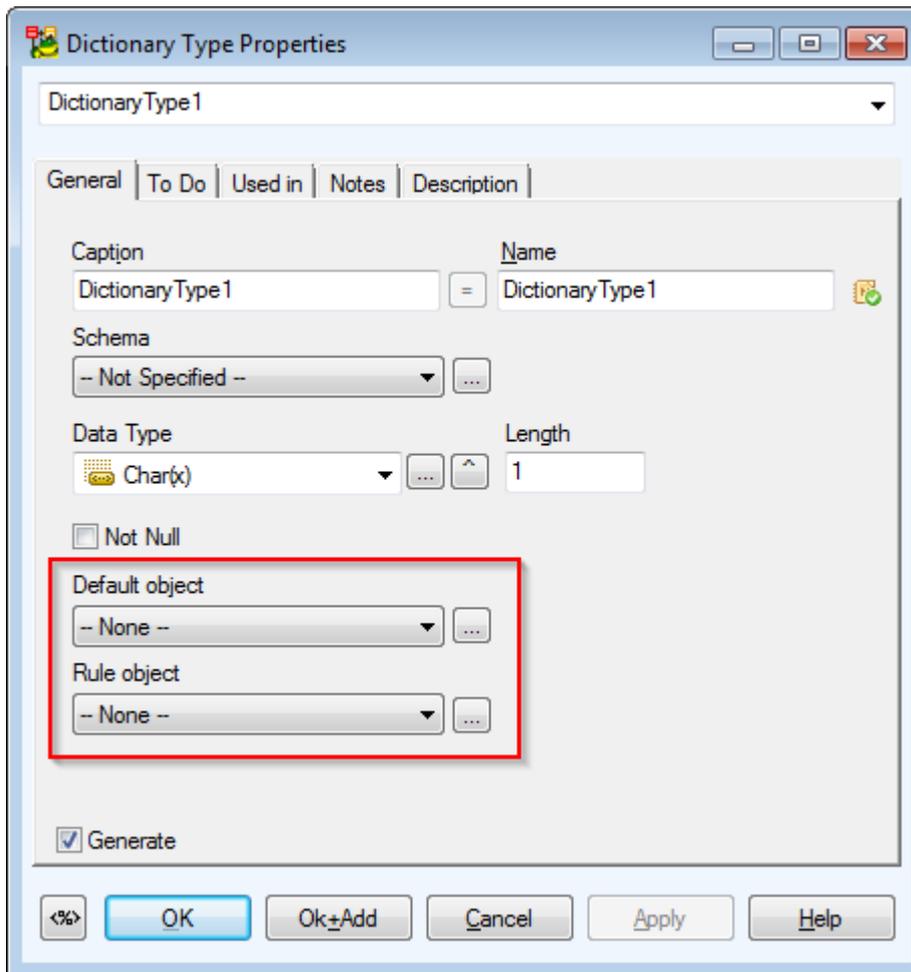


# Index Options Tab

Detailed settings of the index should be defined here.



# Dictionary Type

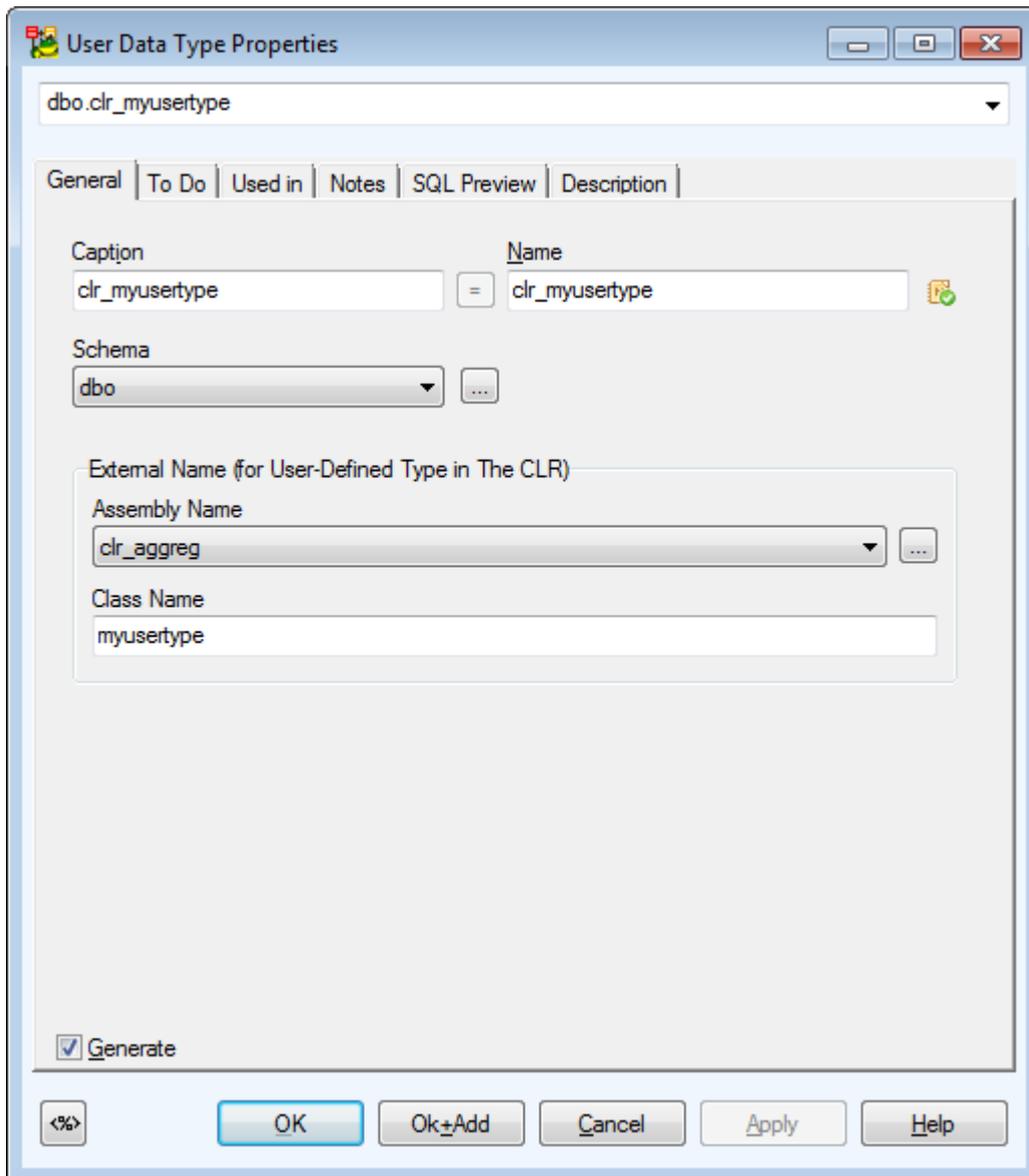


Here, you can enter alias data types.

**Default Object** - Select a default object that will be bound to dictionary type (alias data type in Microsoft SQL).

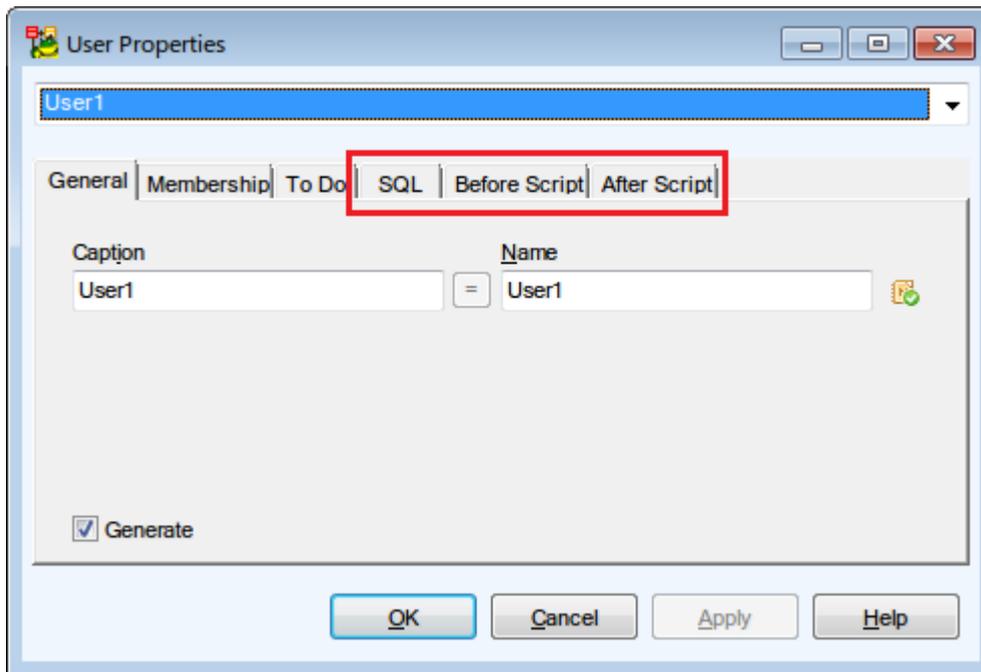
**Rule Object** - Select a rule object that will be bound to dictionary type (alias data type in Microsoft SQL).

# User Data Type



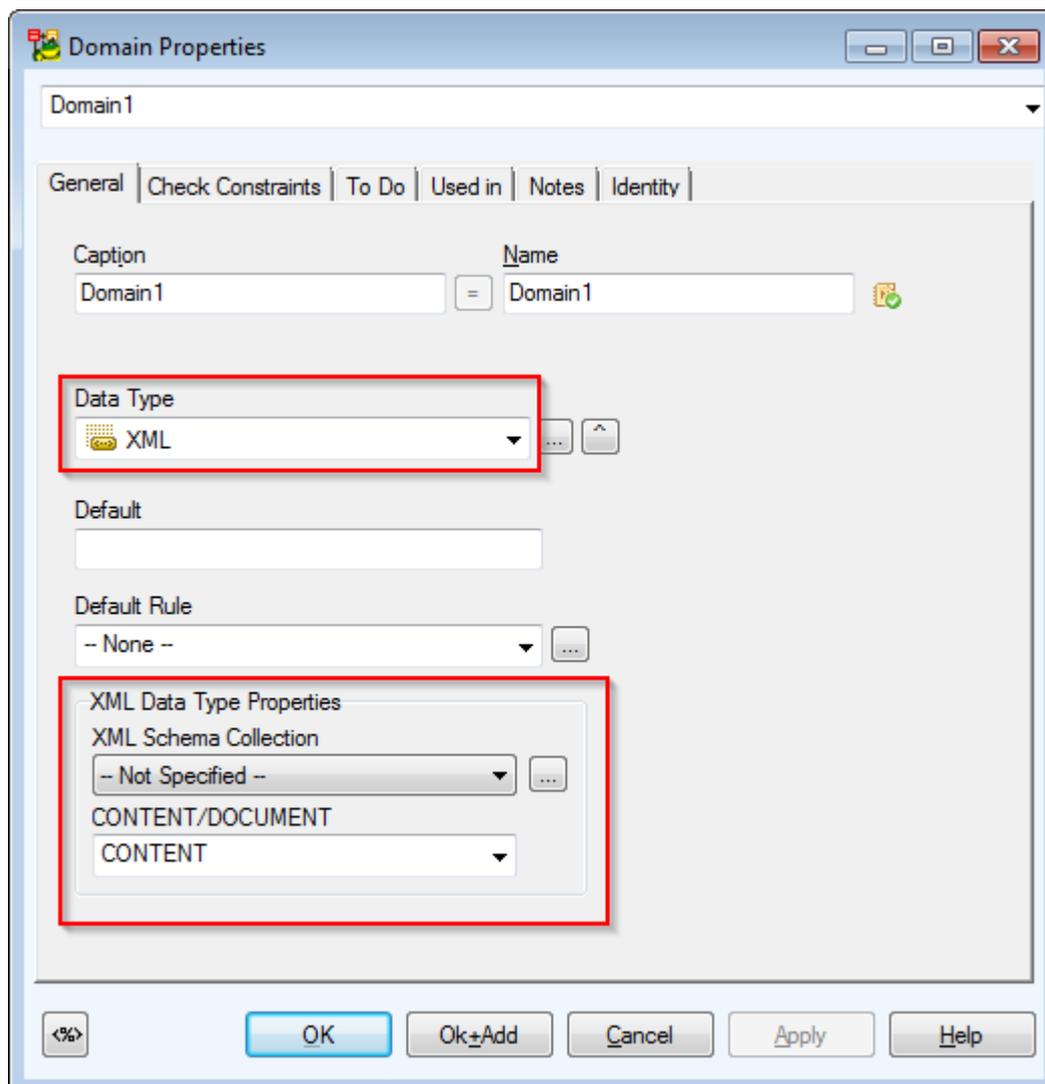
Here, you can enter CLR user-defined types.

# Users



- CREATE USER statements are supported for SQL Server 2005 and higher - they have to be defined in **User Properties** new tabs - **SQL**, **After Script**, **Before Script**.
- **User permissions** are loaded into Users **After Script** when SQL Server databases are Reverse Engineered.

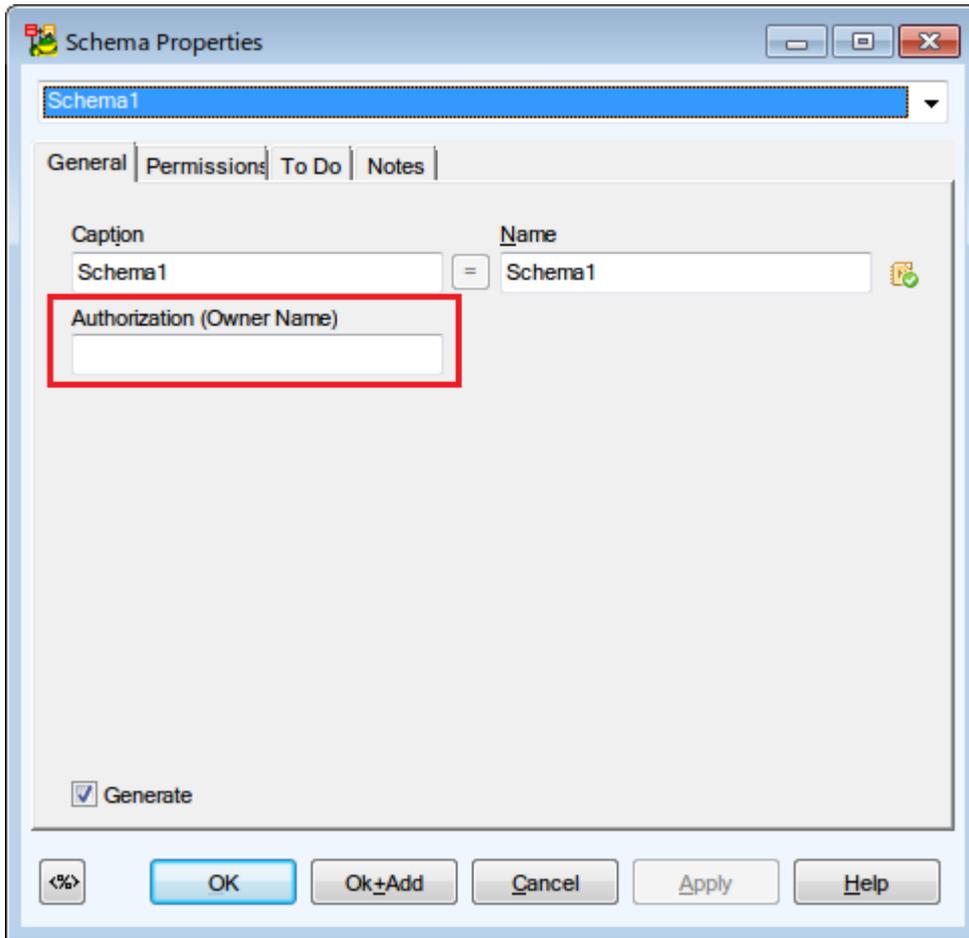
# Domain



Data Types:

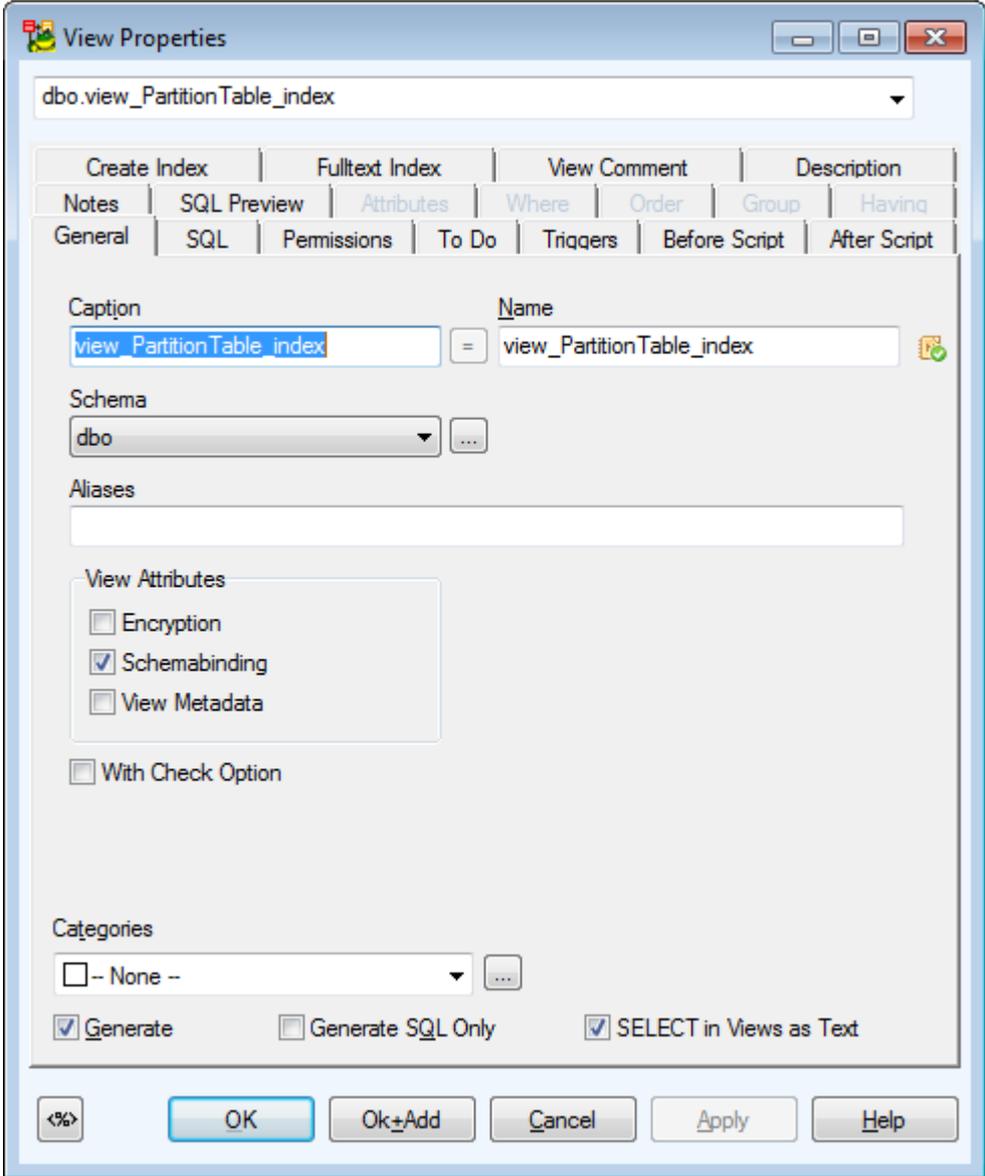
- *XML* data type - define XML data type properties.
- *Computed Column* - **Computed Column Expression** box, **Persisted Computed Column** checkbox.
- *Character* - define **Collation for the Column**.

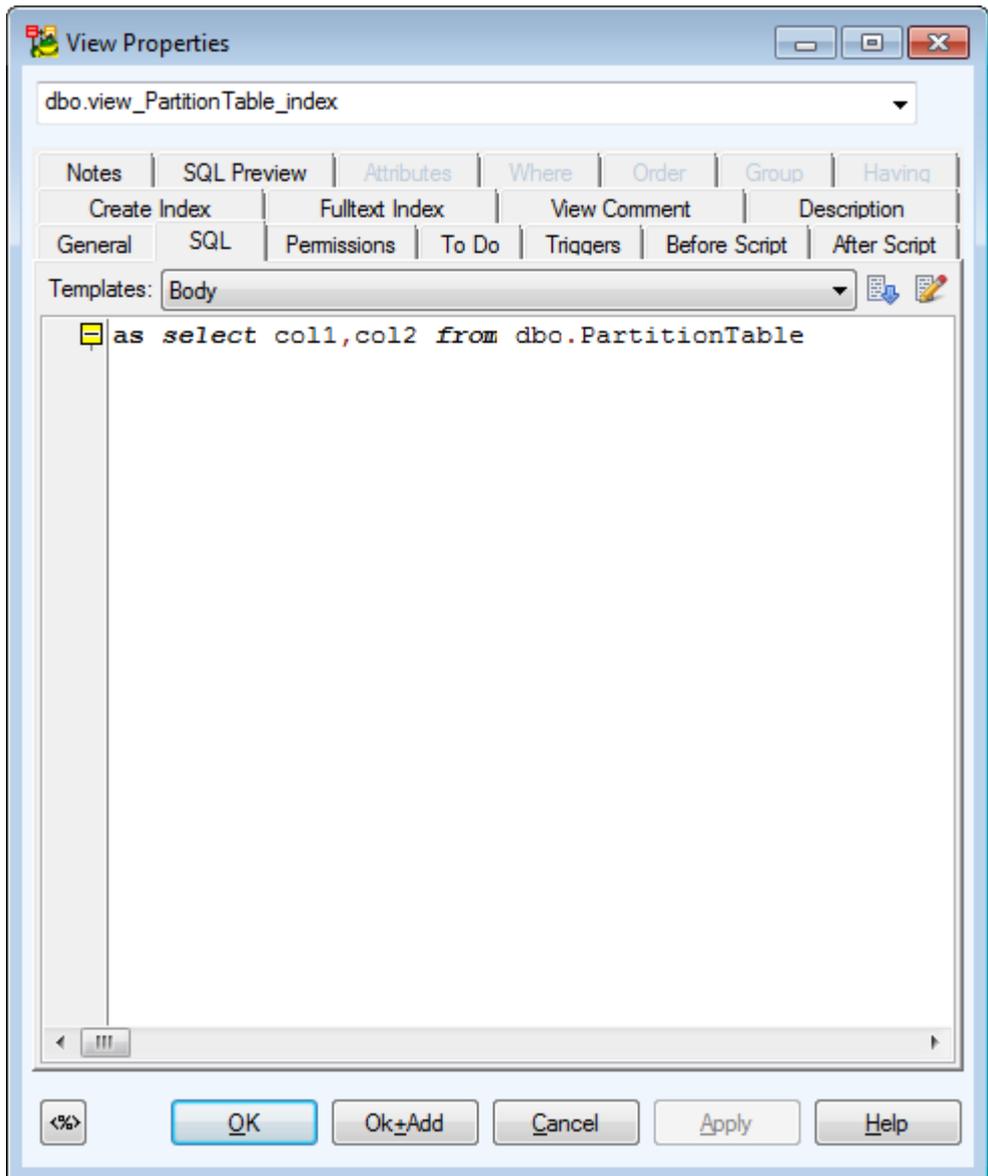
# Schema



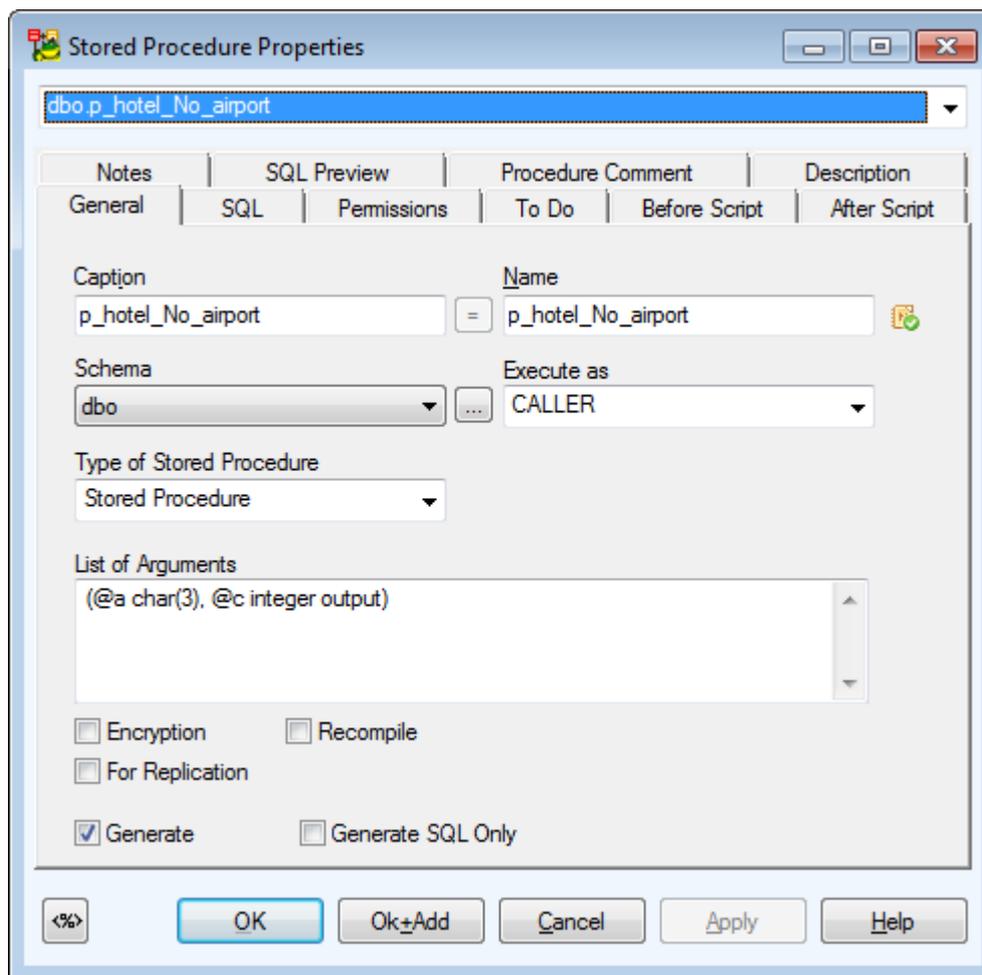
New property for **Schemas - Authorization (Owner Name)**

# View





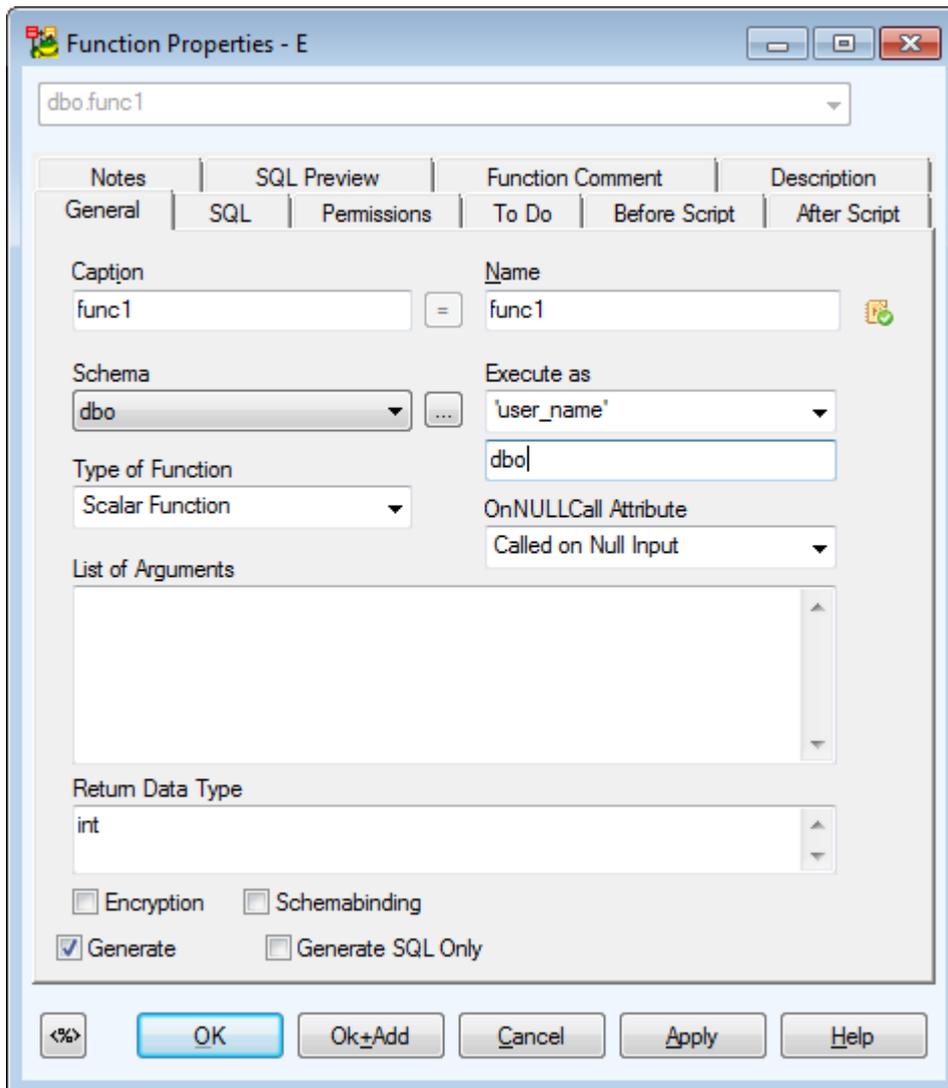
# Procedure



Types of Stored Procedure:

- Stored Procedure
- CLR Stored Procedure
- Extended Stored Procedure

# Functions



Types of functions:

- Scalar Function
- Table Valued Function
- CLR Scalar Function - Appropriate options and tab will appear.
- CLR Table Valued Function - Appropriate tab will appear in the dialog.
- Aggregate Function

Select a type and see the options that display then.

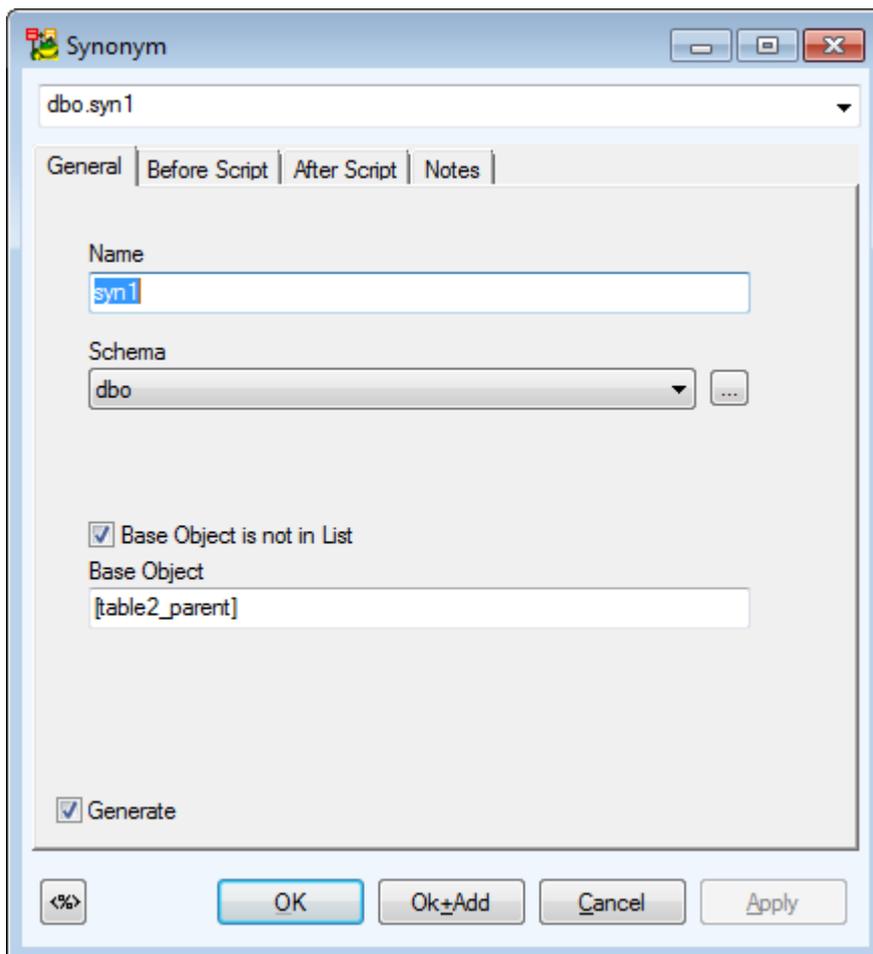
# Defaults

If you want to generate default as default object, you have to select the **Generate** checkbox.

# Check Constraint Rules

If you want to generate check constraint rule as rule object, you have to select the **Generate** checkbox.

# Synonym



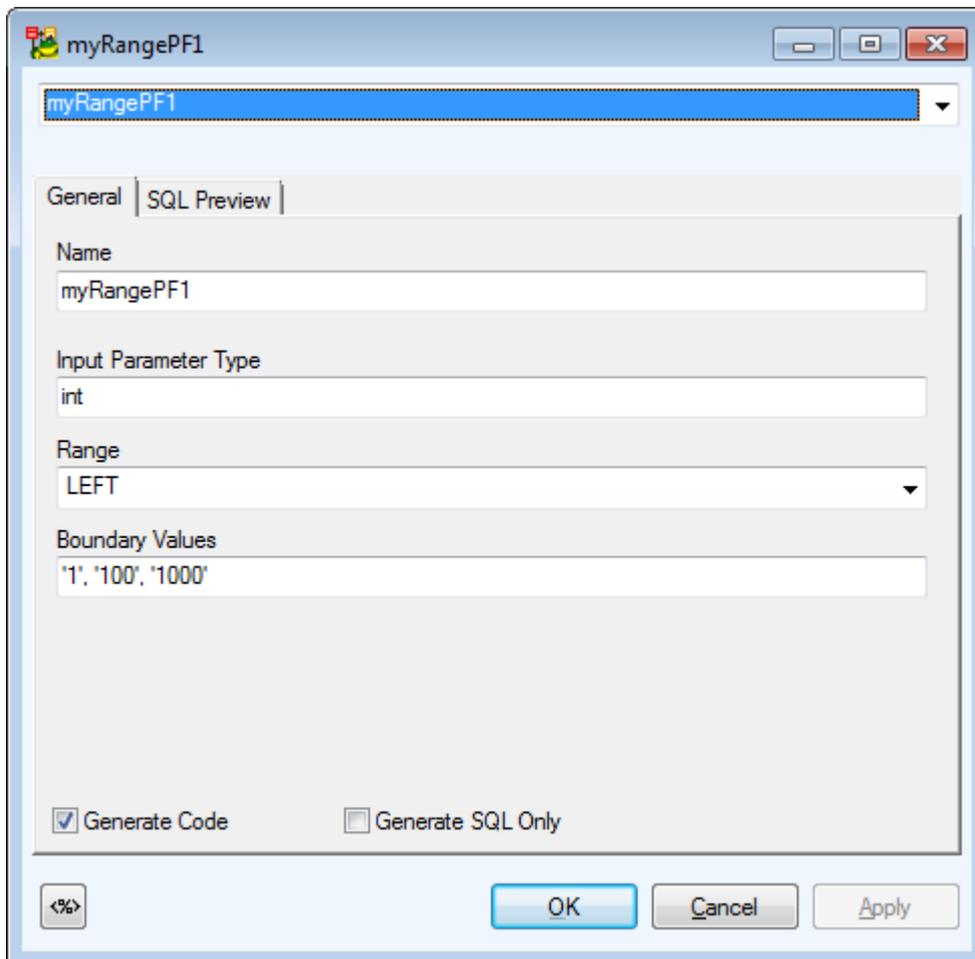
If you want to enter object that is not in the **Object** box, select the **Base Object is not in List** checkbox and enter the object in the new **Base Object** box.

See other objects in Model Explorer:

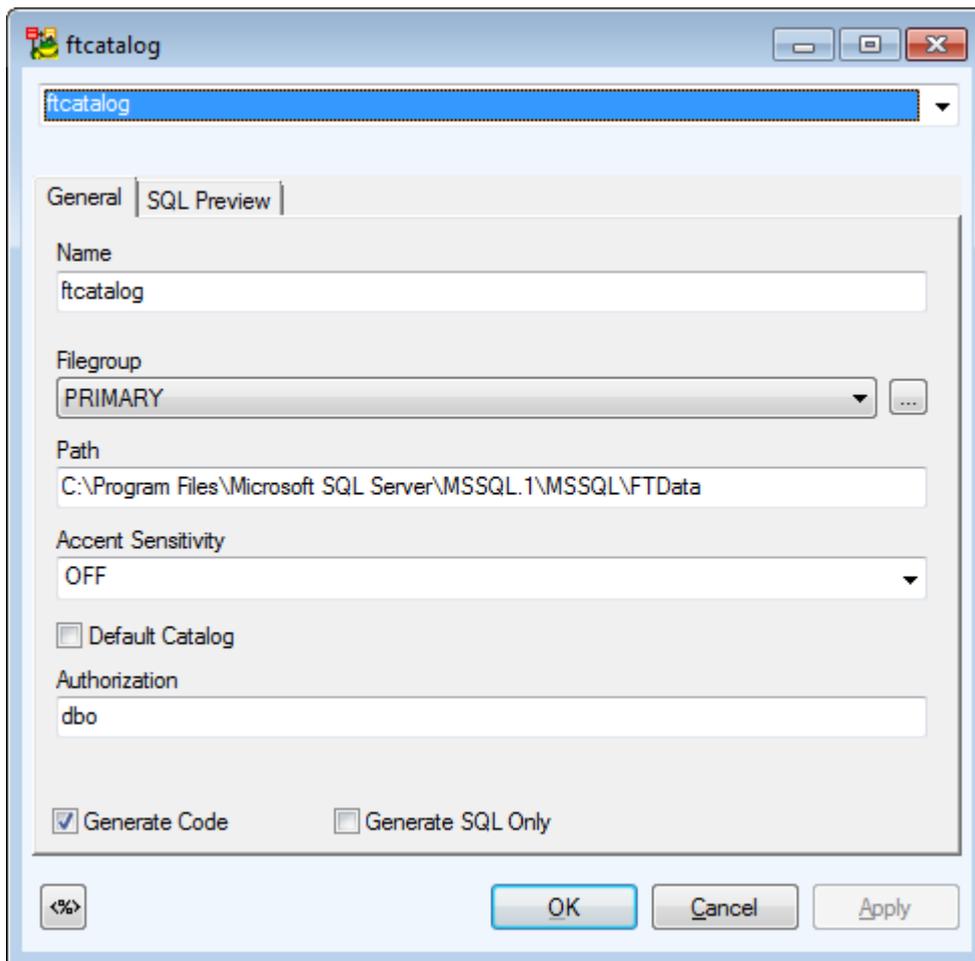
- Partition Functions
- Partition Schemes

- Filegroups
- XML Schema Collections
- Fulltext Catalogs
- Assemblies

**Example** of Partition Function:



Example of Fulltext Catalog:



Filegroup and Path parameters are used only in Microsoft SQL Server 2005.

If you want to write complete DDL statement for fulltext catalog, select **Generate SQL Only** checkbox and write the statement to **SQL** tab that will display then.

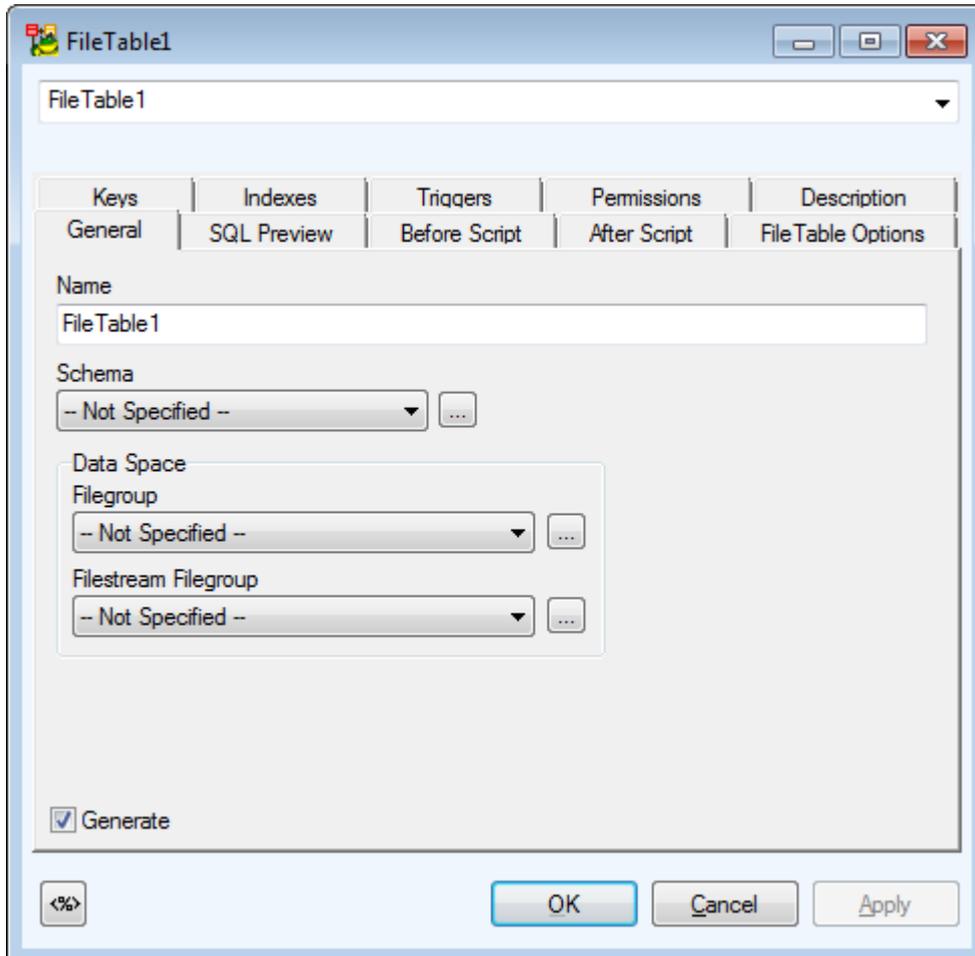
## Extended Properties

Extended Properties are supported for the following objects (generally where comments are supported):

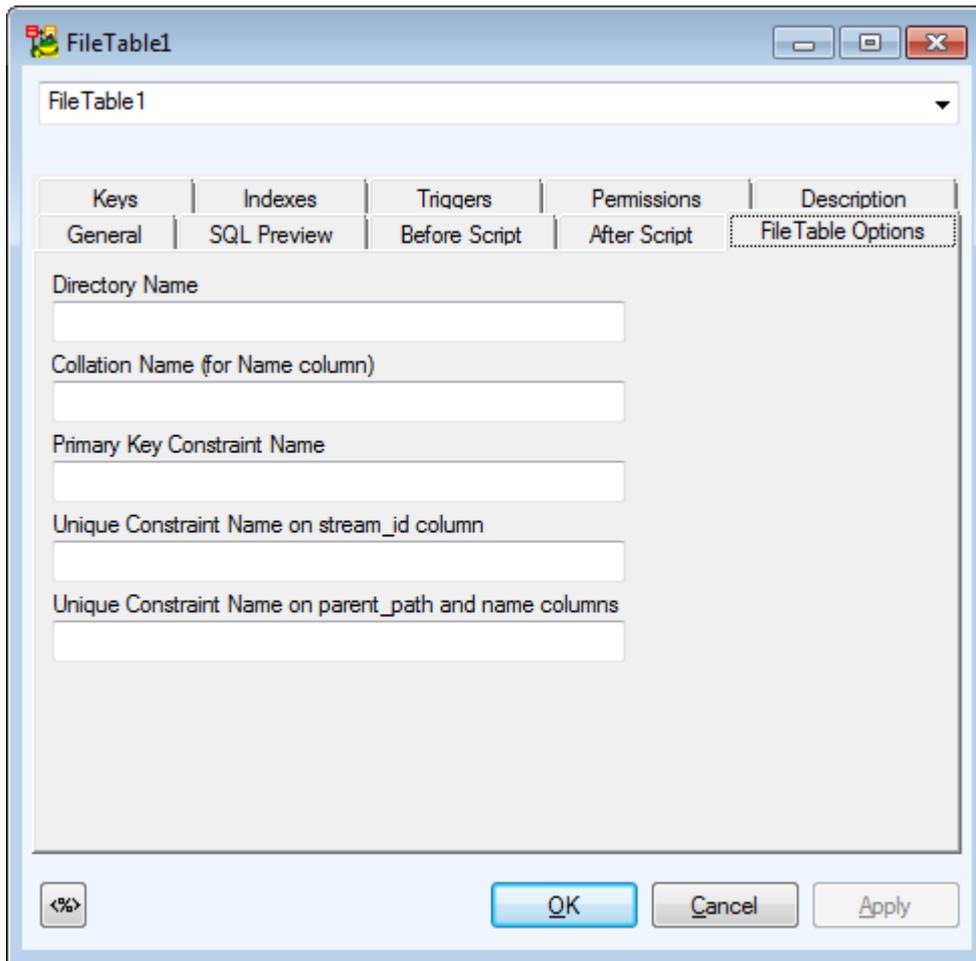
1. Objects with new tab **Extended Properties**.
2. Objects for which Extended Properties are loaded during reverse engineering as a text as for example After Script, function parameters, procedure parameters, view columns, key/index/foreign key for Filetable.

# File Table

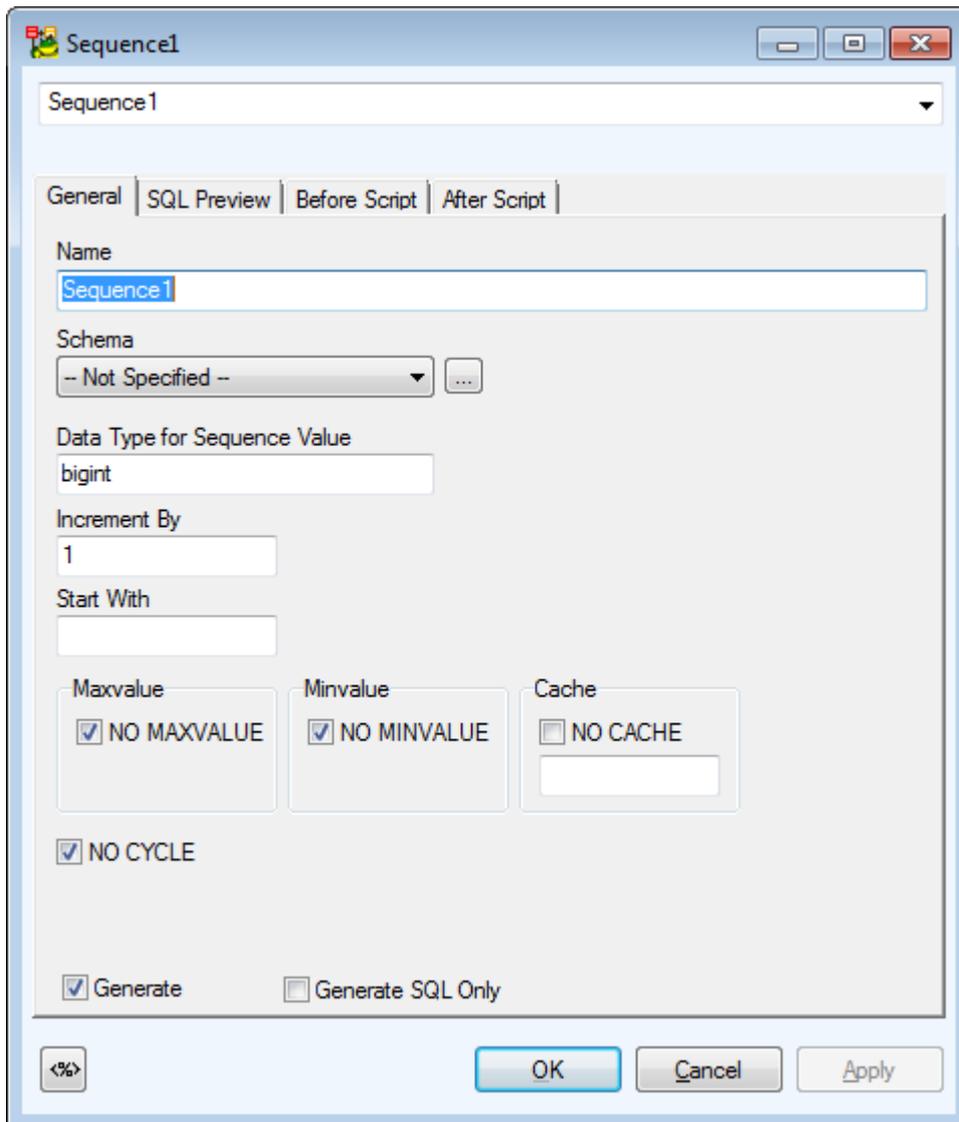
New **File Table** object is available.



- Keys can be defined in text on tab **Keys** (ALTER TABLE tablename ADD UNIQUE (attr)).
- Indexes can be defined in text on tab **Indexes** (CREATE INDEX ...).
- Triggers can be defined in text on tab **Triggers** (CREATE TRIGGER ...).
- Permissions can be defined in text on tab **Permissions** (GRANT ...).
- Comments can be defined on tab **Description**.
- In case the user wants to define checkconstraints in text, the **After Script** tab can be used.
- In case the user wants to define foreign keys in text, the **After Script** tab can be used.



# Sequence



Define Data Type in **Data Type for Sequence Value** box.

Available data types: *tinyint*, *smallint*, *int*, *bigint*, *decimal* and *numeric* with a scale of 0. You can use also any user-defined data type (alias type) that is based on one of the allowed types.

# Extended Properties

Extended Properties are supported for the following objects (generally where comments are supported):

1. Objects with new tab **Extended Properties**.
2. Objects for which Extended Properties are loaded during reverse engineering as a text as for example After Script, function parameters, procedure parameters, view columns, key/index/foreign key for Filetable.

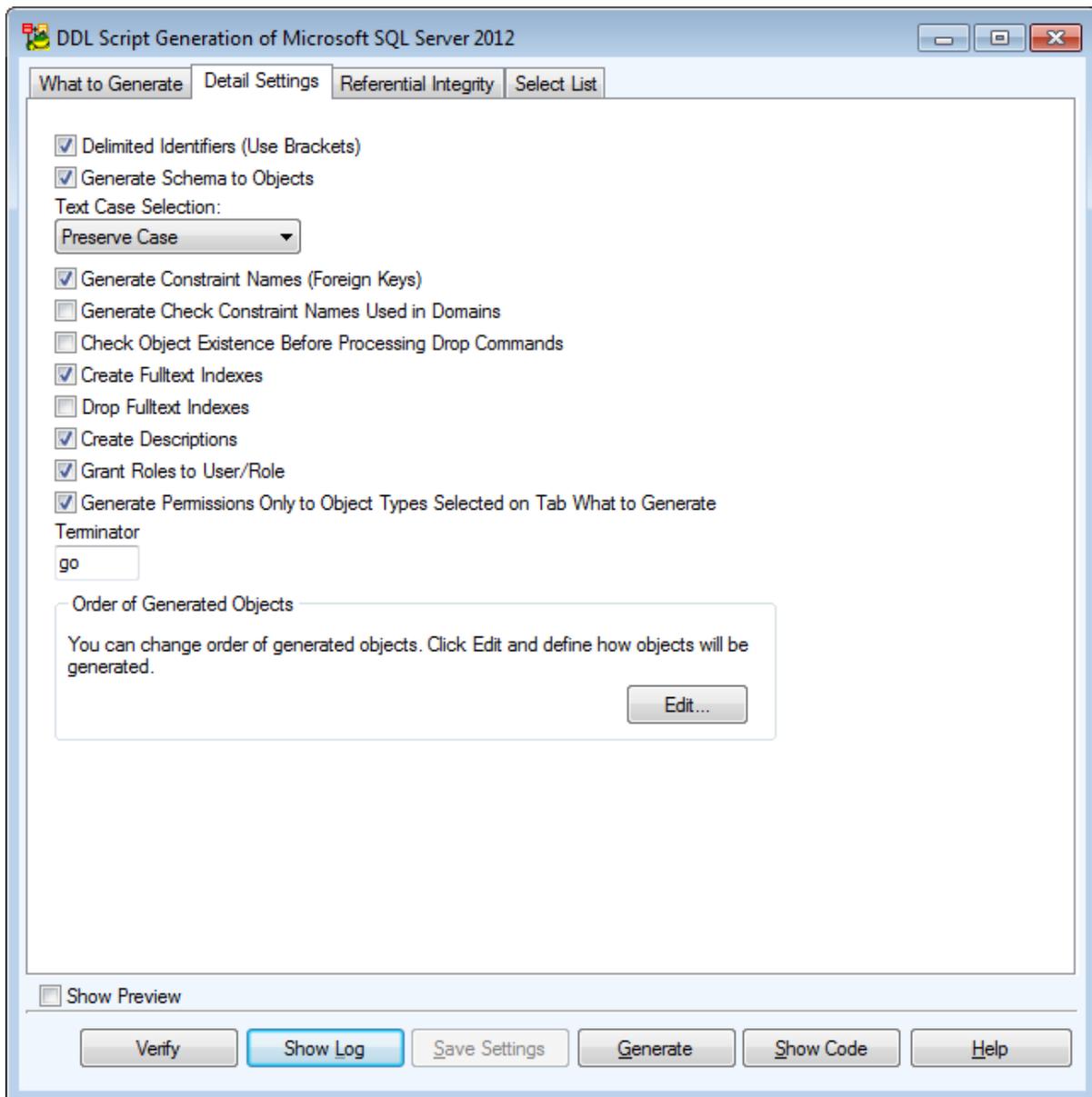
Other objects in Model Explorer:

- **Sequence**
- **Search Property List**
- **FileTable**
- **Filegroups**
- **Partition Schemes**
- **FilestreamFilegroups** - Similarly to Filegroups, FilestreamFilegroups work in Toad Data Modeler only as a list. During script generation, the FilestreamFilegroups definition is not generated and during reverse engineering the definition is not loaded either (only names are loaded).
- **Stoplists** - Similarly to Filegroups, Stoplists work in Toad Data Modeler only as a list. During script generation, the Stoplist definition is not generated and during reverse engineering the definition is not loaded either (only names are loaded).
- **Partition Functions**
- **Fulltext Catalogs**
- **XML Schema Collections**
- **Assemblies**

## Reverse Engineering - Microsoft SQL Server 2012

See [Reverse Engineering - Microsoft SQL Server 2022](#) for more information.

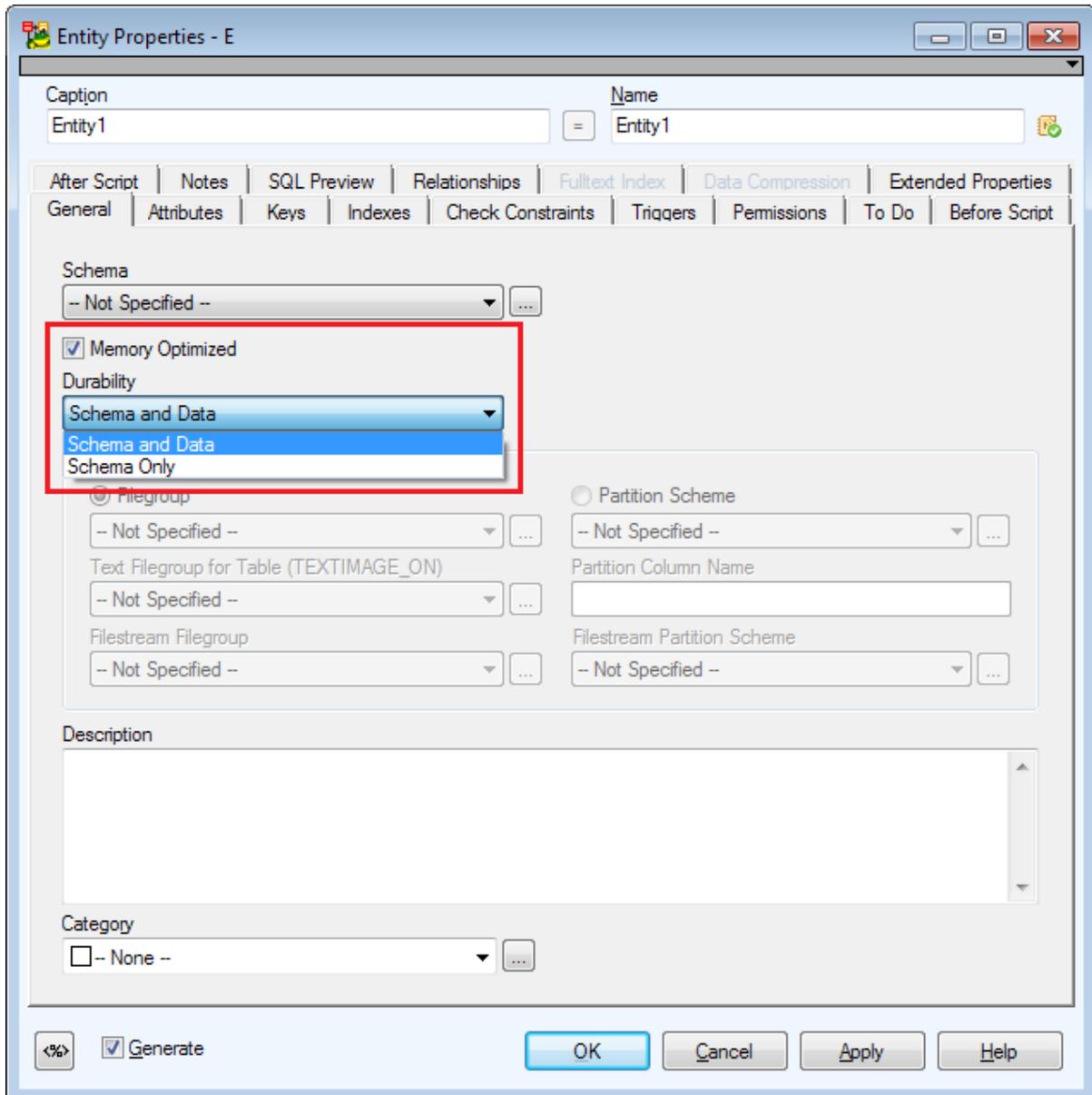
# Script Generation - Microsoft SQL Server 2012



**i** Note: To generate the table, column descriptions for SQL Server db in Toad Data Modeler, **schema/owner has to be defined**. Please see the **Entity Properties** form | **General** tab and from the **Schema** box select a schema or click the icon on the right to open the **Schema** dialog and define a new schema, then select it for the entity.

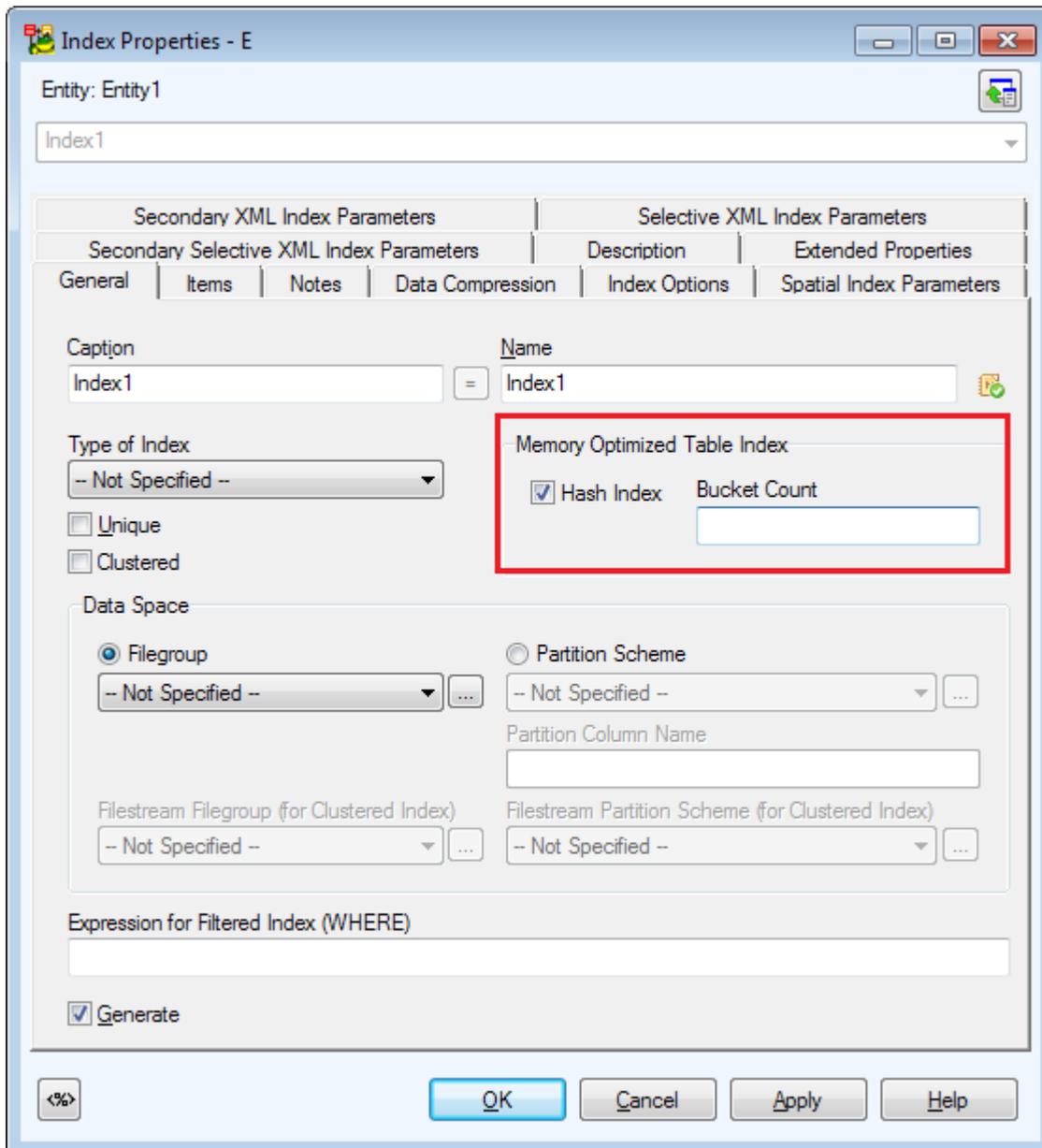
# Specifics - Microsoft SQL Server 2014

## Entity

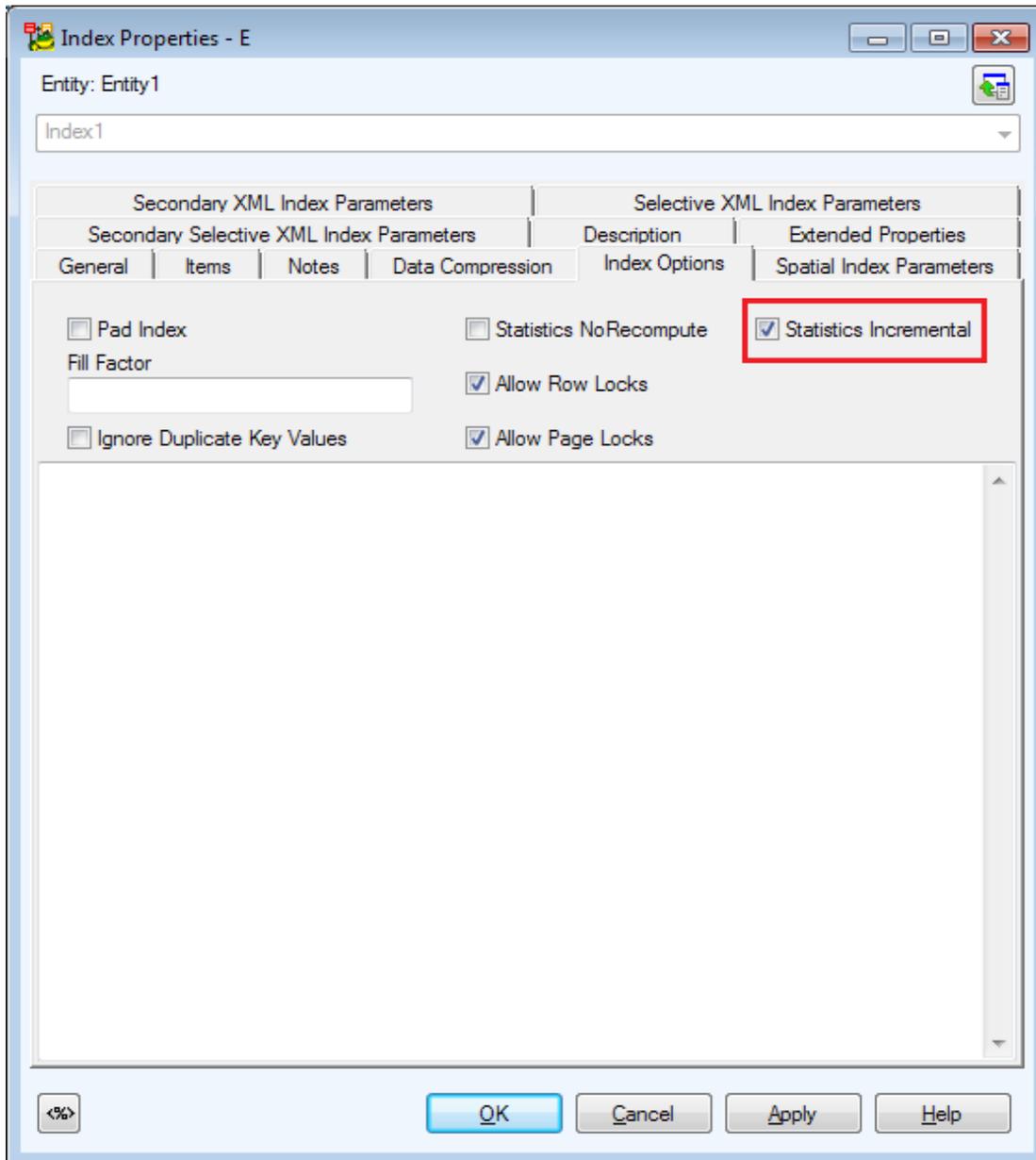


- New option to define Table as **Memory Optimized** using the new checkbox in **General** tab in **Entity Properties**.
- New option to select **Durability**. Only active when **Memory Optimized** is checked

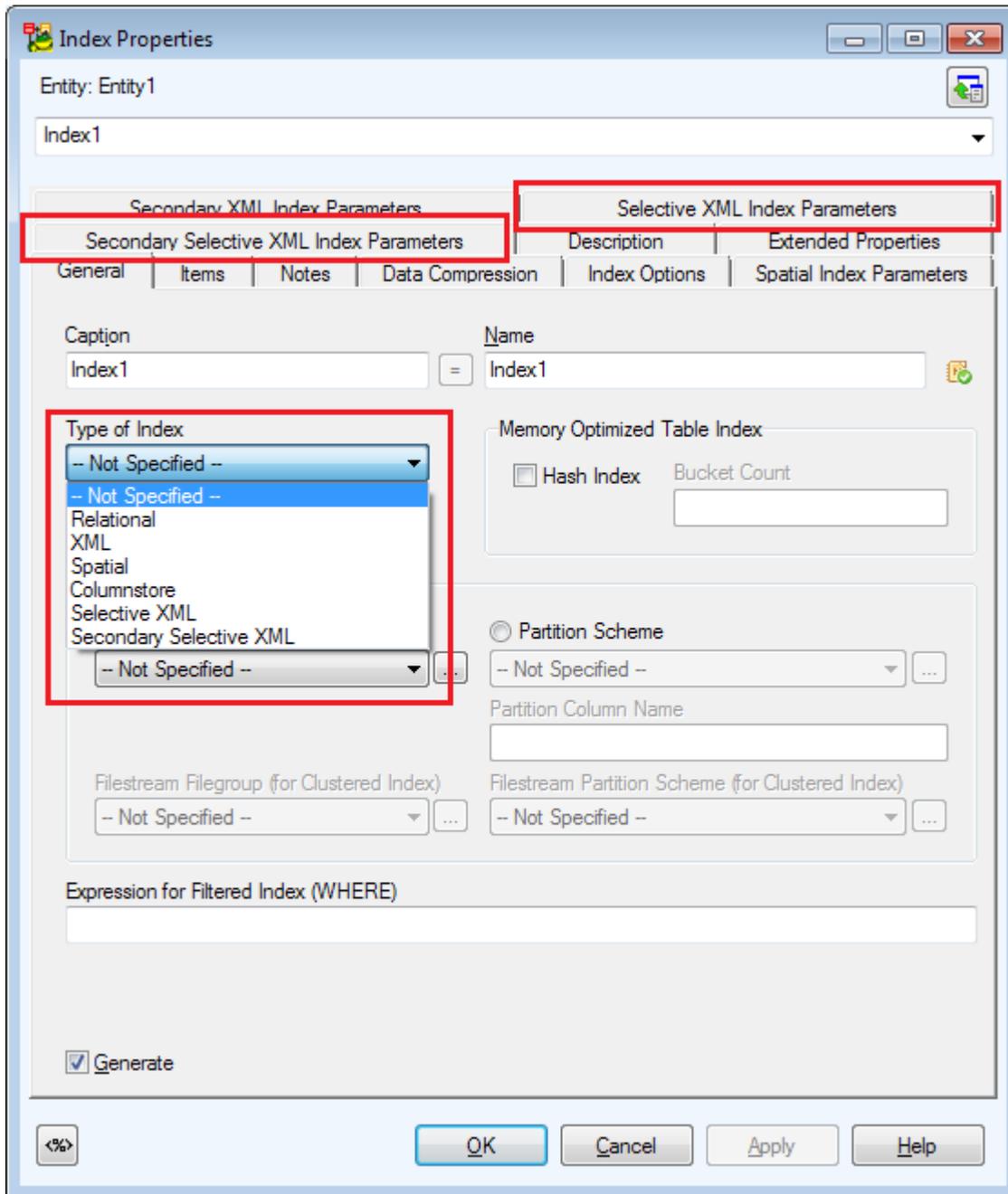
# Index



- New options available for indexes of **Memory Optimized** tables in **Index | General - Hash Index** and **Bucket Count** (only available with checked **Hash Index**)



- New option available in **Index | Index Options - Statistics Incremental**

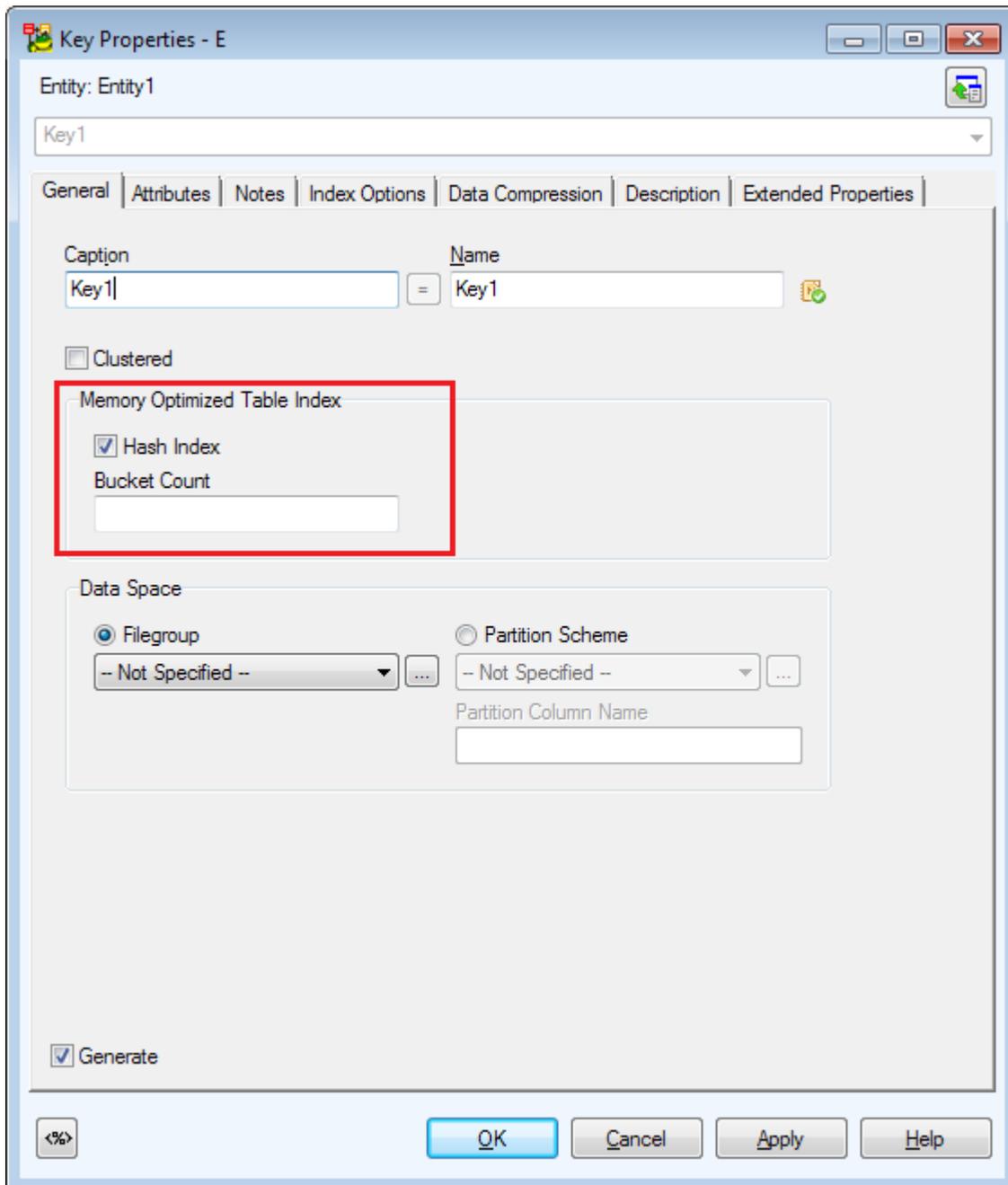


- Added new tabs **Selective XML** and **Selective Secondary XML**
- New Index types available in **Index | General | Type of Index** - **Columnstore**, **Selective XML**, **Secondary Selective XML**

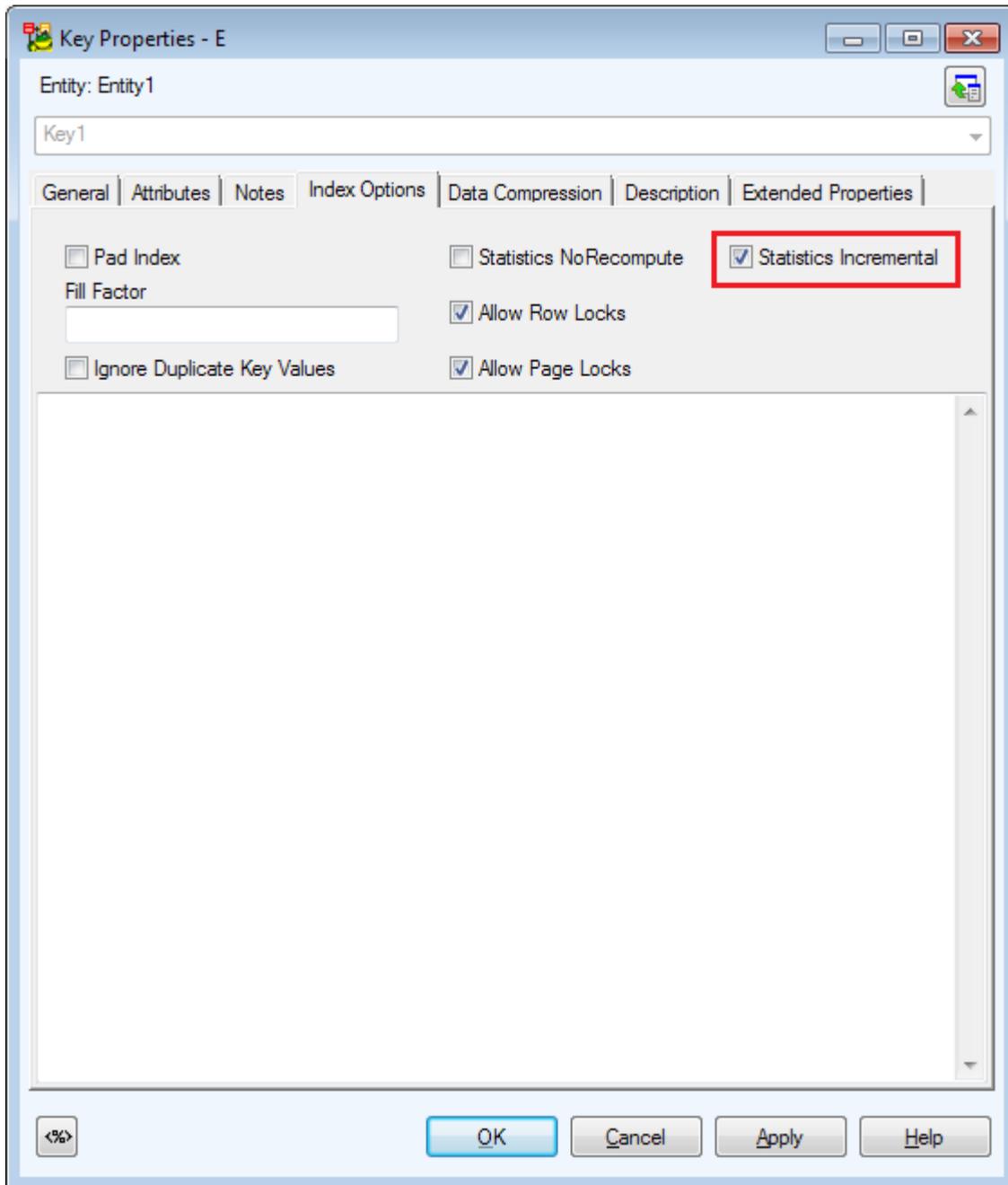
**i** Note: Choosing Columnstore type changes the choices available in **Compression Type** and **Compression Type for Partition** combo boxes in **Data Compression** tab. The specific options for Columnstore type are: COLUMNSTORE and COLUMNSTORE\_ARCHIVE.

**i** Note: You can now set Index as Clustered Columnstore by choosing Columnstore Index type and checking the Clustered checkbox.

# Key

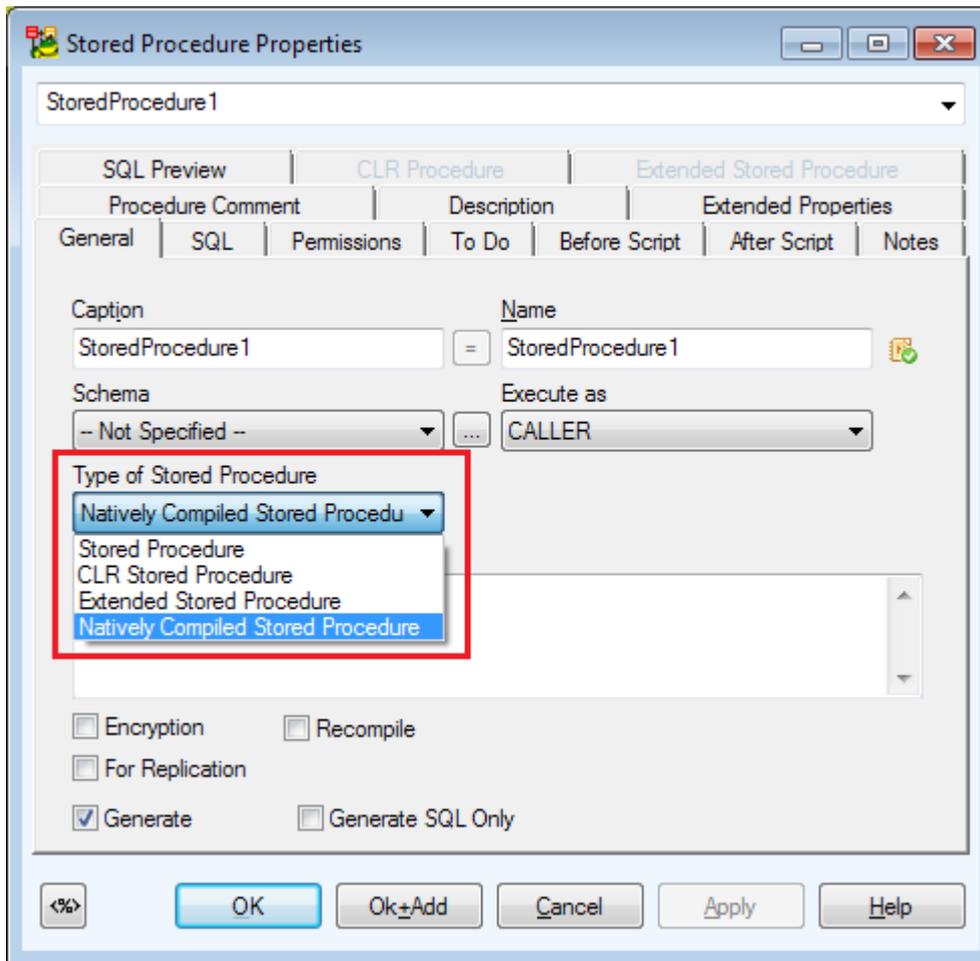


- New options available for keys of **Memory Optimized** tables in **Key | General - Hash Index** and **Bucket Count** (only available with checked **Hash Index**)



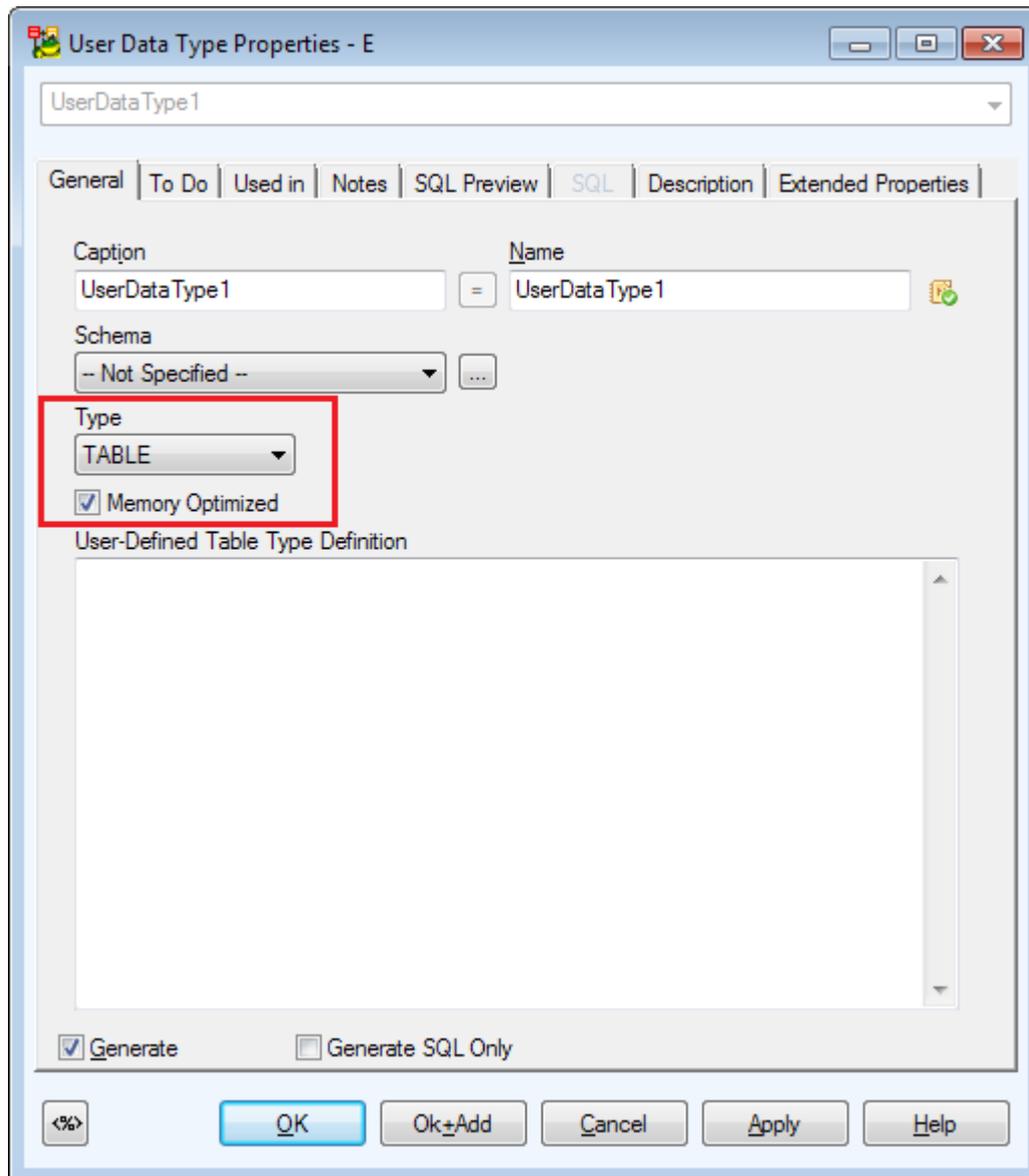
New option available in **Key | Index Options - Statistics Incremental**

# Procedure



- **New Type of Stored Procedure available in Stored Procedure Properties | General - Natively Compiled Stored Procedure**

# User Data Type

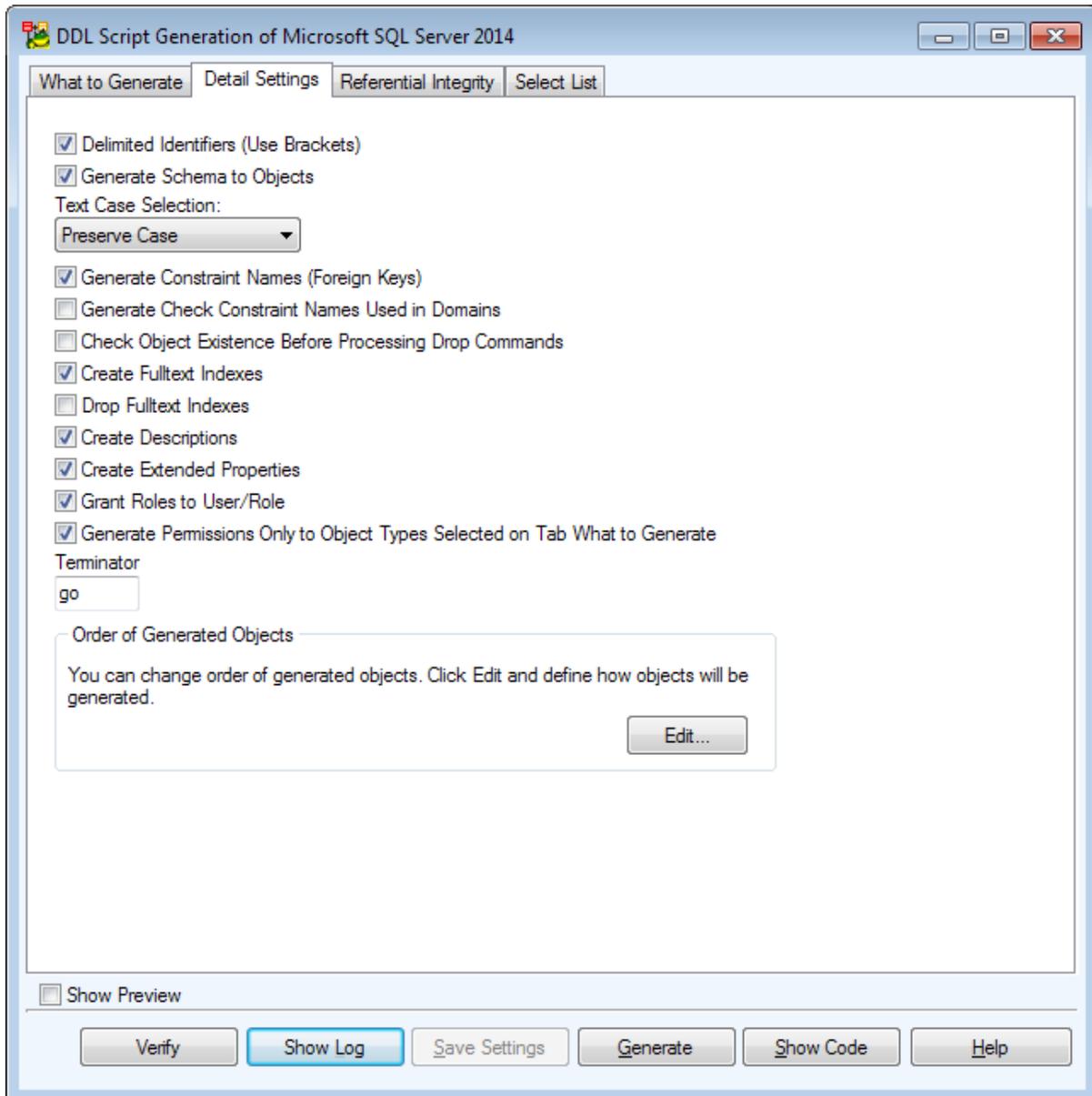


- New option to define User Data Type as **Memory Optimized** (User Data Type Properties | General), User Data Type has to be of TABLE type

## Reverse Engineering - Microsoft SQL Server 2014

See [Reverse Engineering - Microsoft SQL Server 2022](#) for more information.

# Script Generation - Microsoft SQL Server 2014



**i** Note: To generate the table, column descriptions for SQL Server db in Toad Data Modeler, **schema/owner has to be defined**. Please see the **Entity Properties** form | **General** tab and from the **Schema** box select a schema or click the icon on the right to open the **Schema** dialog and define a new schema, then select it for the entity.

# Specifics - Microsoft SQL Server 2016

**External Table** support (PolyBase external table that references data stored in Hadoop cluster or Azure blob storage). External Tables can be found in **Physical Model Explorer**.

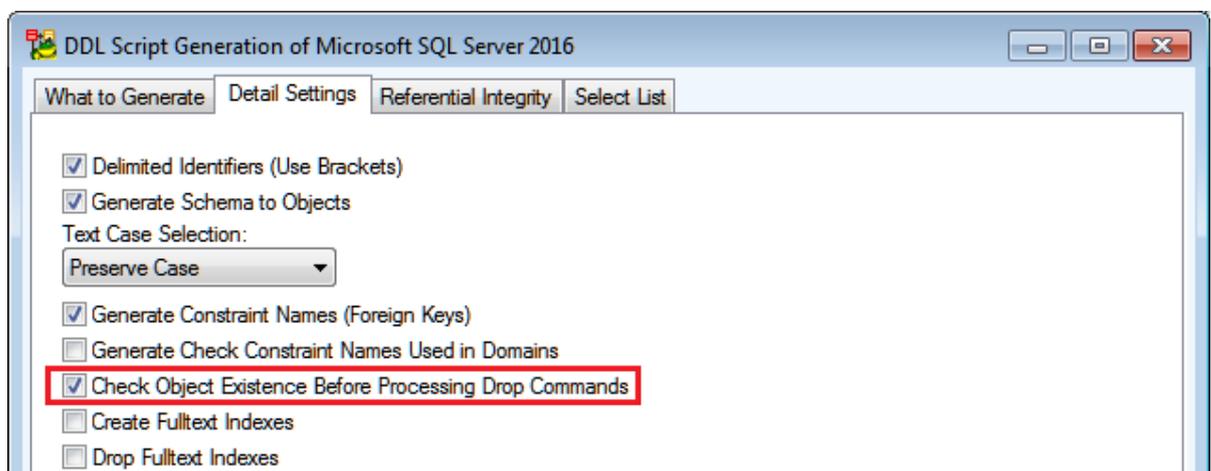
**Security Policy** support (for row-level security). Security Policies can be found in **Physical Model Explorer**.

New model objects with simplified support:

- **Column Encryption Key** - currently without any properties, used for **Encrypting Columns**.
- **External Data Source** - currently without any properties, used for **External Table**.
- **External File Format** - currently without any properties, used for **External Table**.

*IF EXISTS* support for *DROP* commands of certain object types:

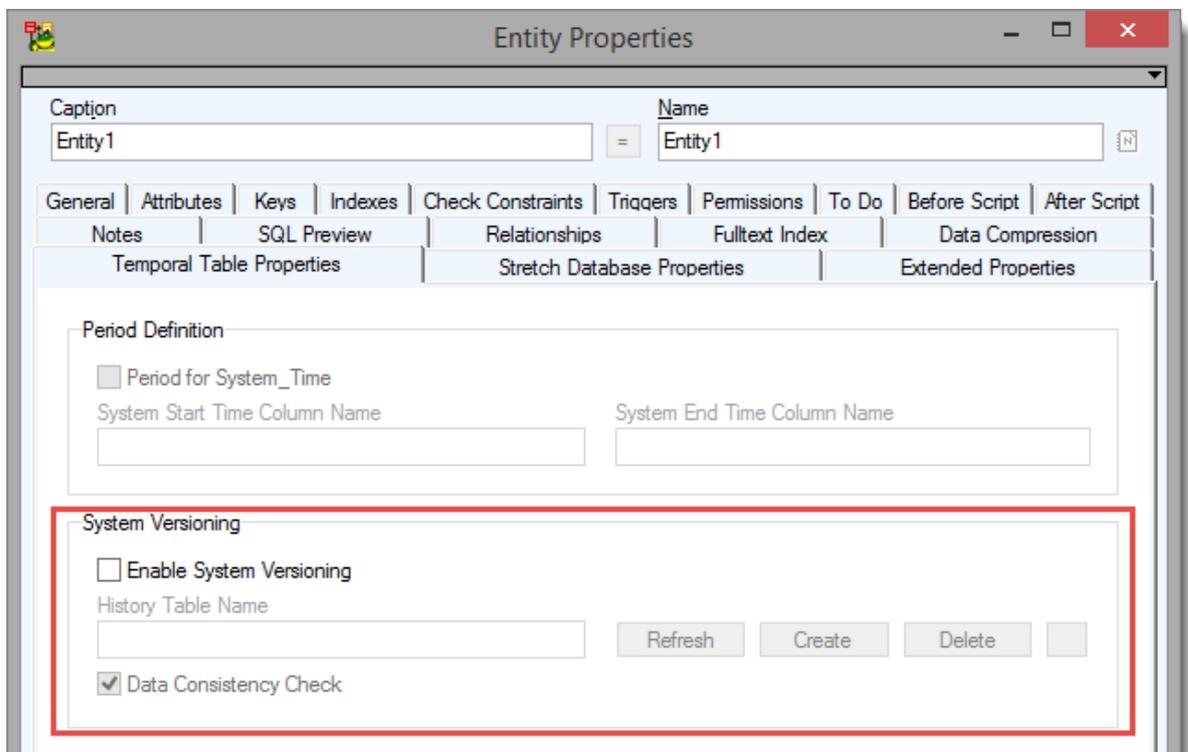
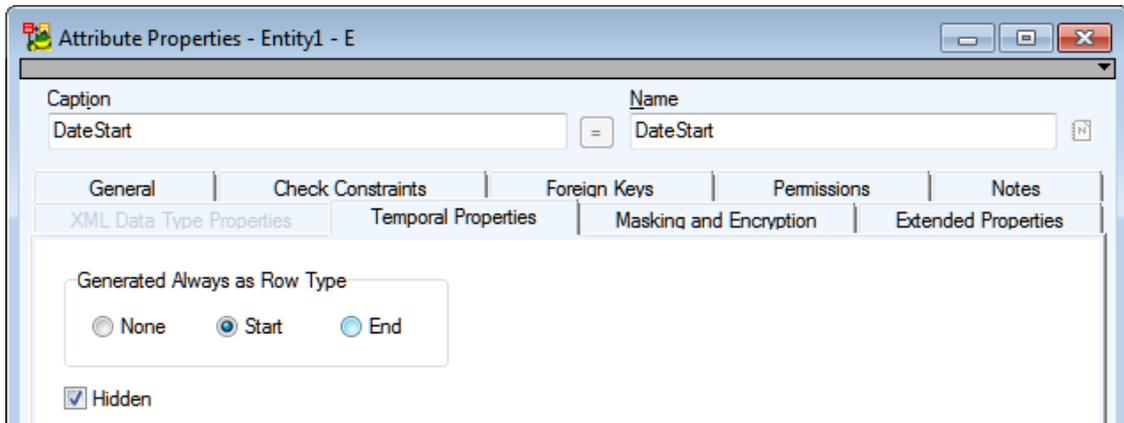
- Entity, View, Relation, Trigger (entity/view), Key, Index, Default, Rule, DictType, UDT, Procedure, Function, Aggregate (Functions), Synonym, Assembly, Sequence, FileTable
- To use the clause, enable the **Check Object Existence Before Processing Drop Commands** option in **DDL Script Generation | Detail Settings**.



## Entity

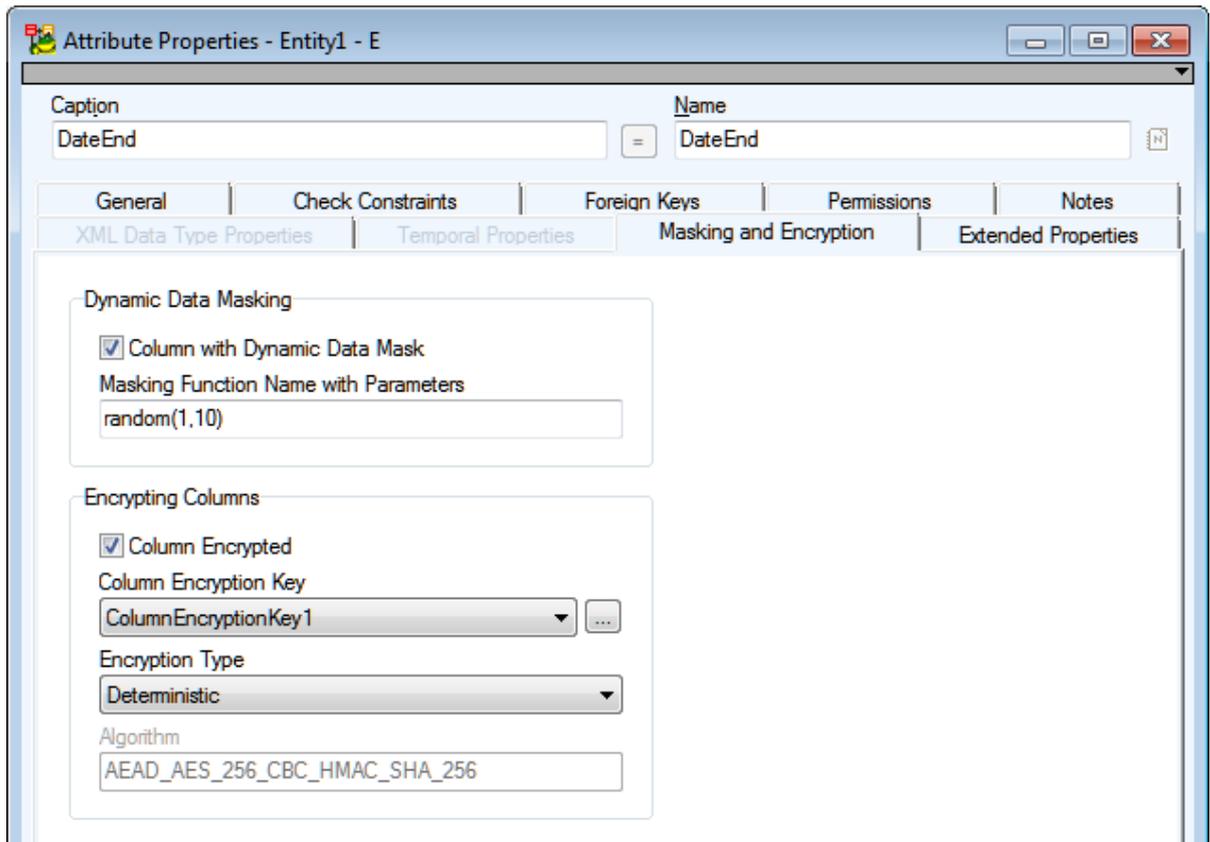
- **Memory Optimized Tables:**
  - It is now possible to create the following objects in memory optimized tables:
    - **Triggers (Natively Compiled only)**
    - **Index (Clustered Columnstore only)**
    - **Foreign Key constraints** (between memory optimized Tables)
    - **Check constraints**
    - **Unique constraints**

- "Period for System\_Time" and "System-Versioned Temporal Table" support for Tables (Attribute Properties | Temporal Properties tab AND Entity Properties | Temporal Table Properties tab)



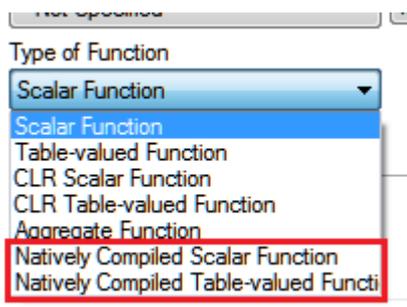
## Attribute

- **Dynamic Data Masking** support for Attributes (Attribute Properties | Masking and Encryption tab | Dynamic Data Masking)
- **Always Encrypted** support for Attributes (Attribute Properties | Masking and Encryption tab | Encrypting Columns)



## Function

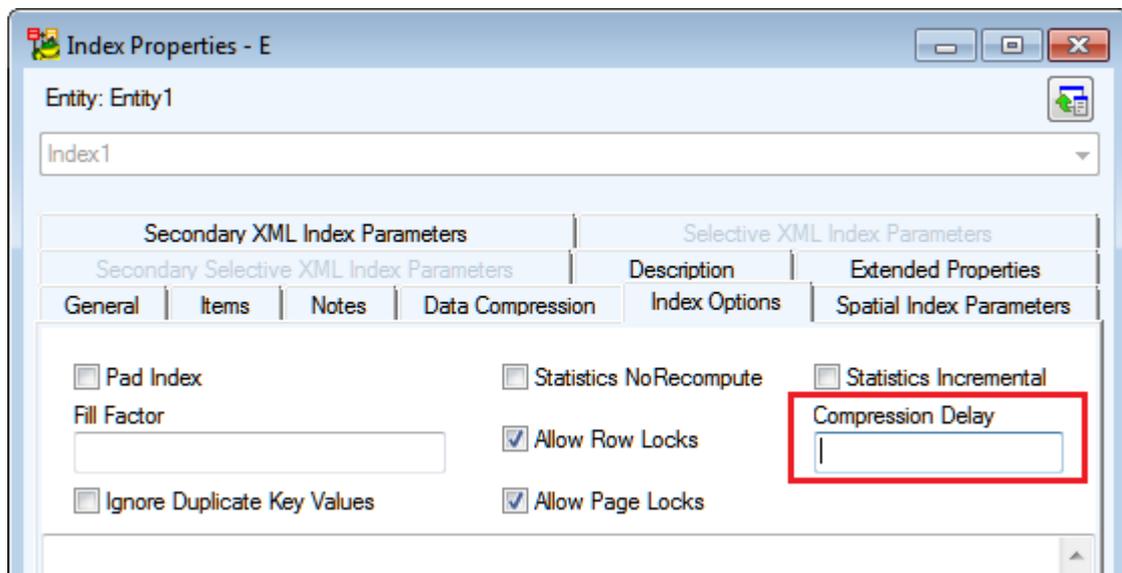
- New **Function** types:
  - **Natively Compiled Scalar Function**
  - **Natively Compiled Inline Table-valued Function**



- *EXECUTE AS CALLER* for **Natively Compiled Stored Procedure Functions** (used by default).

# Index

- New parameter - *COMPRESSION\_DELAY* for *COLUMNSTORE* **Indexes**.



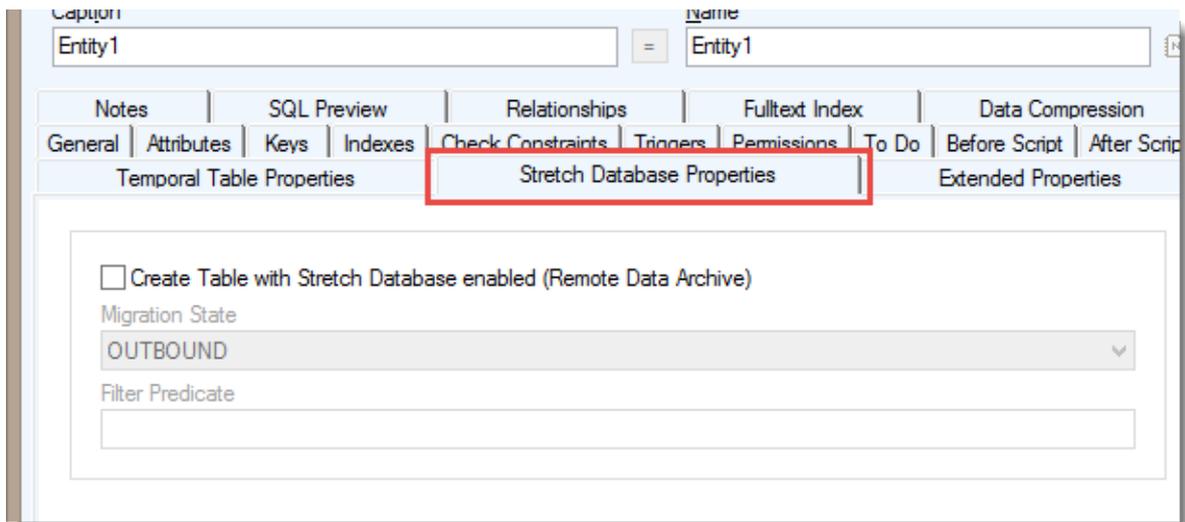
- Filter predicates (WHERE) can be now used in non-clustered *COLUMNSTORE* Indexes.
- Non-clustered Indexes can be now created even when *CLUSTERED COLUMNSTORE* Indexes already exist.

# User

- *ALLOW\_ENCRYPTED\_VALUE\_MODIFICATIONS = ON | OFF* support for **Users** (in Reverse Engineering).

# Misc.

- Support for Stretch Database (*REMOTE\_DATA\_ARCHIVE*) in SQL Server 2016
- To create a Table for remote data archive see **Entity Properties | Stretch Database Properties**

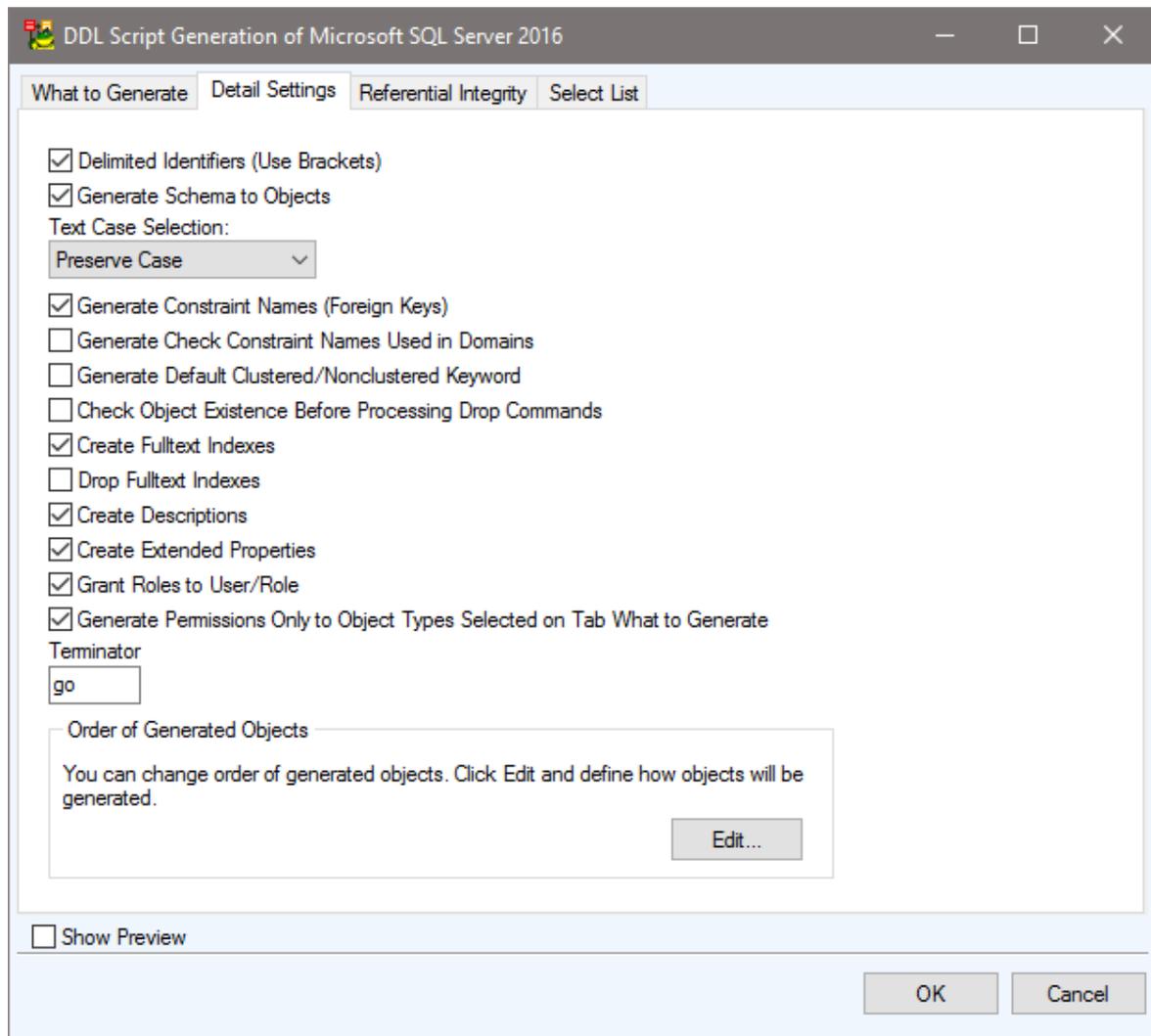


**i** NOTE: REMOTE\_DATA\_ARCHIVE is not supported for Memory Optimized Tables.

## Reverse Engineering - Microsoft SQL Server 2016

See [Reverse Engineering - Microsoft SQL Server 2022](#) for more information.

# Script Generation - Microsoft SQL Server 2016

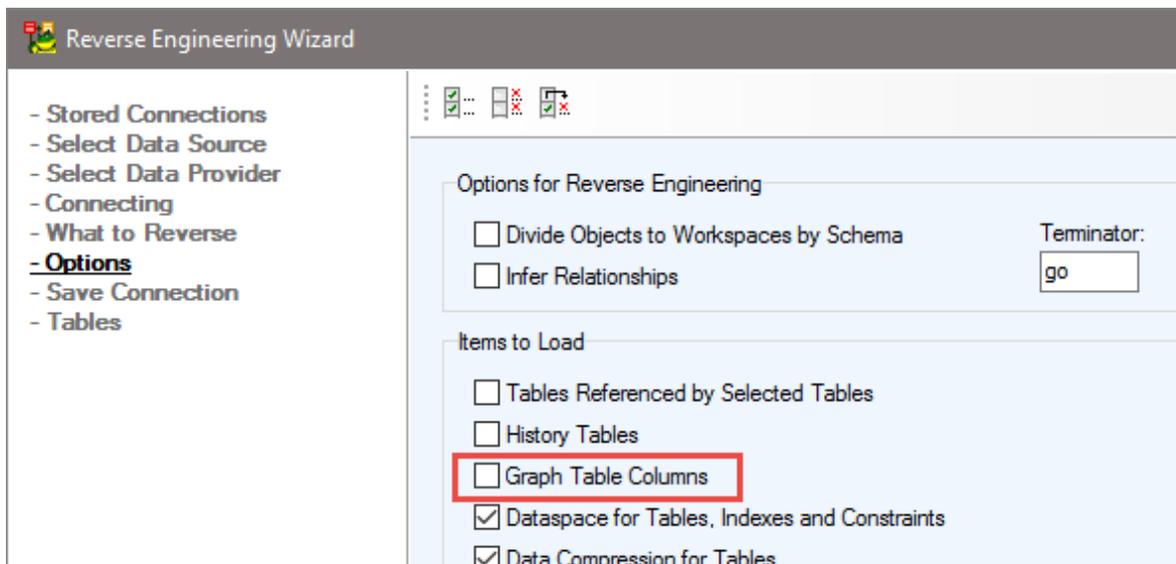
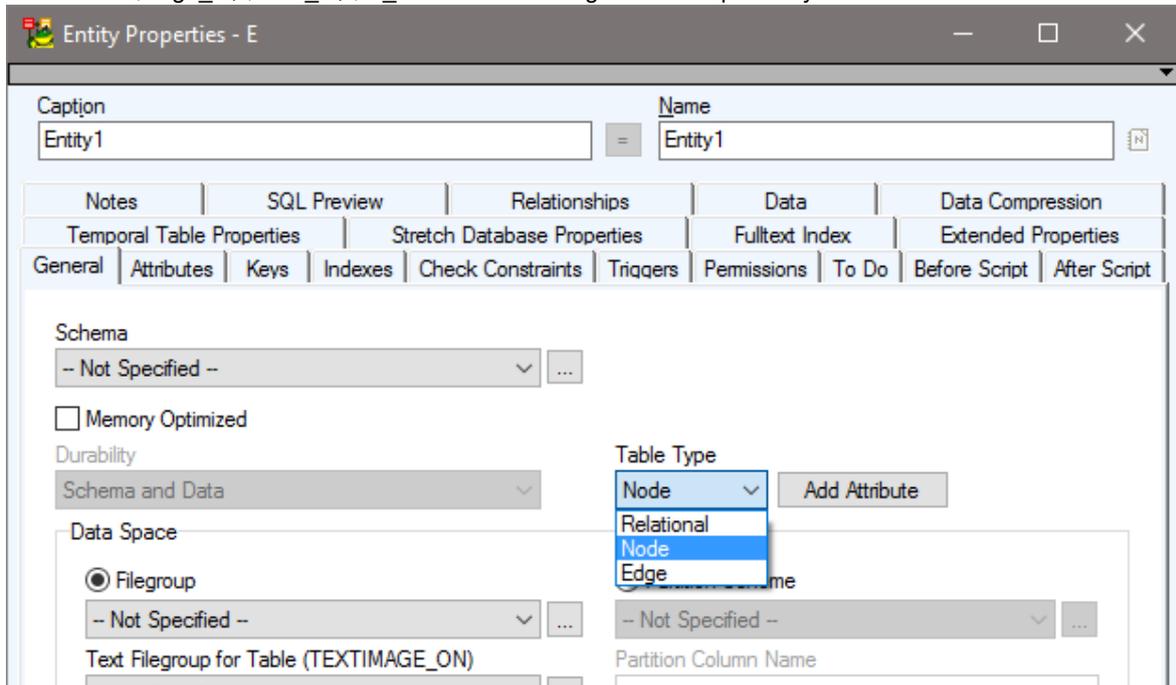


**i** Note: To generate the table, column descriptions for SQL Server db in Toad Data Modeler, **schema/owner has to be defined**. Please see the **Entity Properties** form | **General** tab and from the **Schema** box select a schema or click the icon on the right to open the **Schema** dialog and define a new schema, then select it for the entity.

# Specifics - Microsoft SQL Server 2017

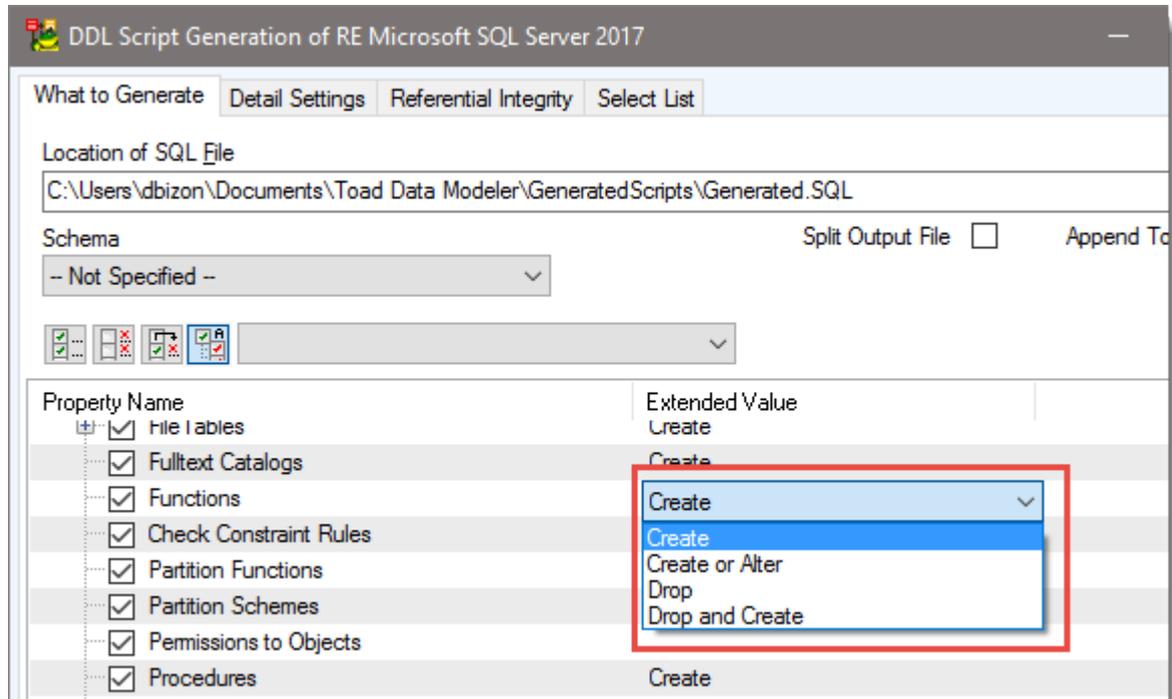
## Entity

- New node and edge (graph) tables are now supported
- Click **Add Attribute** to add graph table specific attributes. In **Attributes**, these will be called using \$node\_id and \$edge\_id, \$from\_id, \$to\_id for node and edge tables respectively



# Details

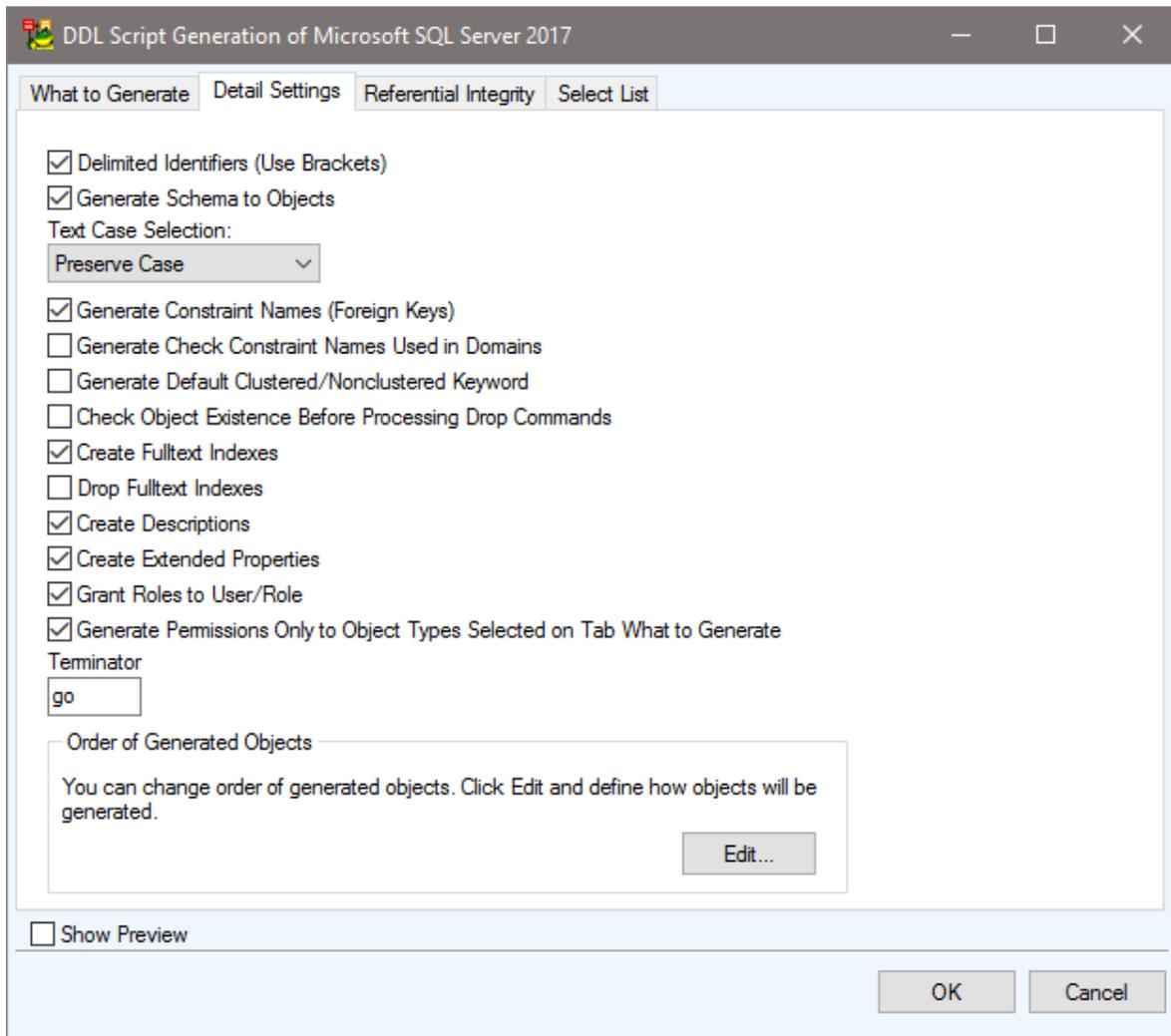
- CREATE EXTERNAL LIBRARY: External libraries are not supported
- CREATE OR ALTER: this clause is now supported for procedures, views, functions, and triggers (with exception for Change Script Generation where only Alter or Drop and Create are used)



## Reverse Engineering - Microsoft SQL Server 2017

See [Reverse Engineering - Microsoft SQL Server 2022](#) for more information.

# Script Generation - Microsoft SQL Server



**i** Note: To generate the table, column descriptions for SQL Server db in Toad Data Modeler, **schema/owner has to be defined**. Please see the **Entity Properties** form | **General** tab and from the **Schema** box select a schema or click the icon on the right to open the **Schema** dialog and define a new schema, then select it for the entity.

# Specifics - Microsoft SQL Server 2019

## Indexes

- New properties Online and Resumable are available in **Index Properties | Online Options**. The default values are `Default`.

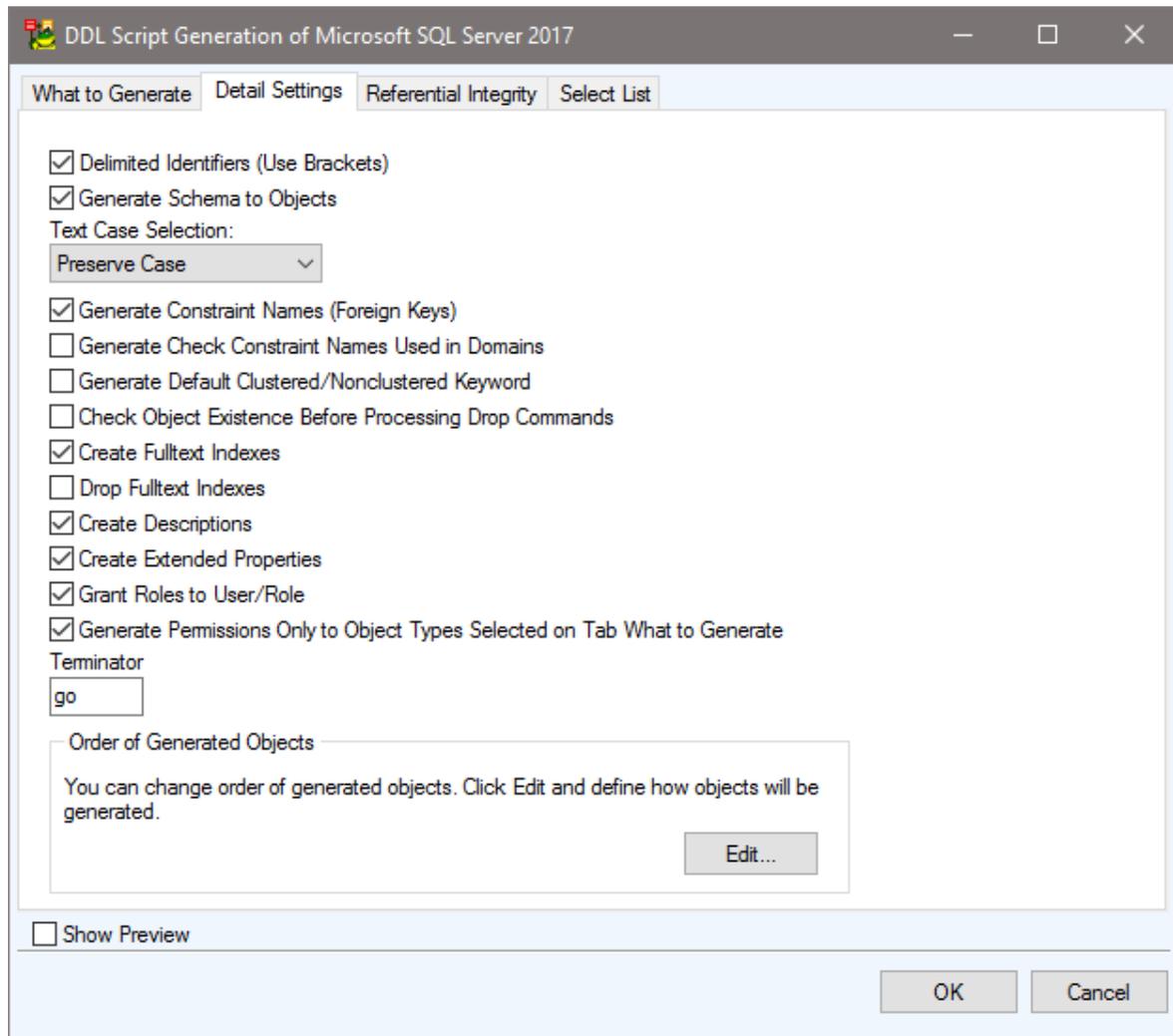
## Functions

- New Inline property is available in **Function Properties | General | Inline**. The default value is `Default`.

## Reverse Engineering - Microsoft SQL Server 2019

See [Reverse Engineering - Microsoft SQL Server 2022](#) for more information.

# Script Generation - Microsoft SQL Server

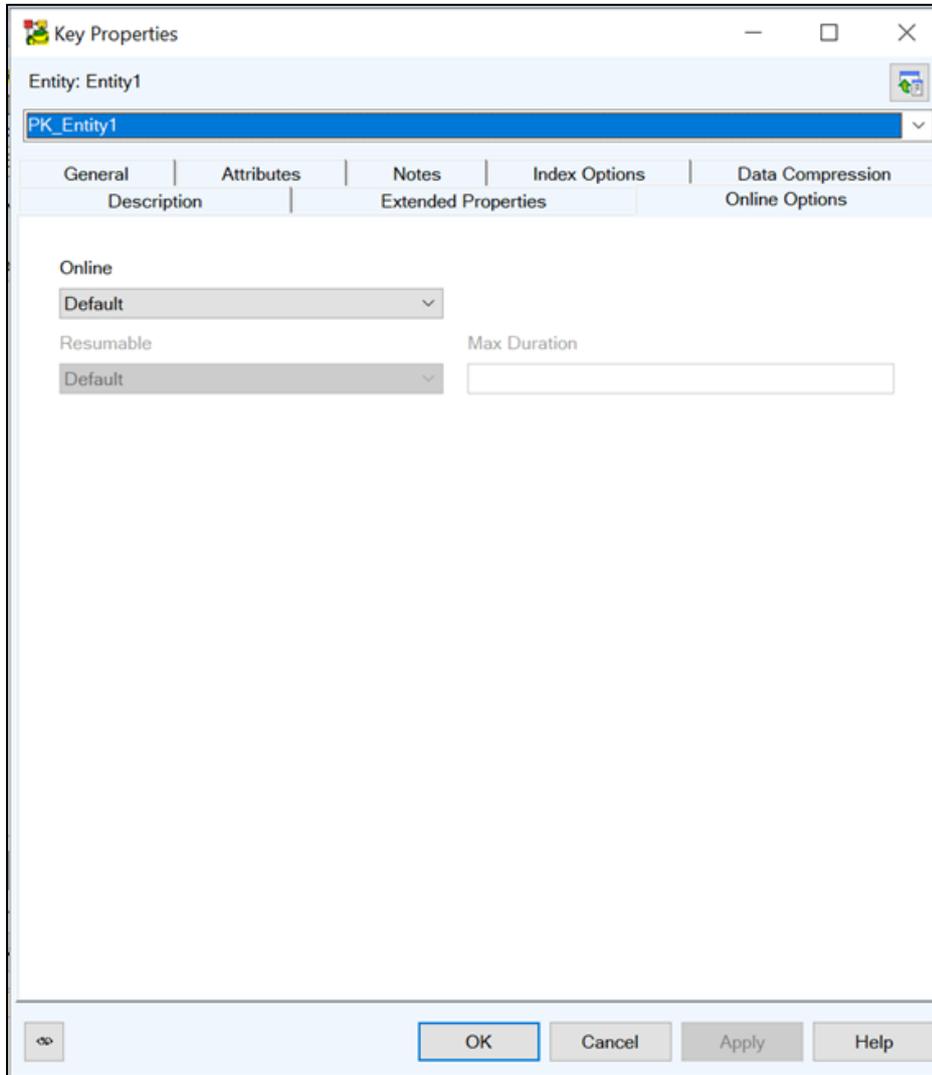


**i** Note: To generate the table, column descriptions for SQL Server db in Toad Data Modeler, **schema/owner has to be defined**. Please see the **Entity Properties** form | **General** tab and from the **Schema** box select a schema or click the icon on the right to open the **Schema** dialog and define a new schema, then select it for the entity.

# Specifics - Microsoft SQL Server 2022

## Keys

New properties Online and Resumable are available in **Keys Properties | Online Options**. The default values are **Default**.

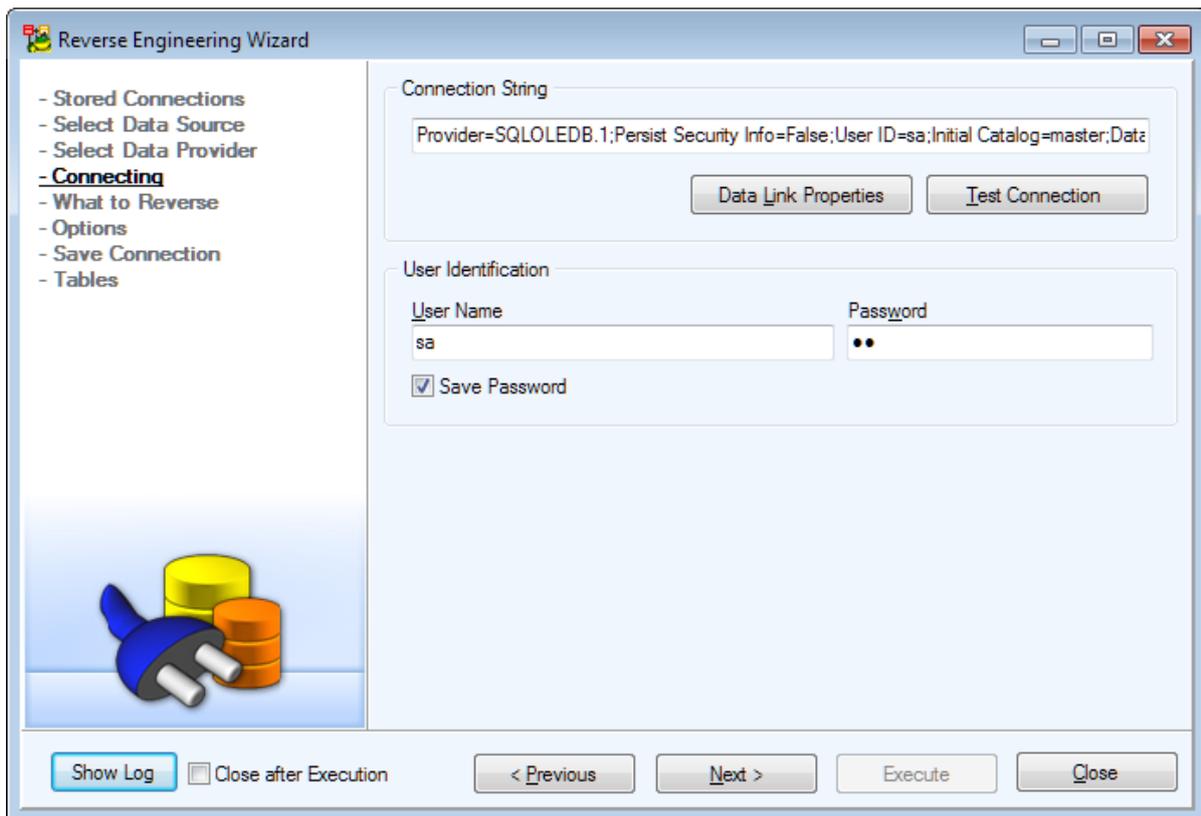


# Reverse Engineering - Microsoft SQL Server 2022

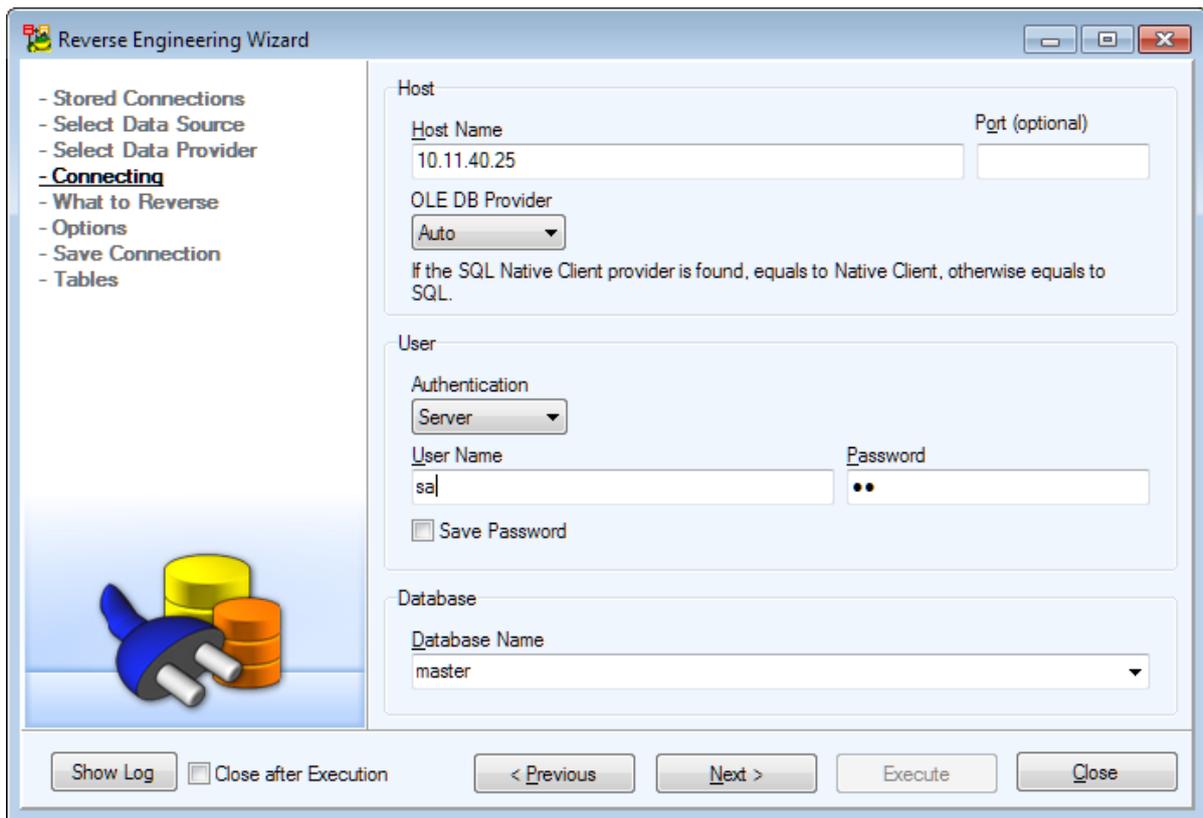
Available **Data Providers** are:

- **Connection via ADO**
- **Native Connection**

**Connection via ADO:**



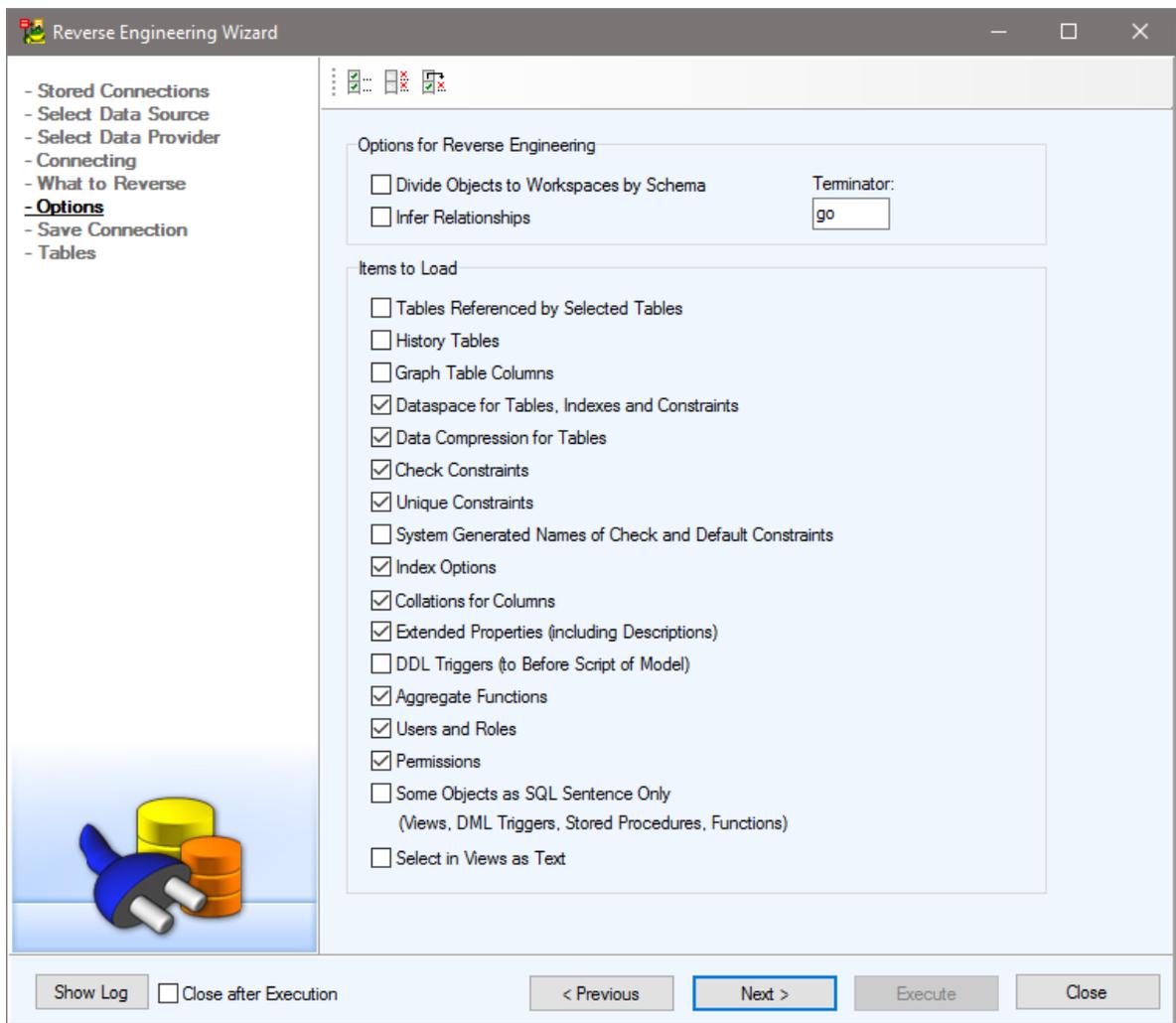
### Native Connection:



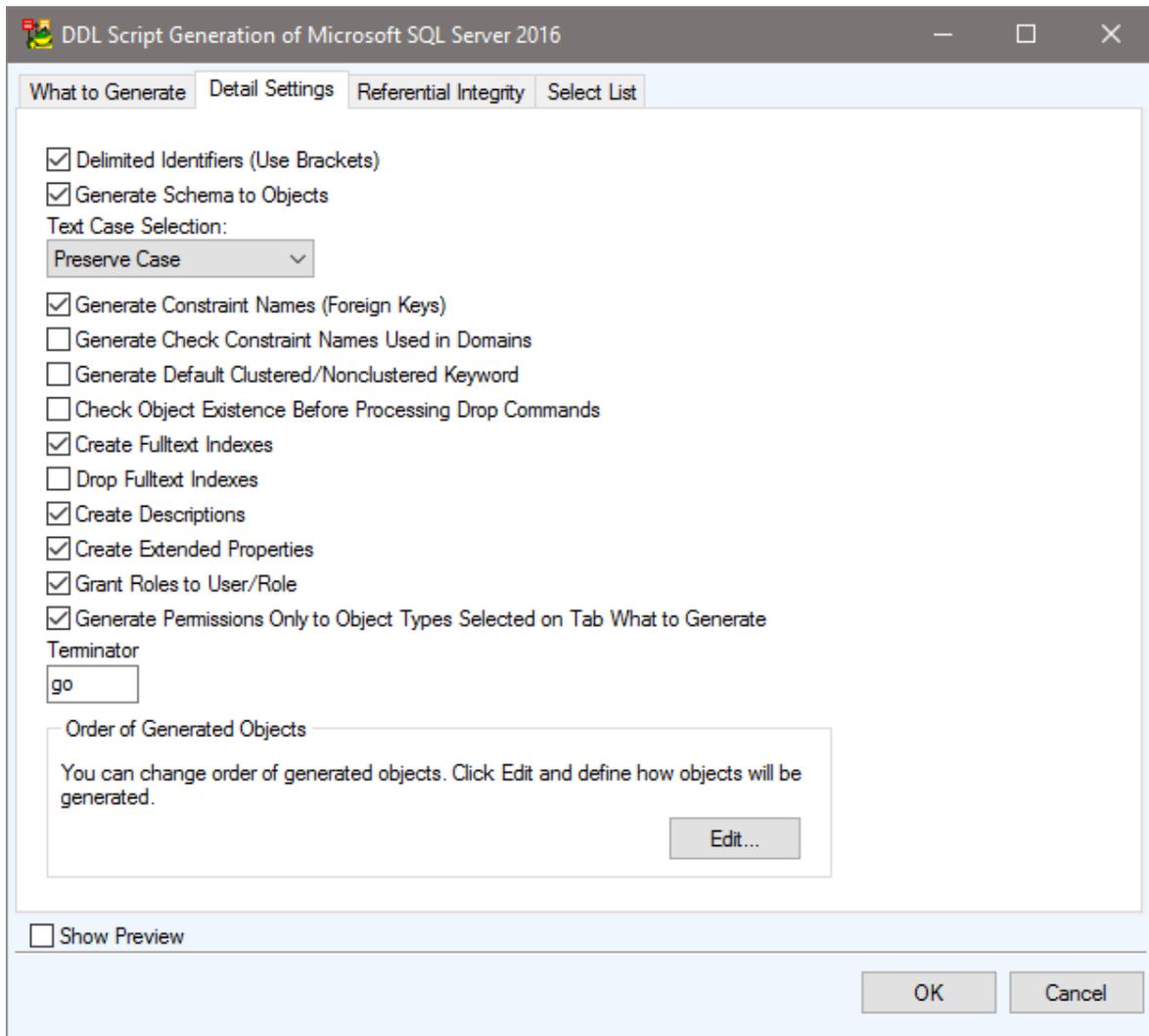
The screenshot shows the 'Reverse Engineering Wizard' dialog box with the 'Native Connection' configuration. The window title is 'Reverse Engineering Wizard'. On the left, a navigation pane lists steps: '- Stored Connections', '- Select Data Source', '- Select Data Provider', '- **Connecting**', '- What to Reverse', '- Options', '- Save Connection', and '- Tables'. Below the list is an icon of a blue network plug and three database cylinders (yellow, orange, and red). The main area is divided into sections: 'Host' with 'Host Name' (10.11.40.25) and 'Port (optional)' (empty); 'OLE DB Provider' (Auto) with a note: 'If the SQL Native Client provider is found, equals to Native Client, otherwise equals to SQL.'; 'User' with 'Authentication' (Server), 'User Name' (sa), 'Password' (masked with dots), and a 'Save Password' checkbox; and 'Database' with 'Database Name' (master). At the bottom, there are buttons for 'Show Log', 'Close after Execution' (checkbox), '< Previous', 'Next >', 'Execute', and 'Close'.

## Reverse Engineering Settings

- Check **Graph Table Columns** to include graph table columns \$node\_id", "\$edge\_id", "\$from\_id", "\$to\_id in your RE



# Script Generation - Microsoft SQL Server

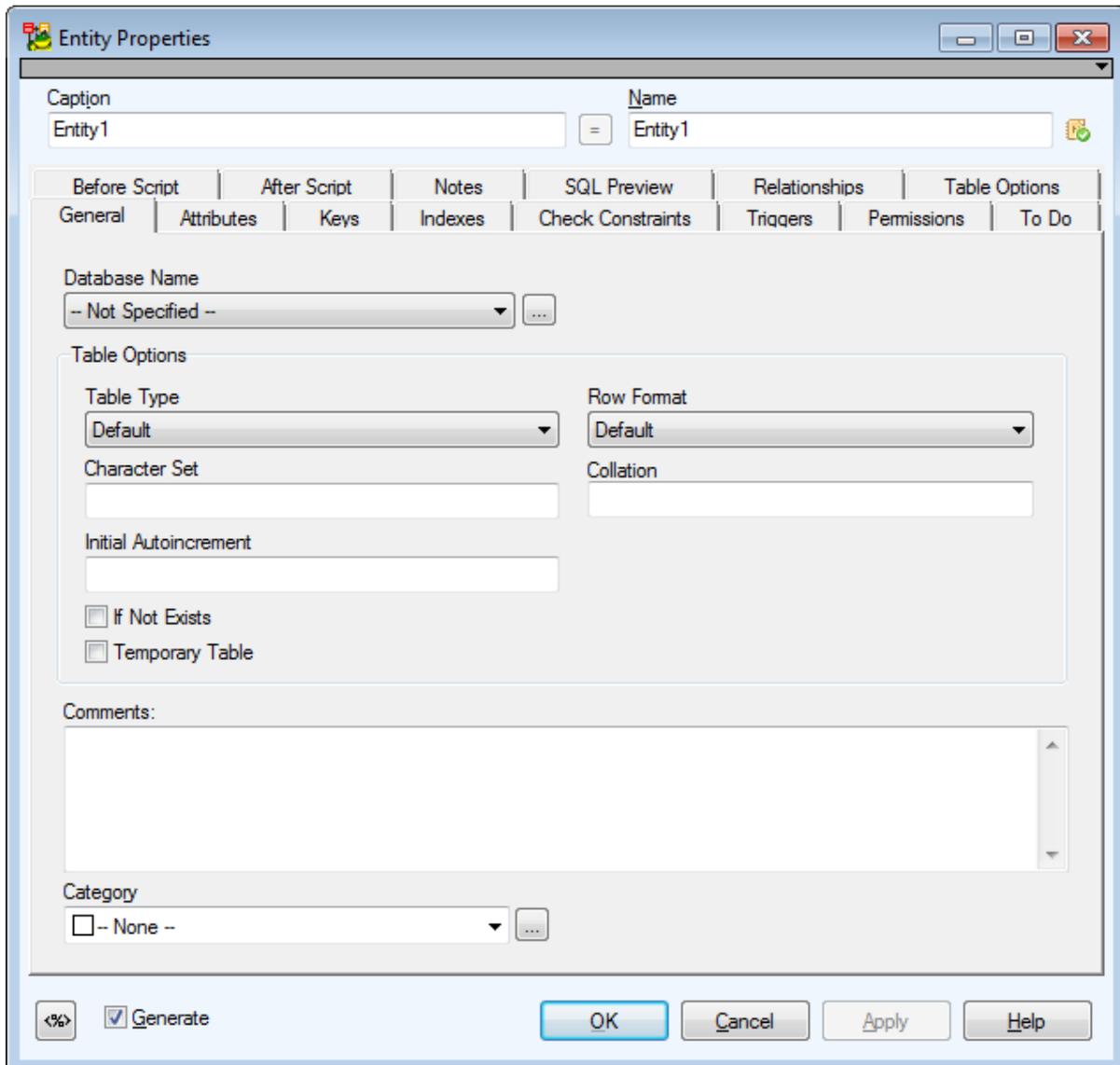


**i** Note: To generate the table, column descriptions for SQL Server db in Toad Data Modeler, **schema/owner has to be defined**. Please see the **Entity Properties** form | **General** tab and from the **Schema** box select a schema or click the icon on the right to open the **Schema** dialog and define a new schema, then select it for the entity.

# Specifics - MySQL 5.6

## Entity

### Row Format



For **Row Format** item, it's possible to select some of these options:

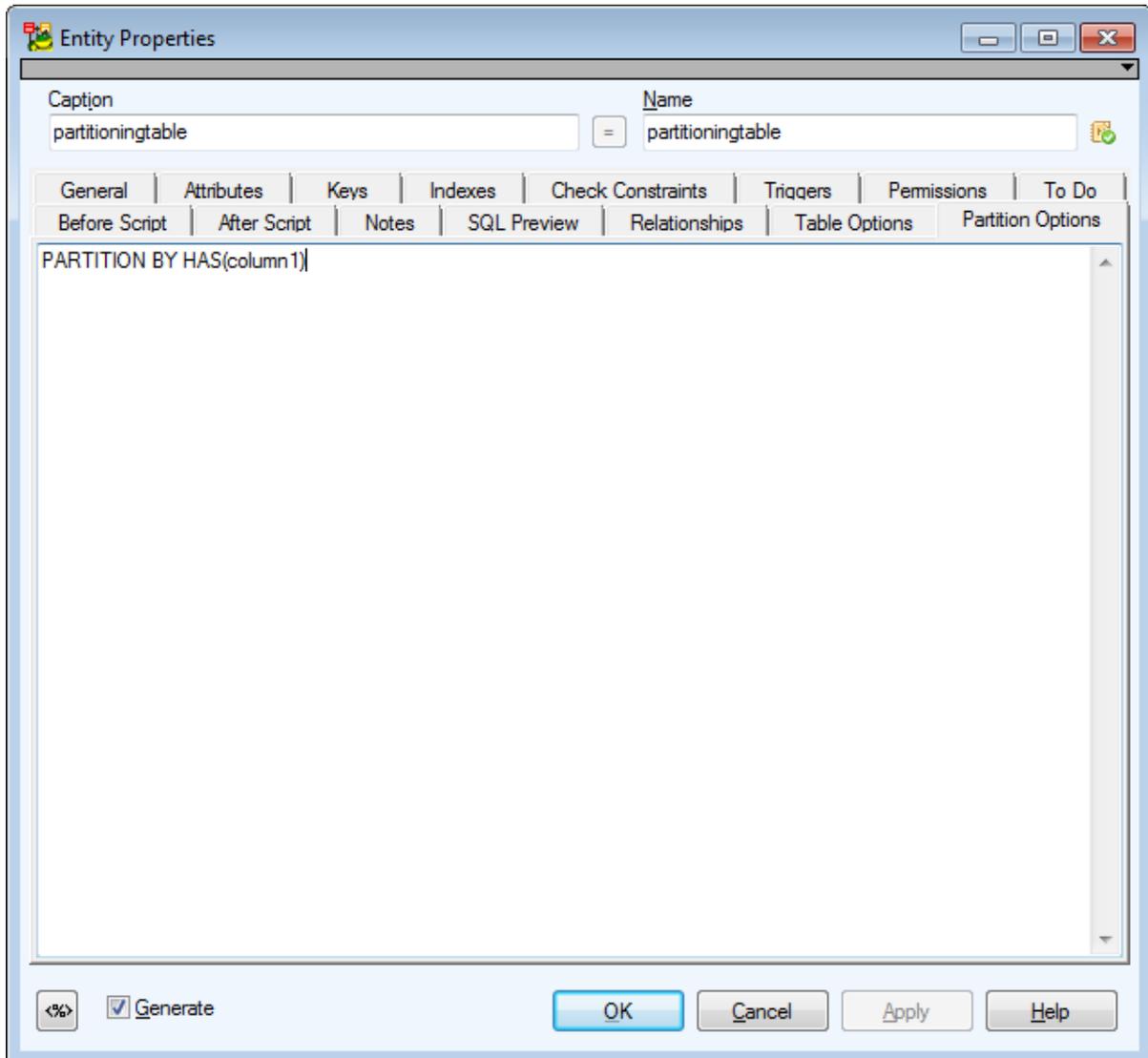
- Default
- Dynamic

- Fixed
- Compressed
- Redundant
- Compact

Properties **Character Set** and **Collation** must be identical for parent and child tables provided that at least one parent attribute is of a text data type. If the properties are not identical, a warning message is returned during model verification.

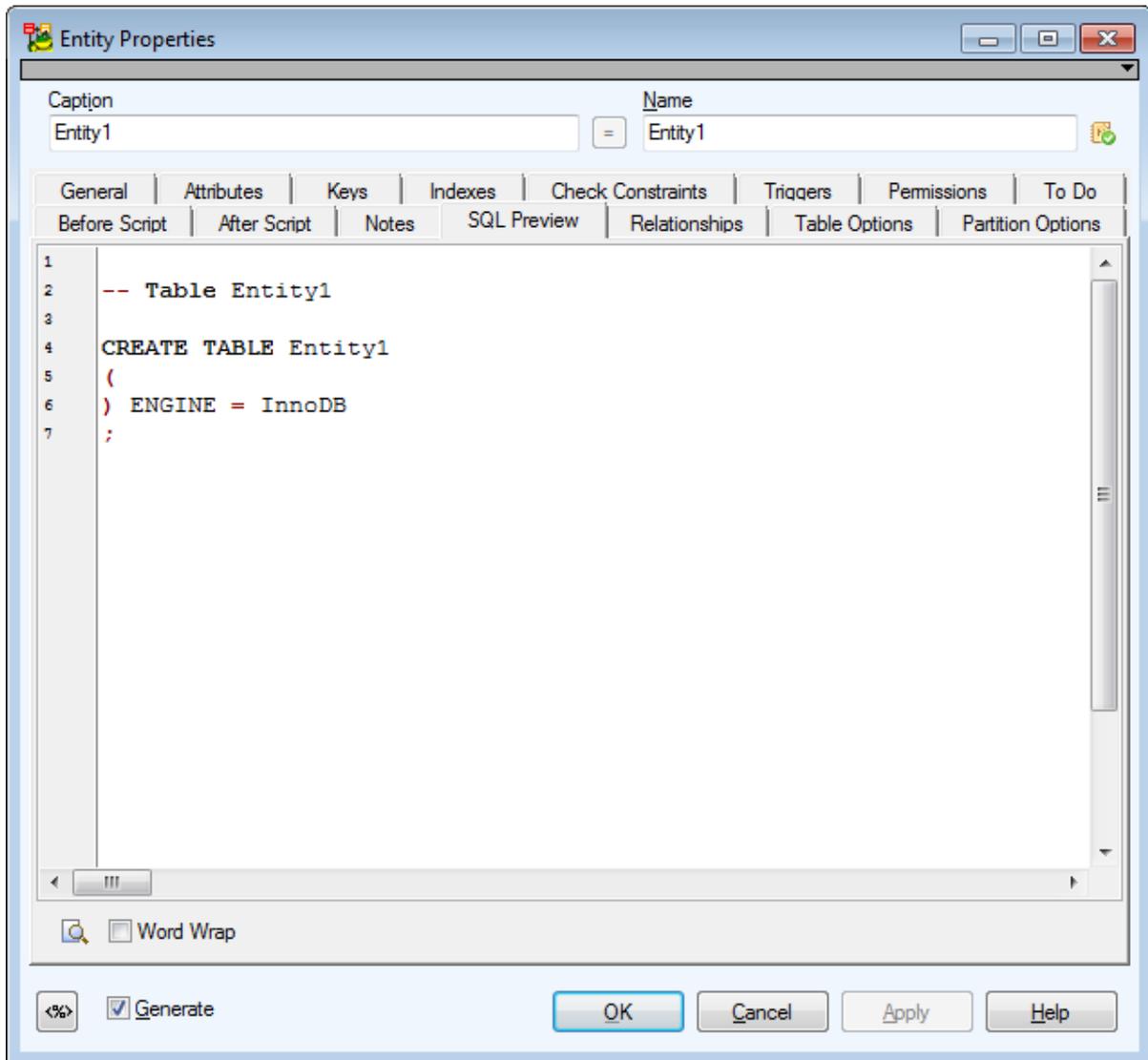
## Partition tab

partitioning table	
column1	int
column2	Char(5)

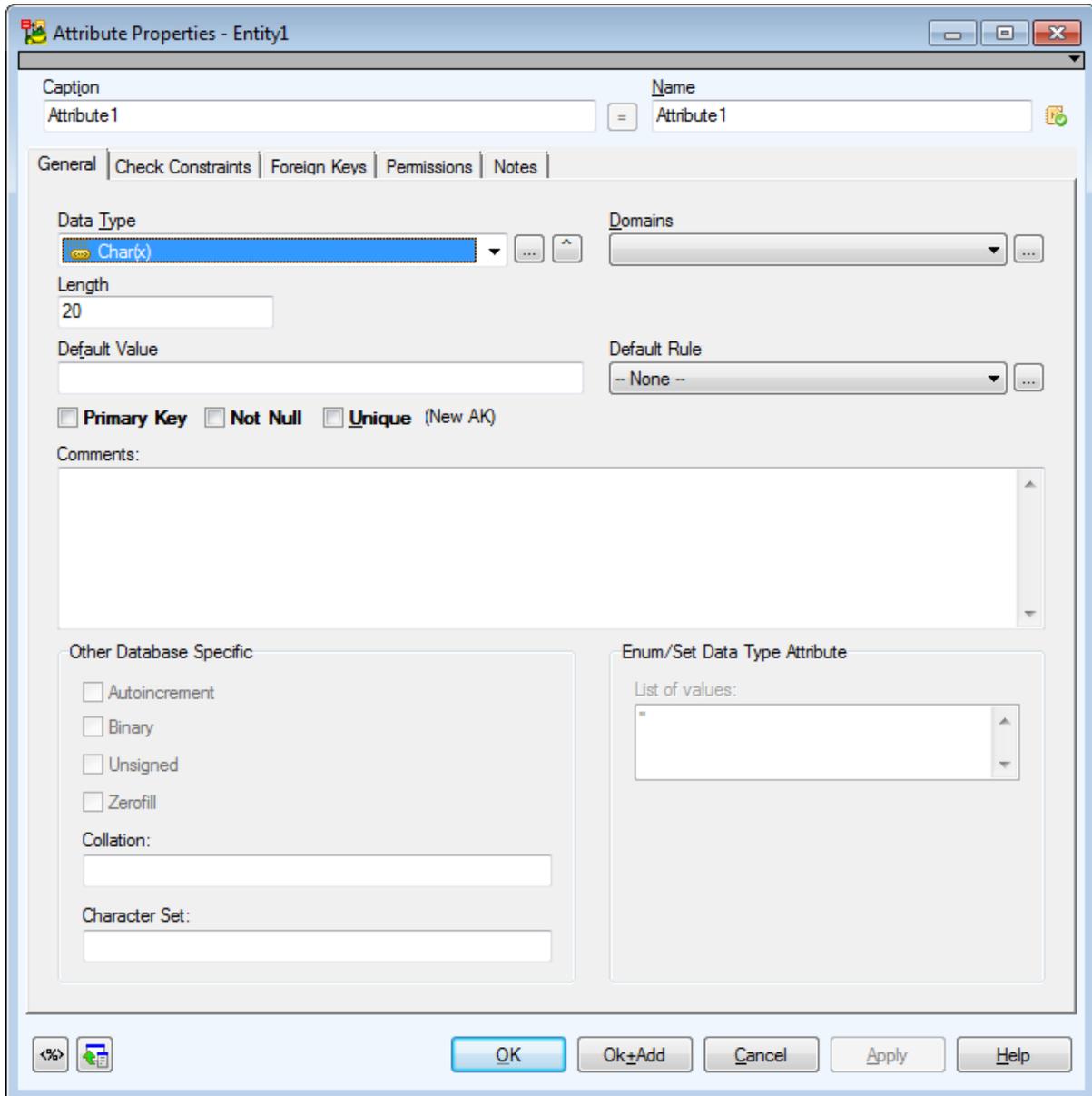


On the **Partition Options** tab, you can write SQL code to create partition. The partition will be generated in SQL code as well as in reports.

See the **SQL Preview** tab:



# Attribute



**Comments** box on tab **General** - The text written in the **Comments** box on entities and attributes is generated in final DDL script.

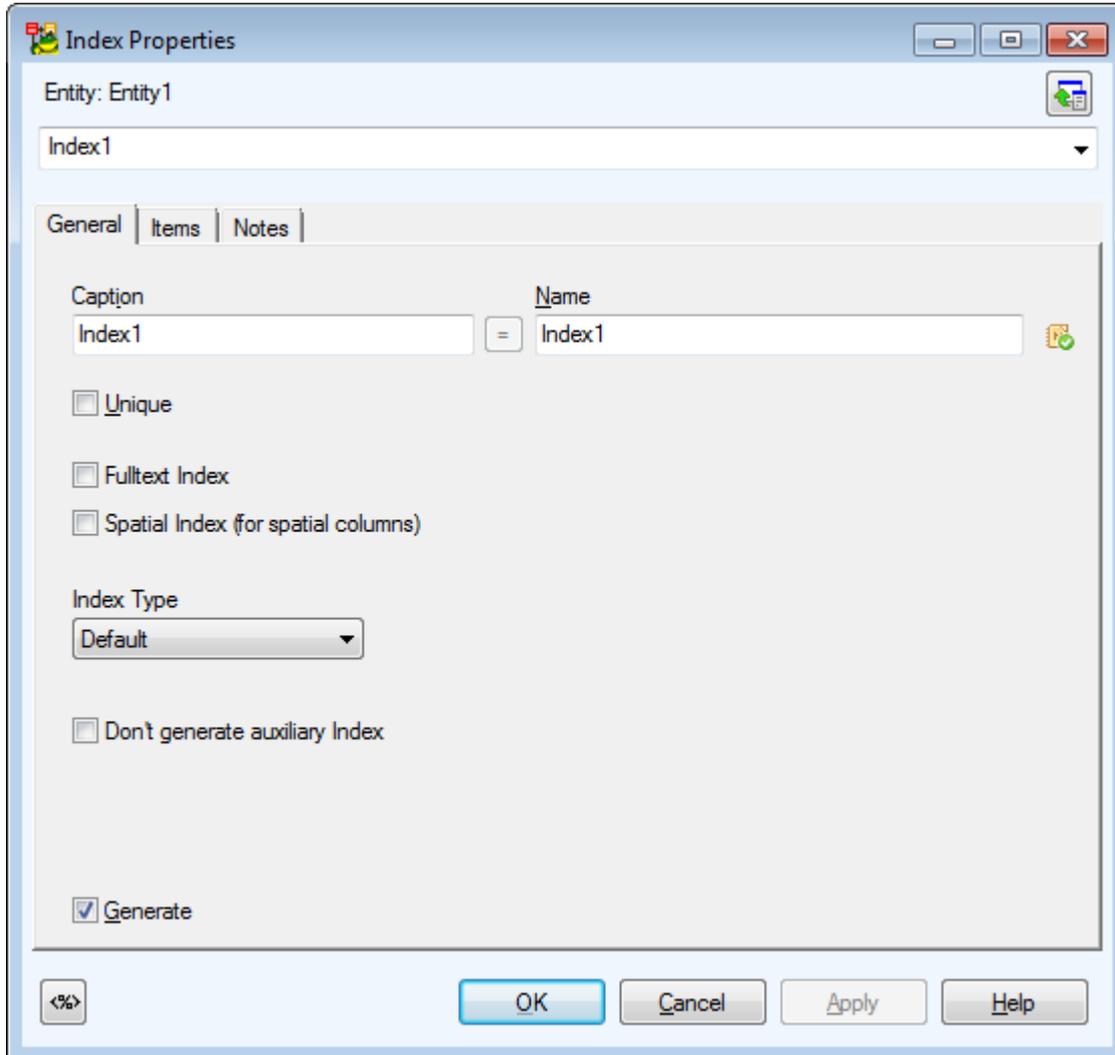
## Model Conversion from MySQL to SQL Server and Oracle

MySQL Enum data type is converted to Char data type, a check constraint for the attribute is created, the parameter is preserved (see the SQL tab of the **Check Constraint Properties** dialog).

# Relationship

Foreign keys are supported only by tables of the InnoDB type.

## Index

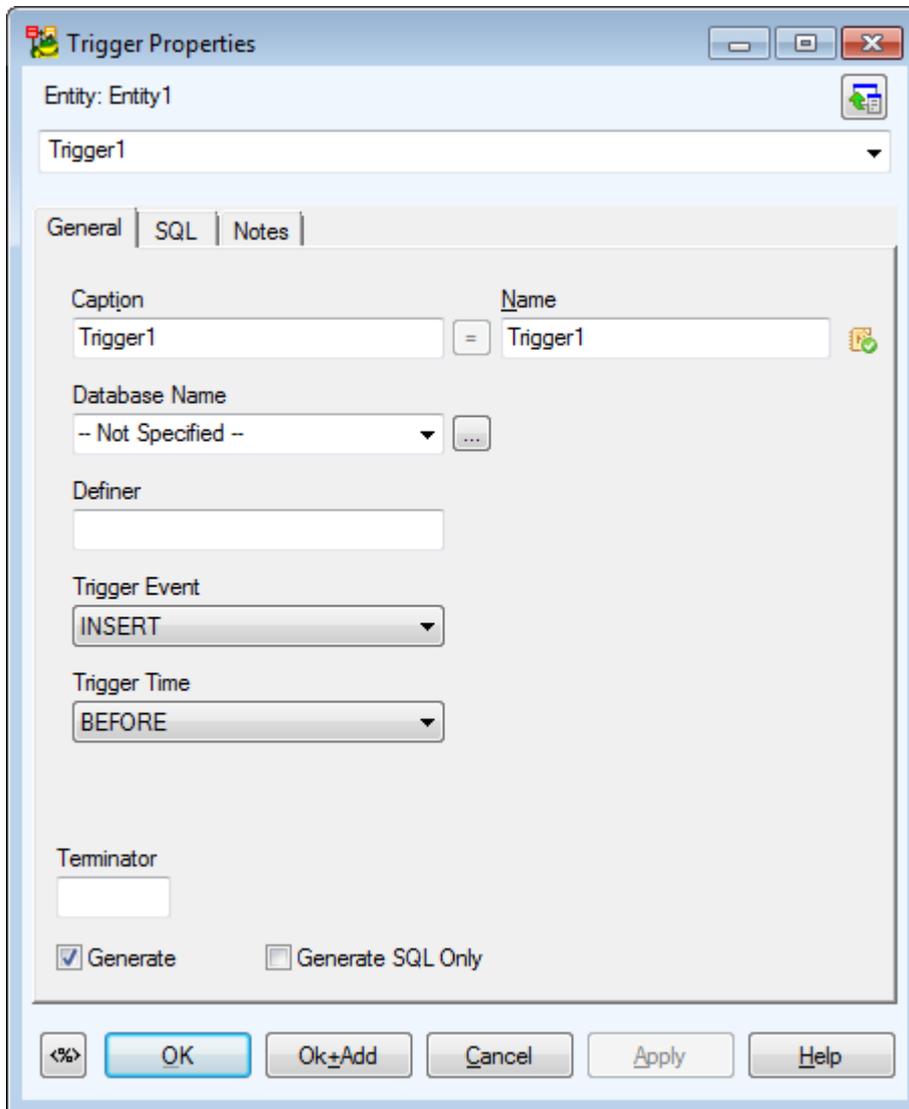


The following types of indexes are supported:

- Default
- BTREE
- HASH

Fulltext indexes are supported only by tables of type MyISAM.

# Trigger



**Trigger Event** options:

- INSERT
- UPDATE
- DELETE

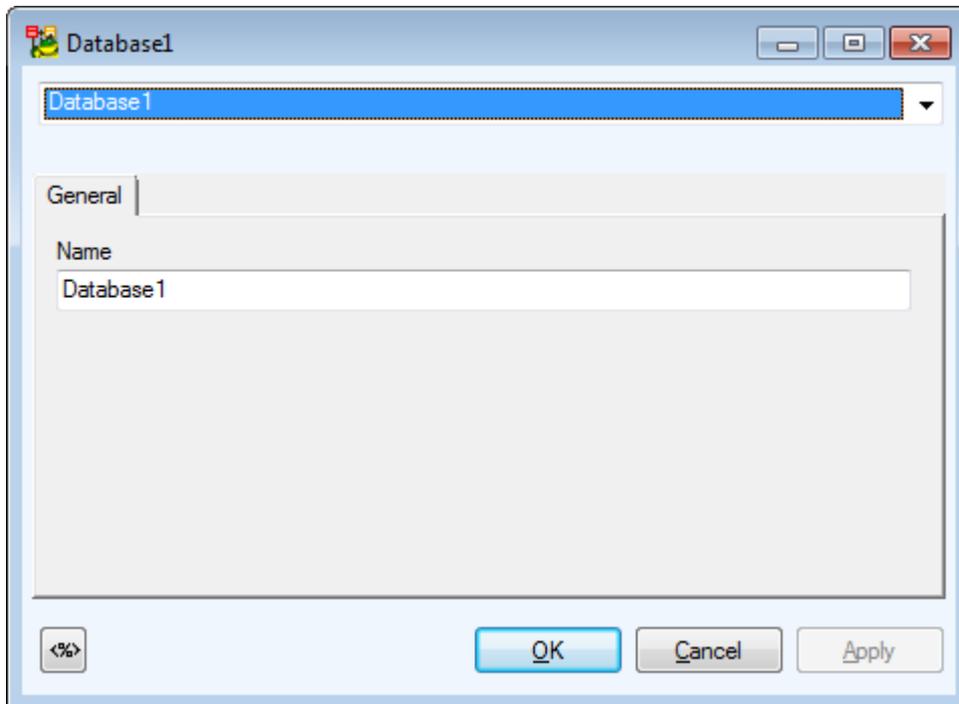
**Trigger Time** options:

- BEFORE
- AFTER

# User Data Types, Dictionary Types

In MySQL 5, User Data Types and Dictionary types are not available.

## Databases



## Data Types

- Implemented several **Data Types** with a **fsp** (fractional seconds precision) parameter - *TIME*, *TIMESTAMP*, *DATETIME*

## Index

- New parameters: *ALGORITHM\_OPTION* (DEFAULT, INPLACE, COPY), *LOCK\_OPTION* (DEFAULT, NONE, SHARED, EXCLUSIVE)
- Support for Index Comments (tab **Comment** added in the **Index Properties** dialog)

## Entities

- New parameters: *STATS\_AUTO\_RECALC*, *STATS\_PERSISTENT*, *STATS\_SIMPLE\_PAGES*

# Reverse Engineering - MySQL 5.6

See [Reverse Engineering - MySQL 8.0](#) for more information.

# Script Generation - MySQL 5.6

See [Script Generation - MySQL 8.0](#) for more information.

# Specifics - MySQL 5.7

## Attributes

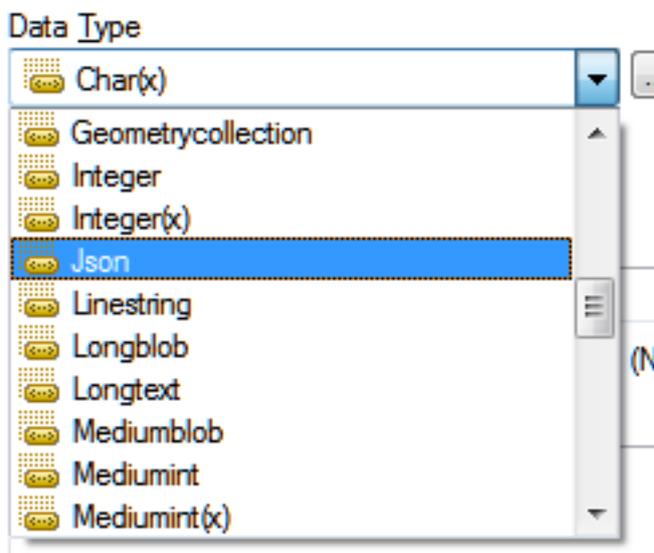
[*GENERATED ALWAYS*] AS (*expression*) [*VIRTUAL* | *STORED*]

Configurable in **Attribute Properties** | **Generated** tab.



## Data Types

New data type - **JSON**

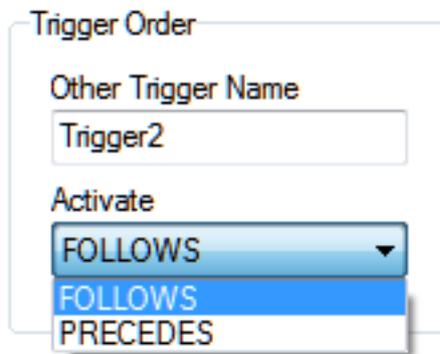


## Tables

*COMPRESSION, ENCRYPTION* parameters ([Entity Properties](#) | [Table Options tab](#) | [Other Table Options](#))

## Triggers

**Trigger Order** - *FOLLOWS, PRECEDES*



Trigger Order

Other Trigger Name  
Trigger2

Activate  
FOLLOWS  
FOLLOWS  
PRECEDES

## Change Script Generation

- Support for generated columns for ALTER TABLE
- Support for multiple TRIGGERS

## Reverse Engineering - MySQL 5.7

See [Reverse Engineering - MySQL 8.0](#) for more information.

## Script Generation - MySQL 5.7

See [Script Generation - MySQL 8.0](#) for more information.

# Specifics - MySQL 8.0

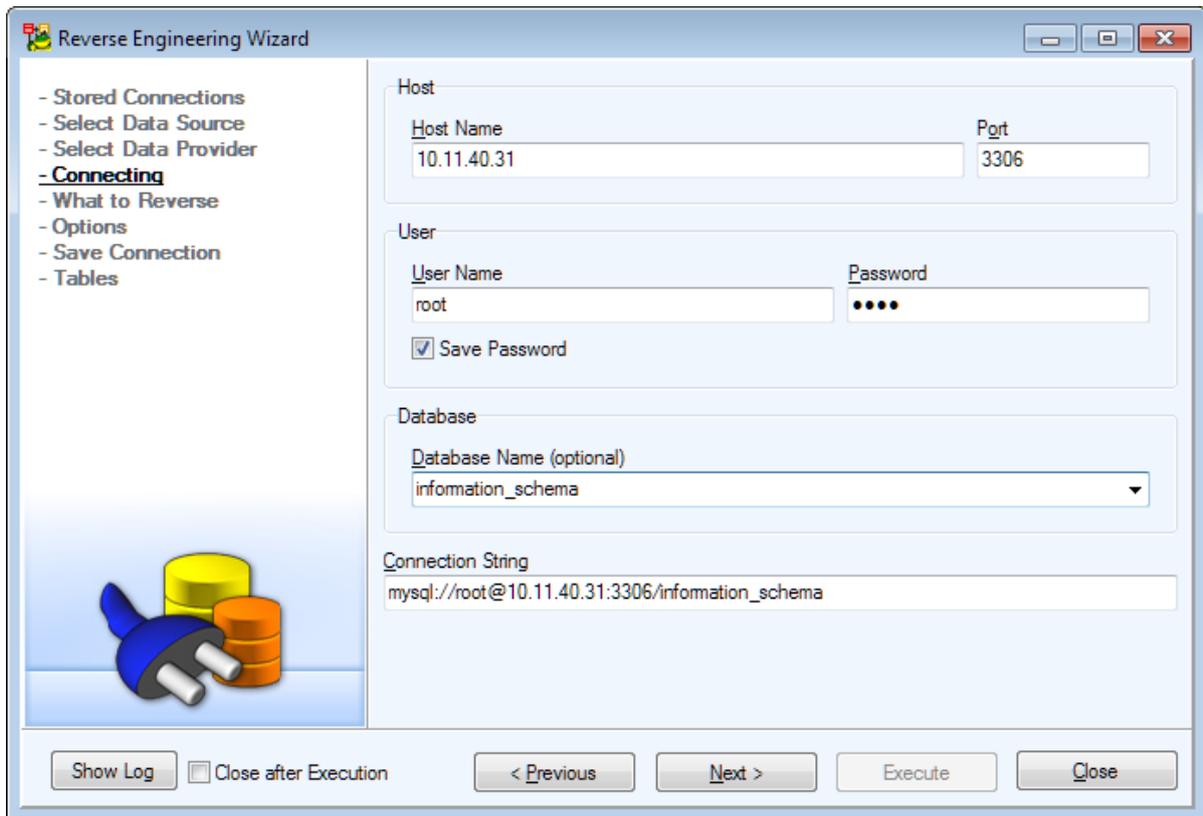
Toad Data Modeler supports MySQL 8.0 in Reverse Engineering from a database, from a SQL file, Change Script Generation and SQL/DDDL Code Generation

## Reverse Engineering - MySQL 8.0

Available **Data Providers** are:

- **Connection via TCP/IP**
- **Connection via ODBC**

**Connection via TCP/IP:**

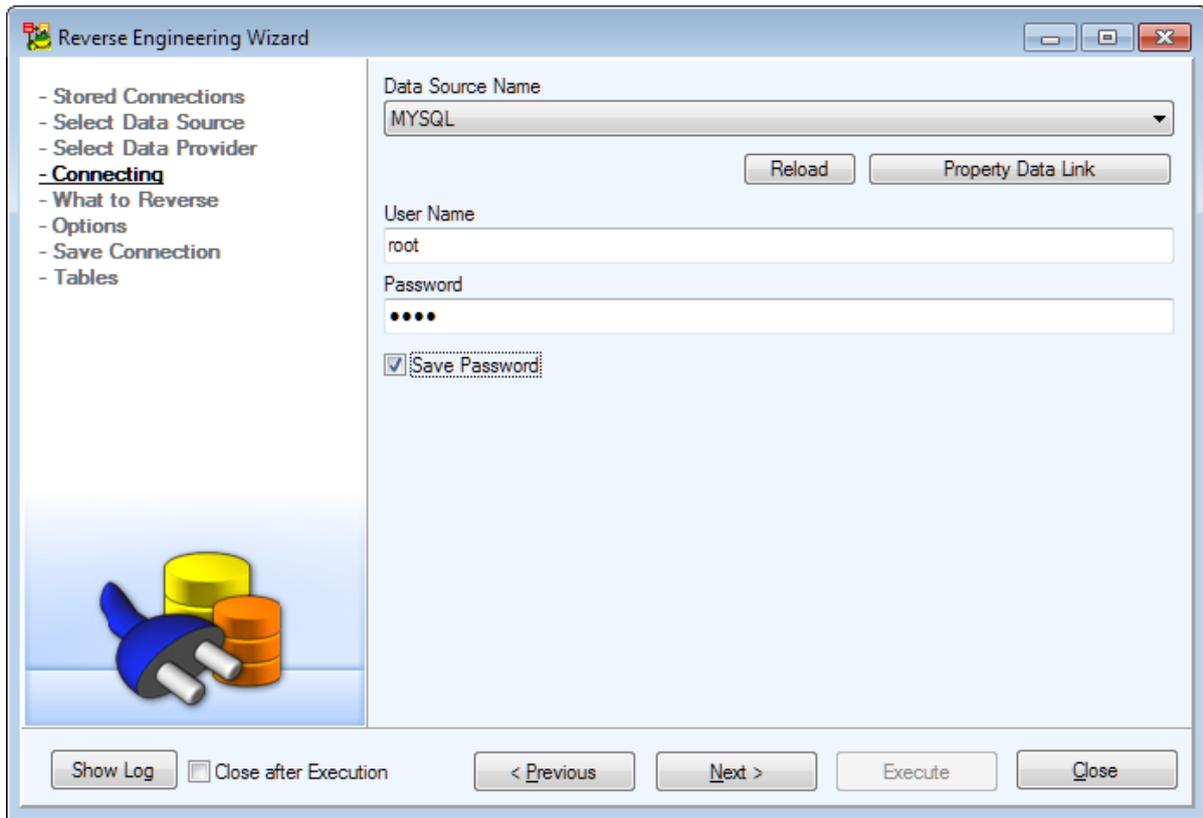


The screenshot shows the 'Reverse Engineering Wizard' dialog box with the 'Connecting' step selected in the left-hand navigation pane. The main area is divided into several sections:

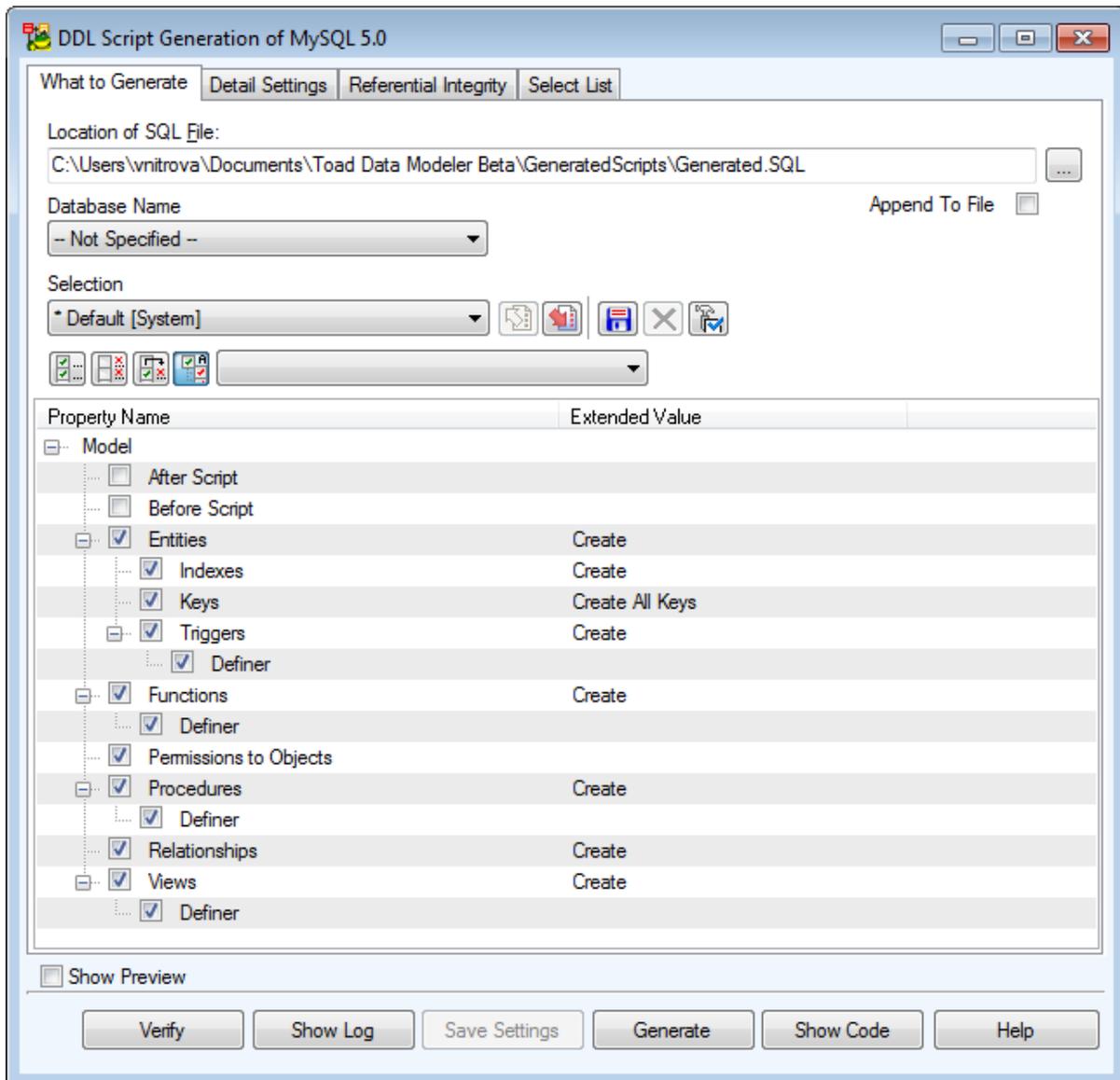
- Host:** Host Name: 10.11.40.31, Port: 3306
- User:** User Name: root, Password: masked with dots. A 'Save Password' checkbox is checked.
- Database:** Database Name (optional): information\_schema
- Connection String:** mysql://root@10.11.40.31:3306/information\_schema

At the bottom, there are buttons for 'Show Log', 'Close after Execution' (checkbox), '< Previous', 'Next >', 'Execute', and 'Close'. An icon of a network plug and database cylinders is visible in the bottom-left corner of the dialog.

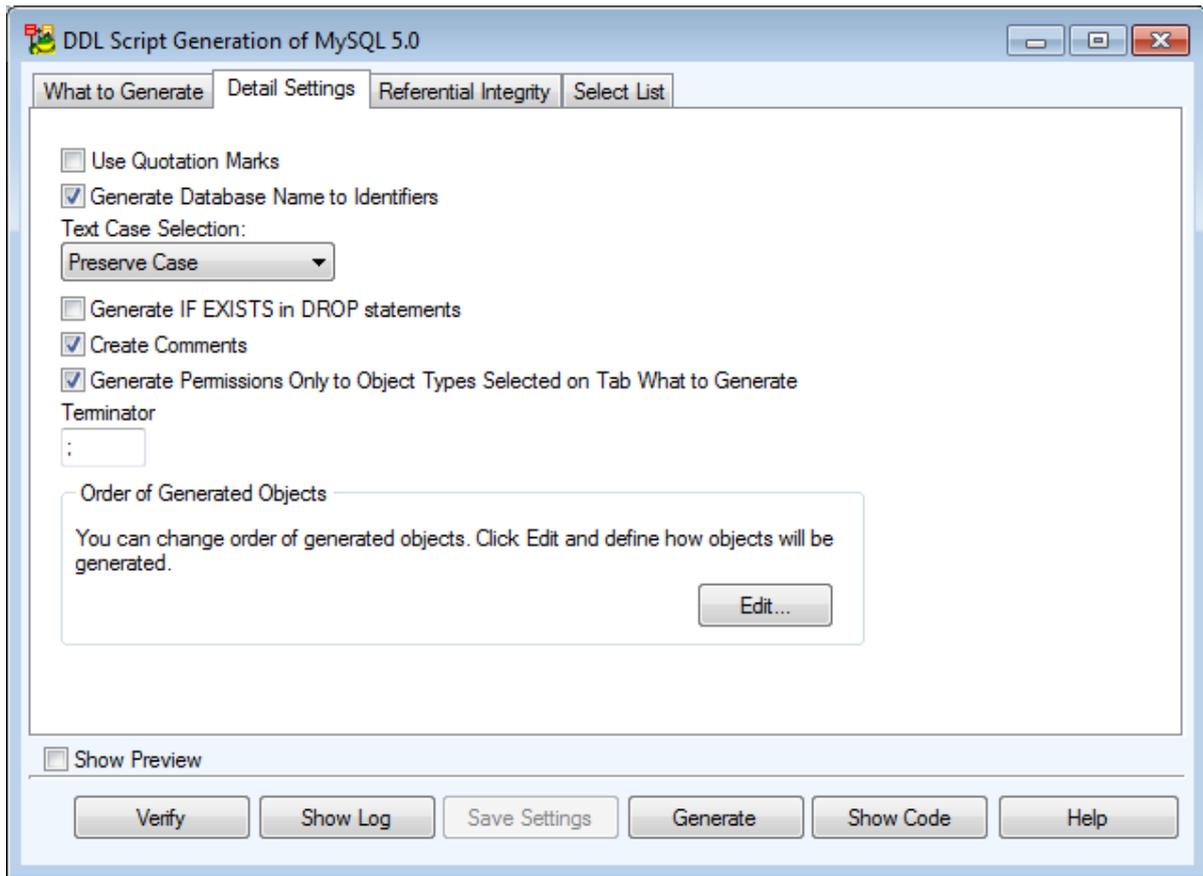
## Connection via ODBC



# Script Generation - MySQL 8.0



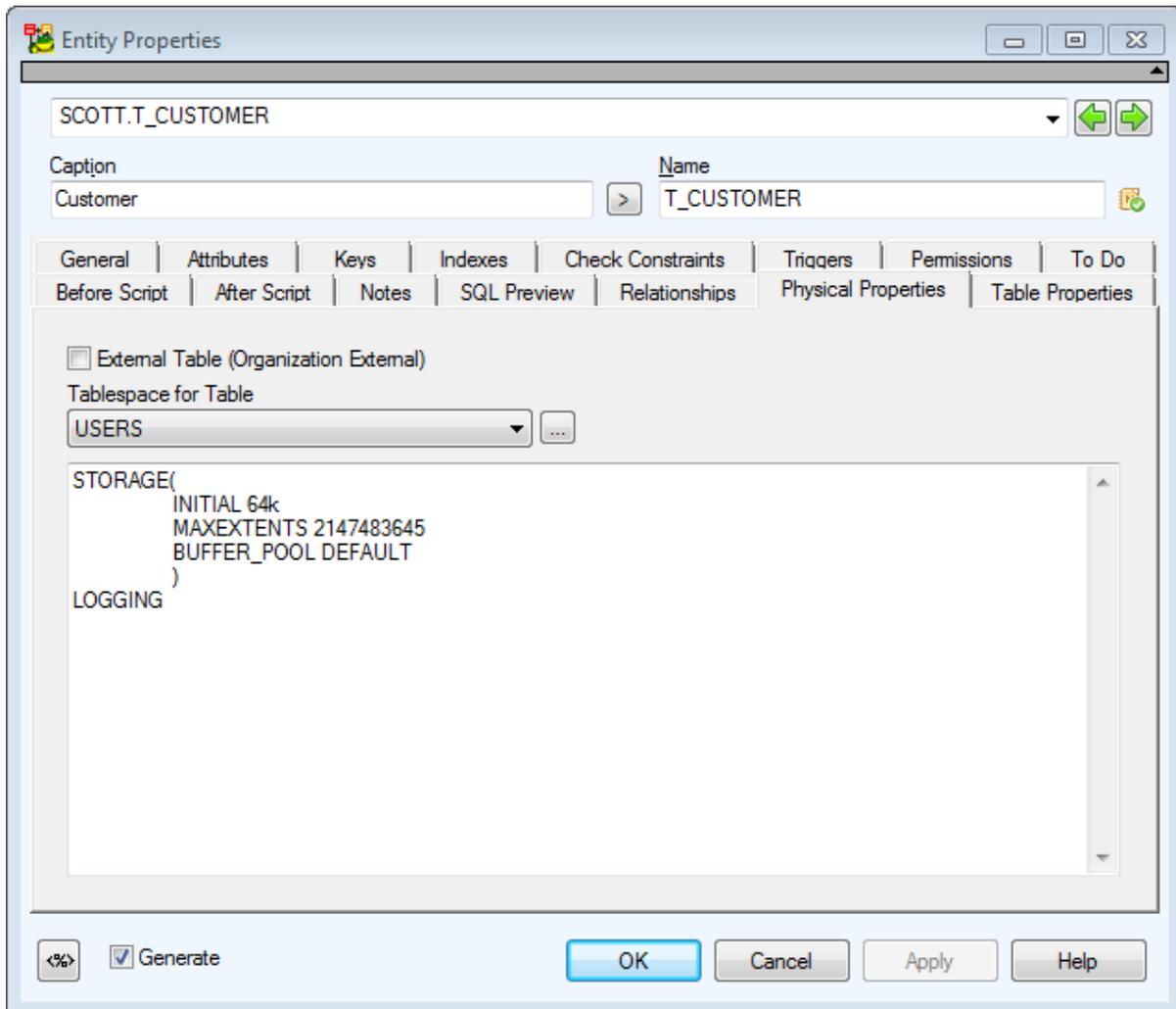
See the property *Definer* on tab **What to Generate**, under Entities/Triggers, Views, Procedures, Functions. During reverse engineering, the *Definer* property is loaded. Nevertheless, if user does not have particular permissions, the generated DDL script with the *Definer* property would not work. Now it is possible to deselect this property for the DDL script generation.



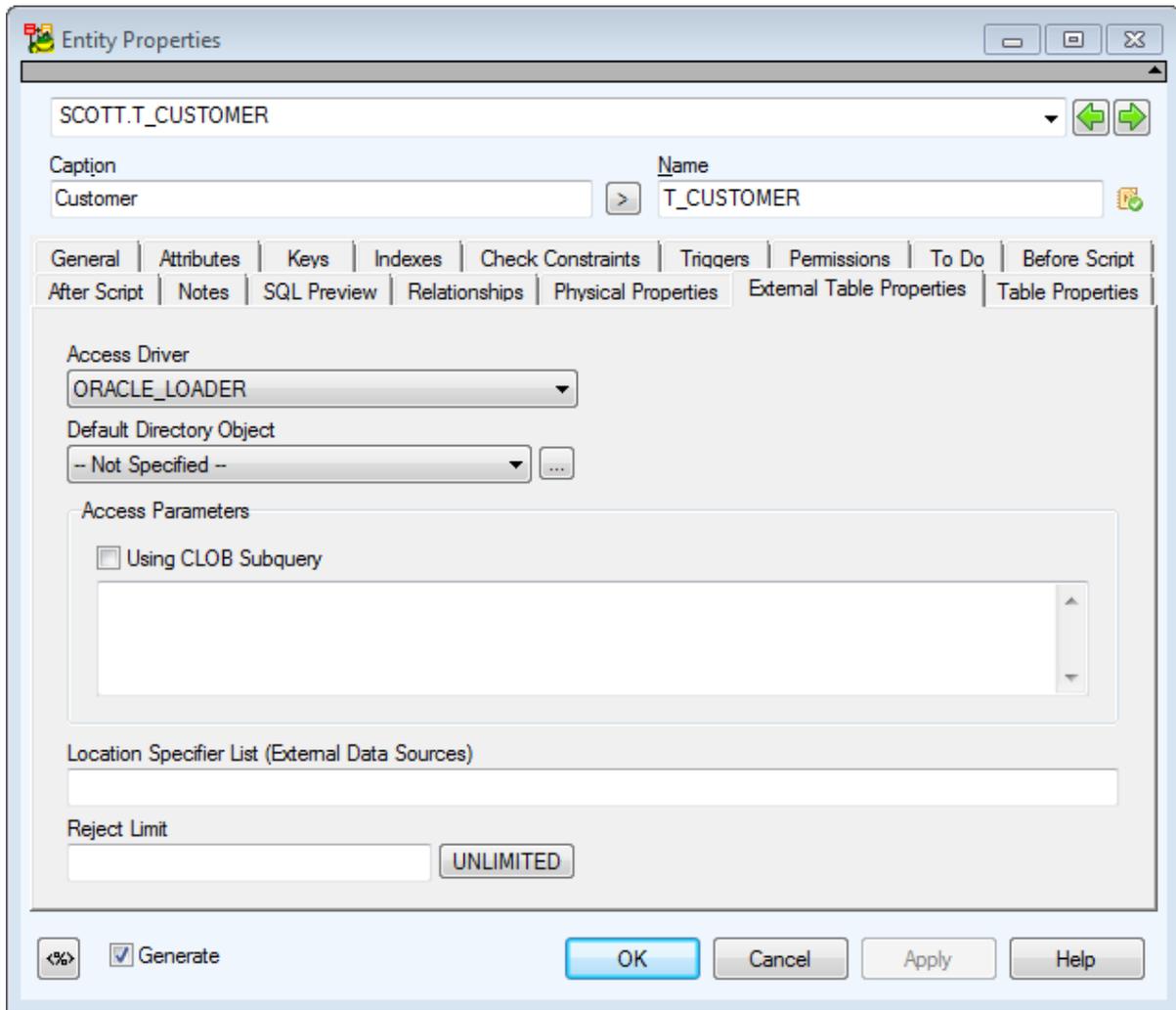
# Specifics - Oracle 11g Release 1

## Entity

### External tables



External tables are supported. On tab **Physical Properties**, select the **External Table (Organization External)** checkbox. The **External Table Properties** tab will occur on the form.



# Attribute

## Data Types

Attribute Properties - SCOTT.T\_CUSTOMER

Caption: Name | Name: name

General | Check Constraints | Foreign Keys | Permissions | Notes

Data Type: Varchar2(x) | Domains: | Length: 20 | Column Length in: CHAR | Default Value: | Default Rule: -- None --

Primary Key  Not Null  Unique (New AK)

Comment: Name column can contain First and Middle name. Sumame must be in different column.

Not Null Constraint: Constraint Name: |  Deferrable  Initially Deferred  Disable  No Validate  Rely

Other Database Specifics: Used Sequence (trigger): -- None -- | Encryption Specification: | REF Type:  Use REF Type | Inline REF Constraint: |

<%> >%> OK Ok+Add Cancel Apply Help

### Data Types:

For Char and Varchar2 data types, you can define **Column Length in**.

For User data type, you can define REF options.

**i** Note: Change for CHAR/BYTE of attributes: Previously, it was necessary to select CHAR or BYTE for Char (x) and Varchar(x) data types of attribute (domain) in **Column Length in** combo box. Now there is a new item "Default" that is selected by default.

Reverse engineering: During RE TDM finds out what is the default item in the Oracle database. Example: BYTE is a default item in Oracle database.

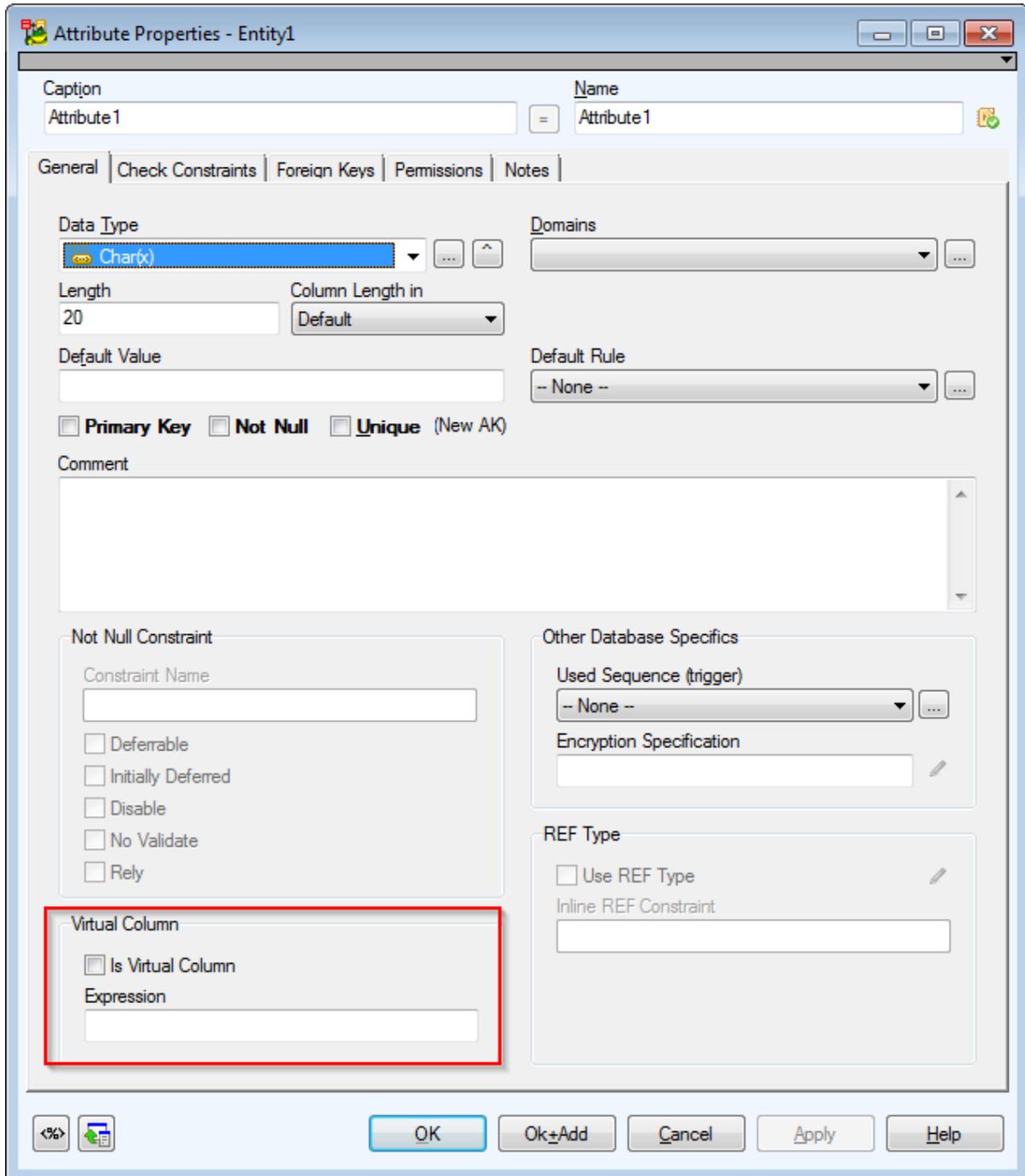
- Previous behavior: BYTE was loaded for the attribute.
- Current behavior: "Default" item is shown for the attribute. BYTE will be shown in Model Properties dialog | Database Parameters tab | Length Semantics box (it is not used during script generation).
- If you want to preserve the previous behavior of TDM, select the "Load CHAR/BYTE to Attributes Regardless Database Default Settings" checkbox in the RE Wizard.

**Used Sequence (trigger)** - From this box, you can select a sequence that you want to use for the attribute. (A new trigger will be created in SQL script.)

**Encryption Specification** - Here, you can write e.g. USING 'AES192' NO SALT.

# Virtual Column

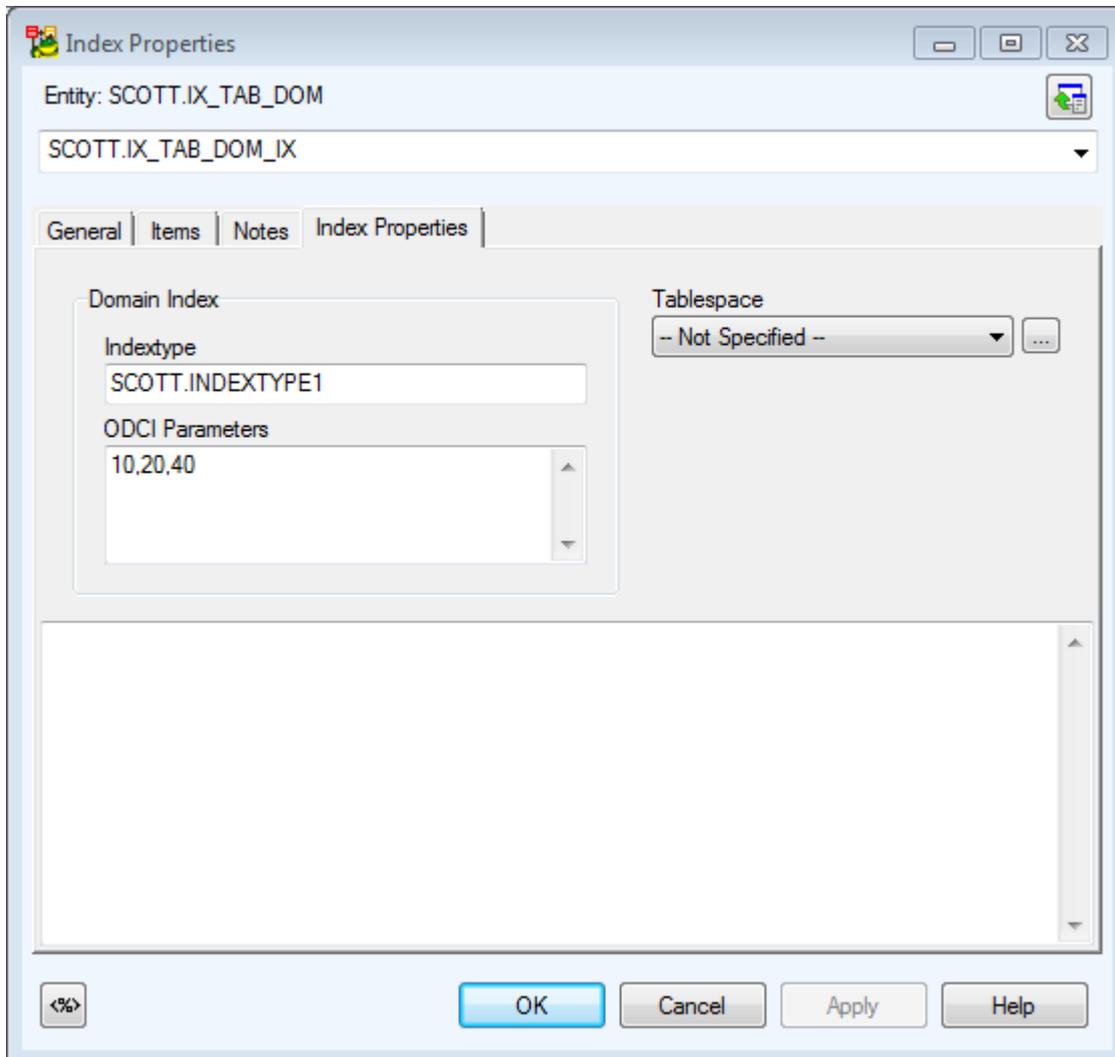
**Virtual Column area** - If you want the column to be virtual column, select the **Is Virtual Column** checkbox and fill out the **Expression** box (e.g. Attribute2 \* Attribute3) and set appropriate data type.



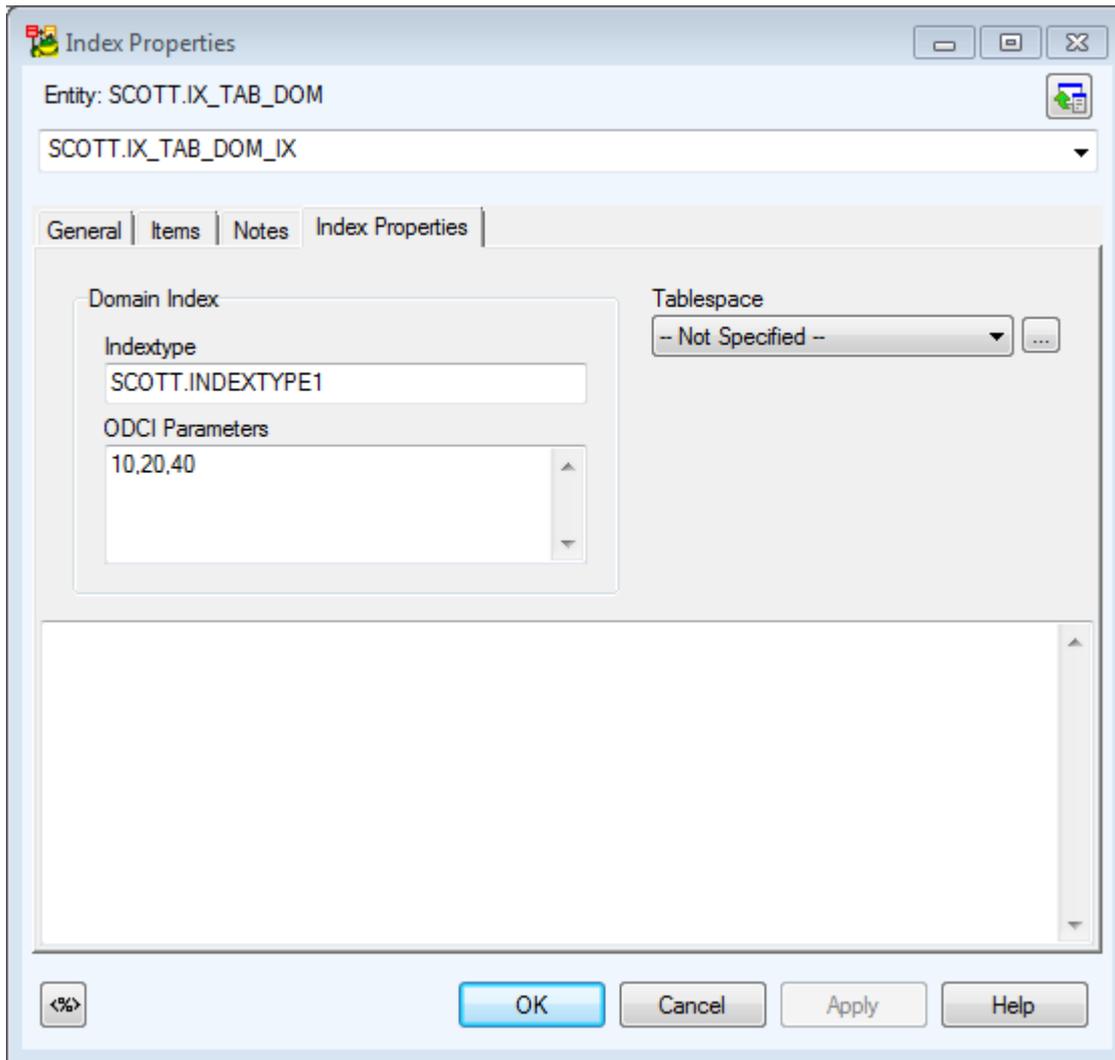
# Index

## Expression Index

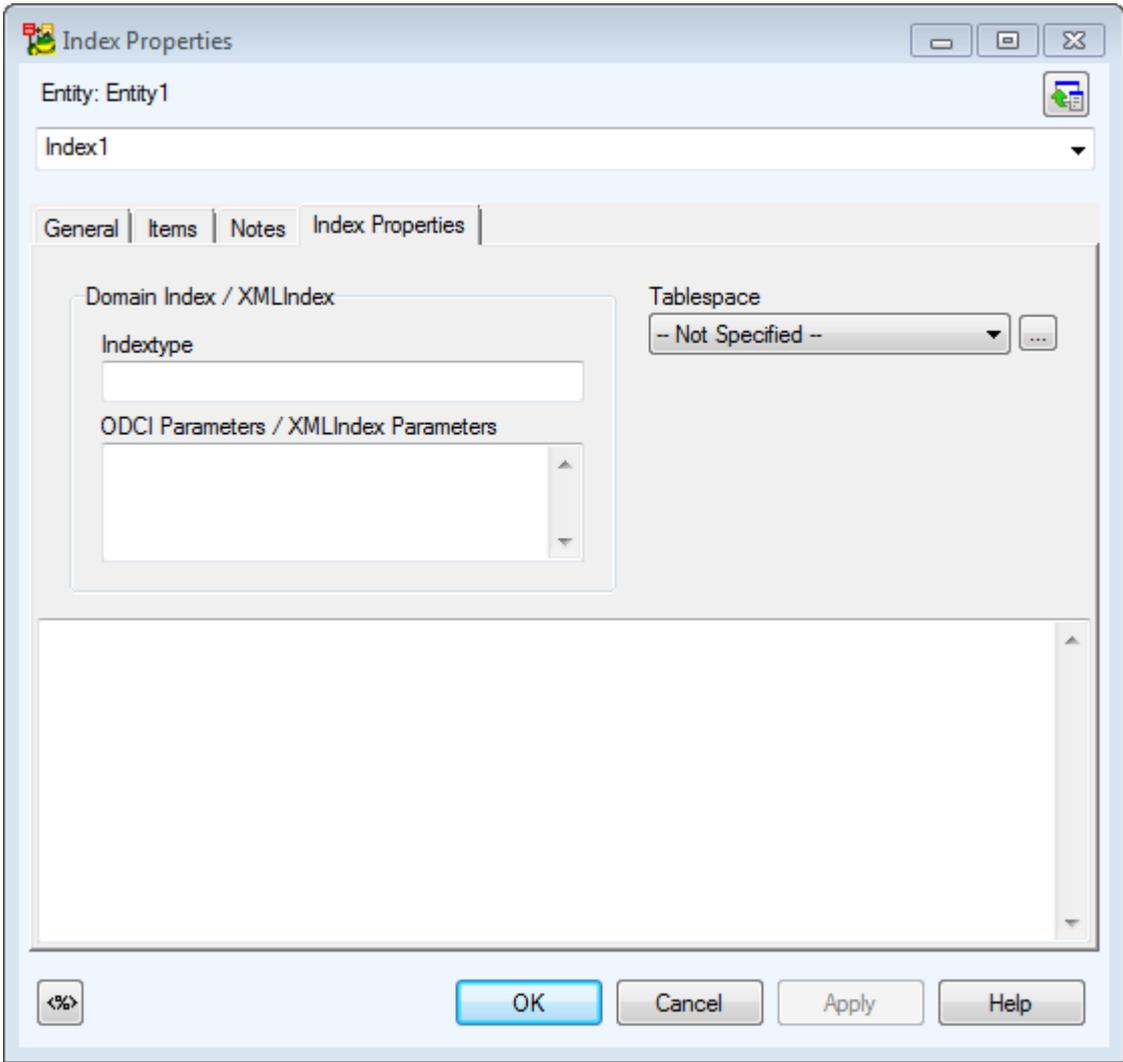
Example of Expression index:



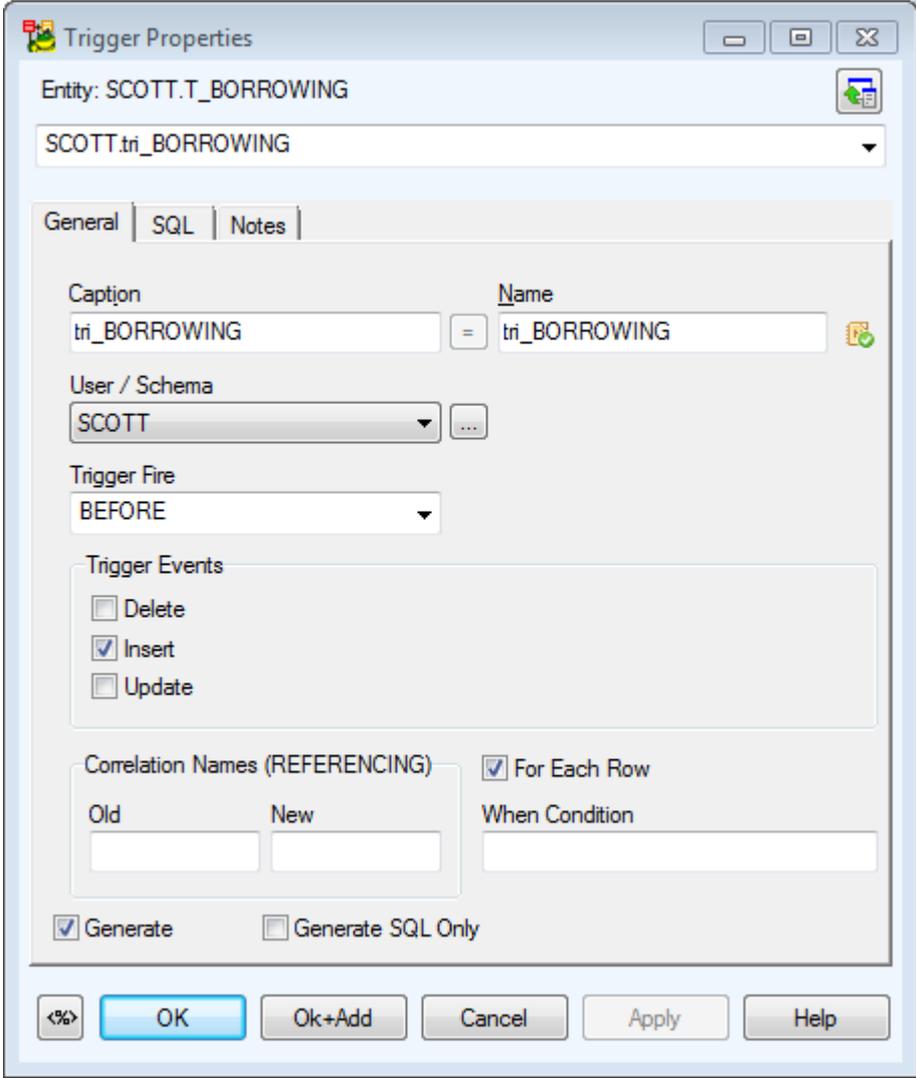
**Example:** Index properties (for domain index):

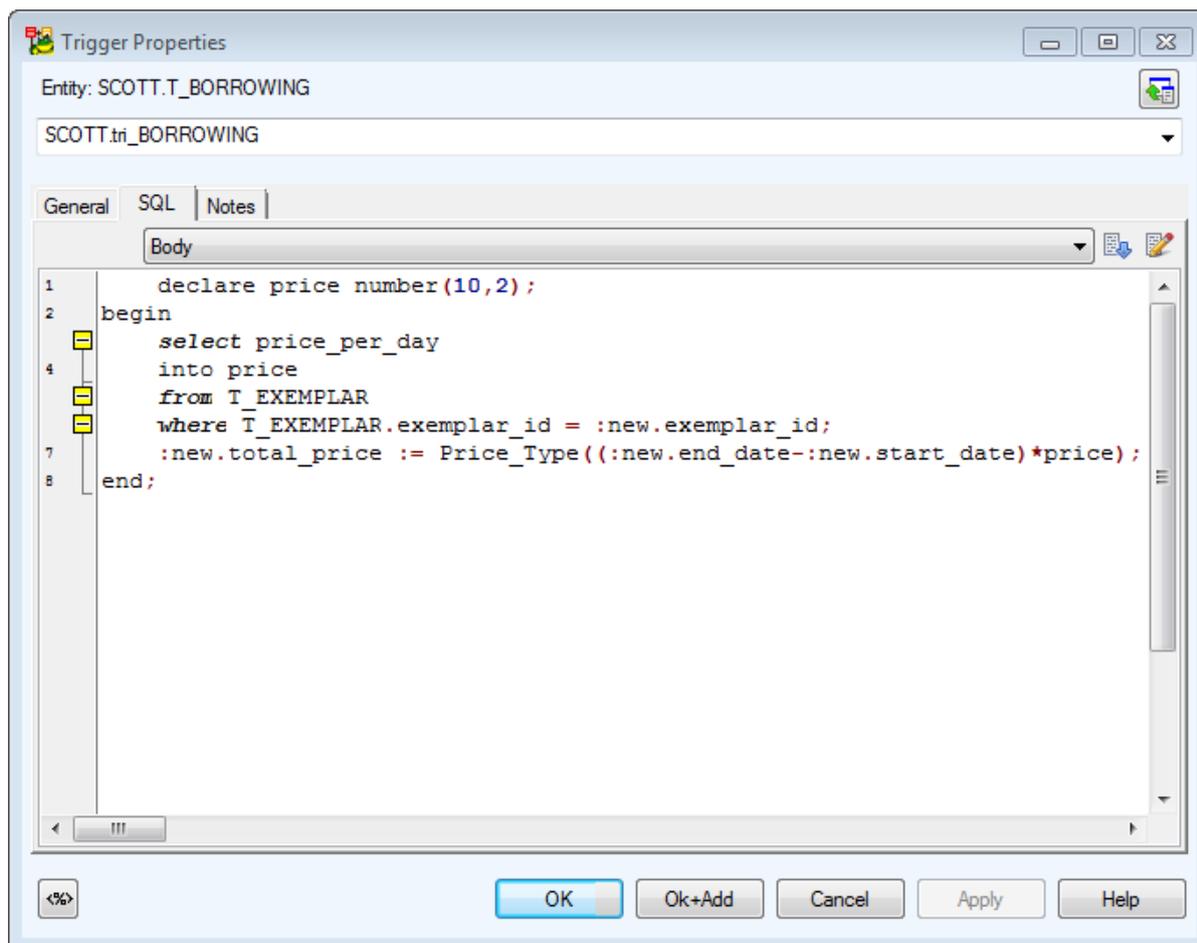


# XML Index



# Trigger





## Trigger (Entity)

Trigger Properties

Entity: Entity1

Trigger1

General | SQL | Notes

Caption: Trigger1 = Name: Trigger1

User / Schema: - Not Specified -

Trigger Fire: COMPOUND

Trigger Events:

- Delete
- Insert
- Update

Correlation Names (REFERENCING):

Old: [ ] New: [ ] When Condition: [ ]

For Each Row

Trigger Ordering:

Follows: [ ]

Generate  Enabled  Generate SQL Only

<%> OK Ok+Add Cancel Apply Help

The **Trigger Fire** box with an item **COMPOUND**.

Body of Compound Trigger, including initial key sentence "COMPOUND TRIGGER", should be written on tab **SQL**.

From the **Trigger Fire** box, you can select items **COMPOUND** or **BEFORE** or **AFTER**.

## Trigger (View)

Trigger Properties

View: View1

Trigger1

General | SQL | Notes

Caption: Trigger1 = Name: Trigger1

User / Schema: -- Not Specified --

Trigger Fire: COMPOUND

Trigger Events:

- Delete
- Insert
- Update

Nested Table Column:

Correlation Names (REFERENCING):

Old	New	Parent

For Each Row

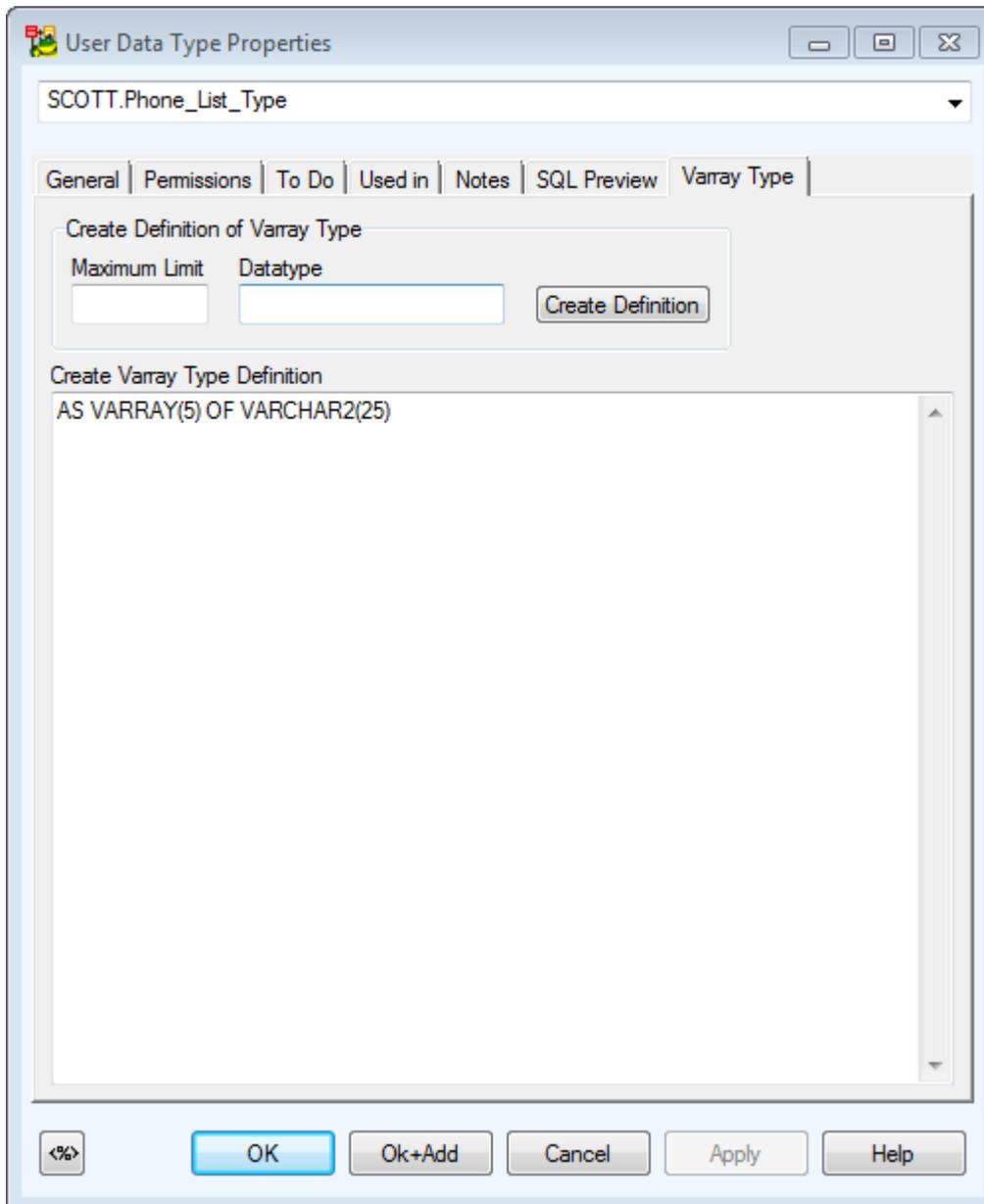
Trigger Ordering: Follows

Generate  Enabled  Generate SQL Only

<%> OK Ok+Add Cancel Apply Help

From the **Trigger Fire** box, you can select items **COMPOUND** or **INSTEAD OF**.

# User Data Type



When you select *Varray Type* or *Nested Table Type*, a new tab displays (*Varray Type* tab or *Nested Table Type* tab). There, you can see new boxes where you write appropriate values. Click **Create Definition** and the definition will be written in the memobox.

The content of the boxes is not saved and is not used for reverse engineering, script or report generation.

Hints in memoboxes for User Data Types are shown.

## Other Notes

- Dictionary types are not supported for Oracle models.
- [User Data Types](#) on user data types in Oracle model.

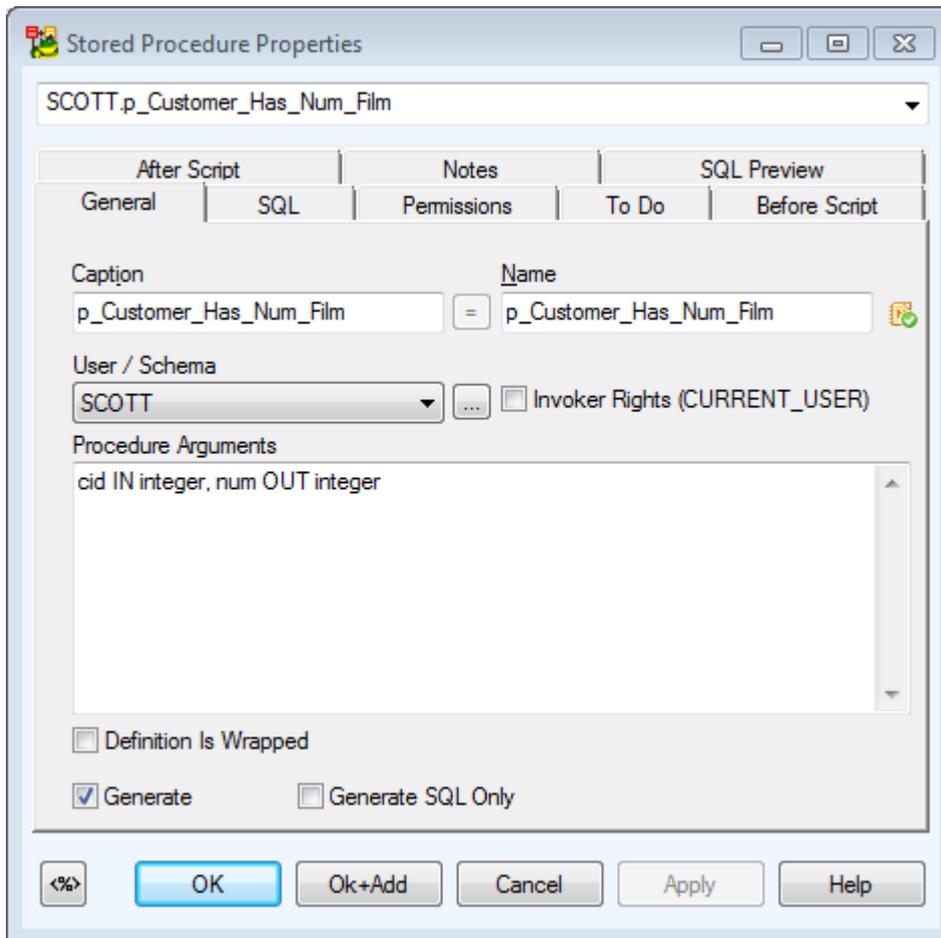
## Materialized View

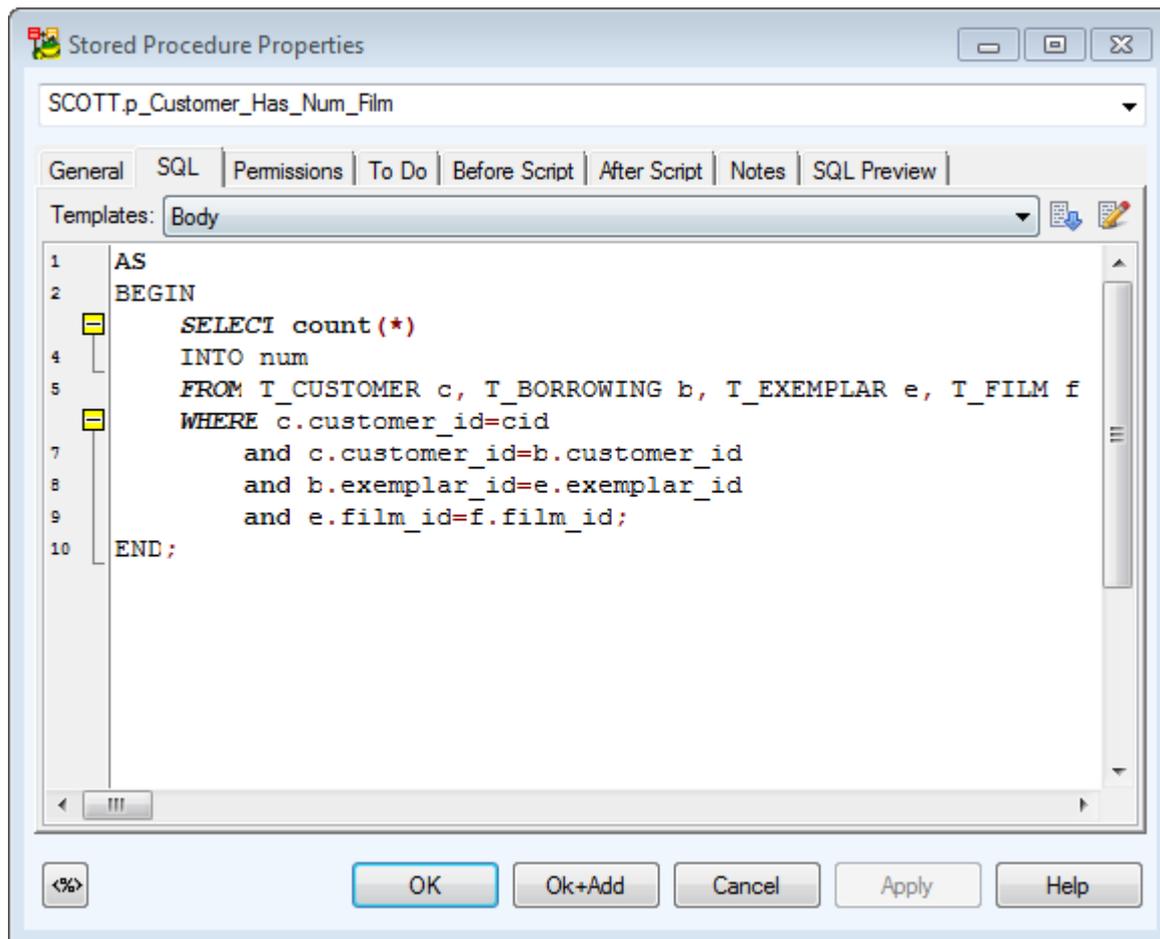
The screenshot shows the 'Materialized View Properties' dialog box for the materialized view 'SCOTT.MV\_EMP\_PK\_NEVERREFRESH'. The dialog has several tabs: 'SQL Preview', 'Refresh', 'Physical Properties', 'Materialized View Properties' (selected), and 'Create Index'. Below the tabs are sub-tabs: 'General', 'SQL', 'Permissions', 'To Do', 'Before Script', 'After Script', and 'Notes'. The main area contains the following fields and options:

- Caption:** MV\_EMP\_PK\_NEVERREFRESH
- Name:** MV\_EMP\_PK\_NEVERREFRESH
- User / Schema:** SCOTT
- Column Aliases:** "EMPNO","ENAME","JOB","MGR","HIREDATE","SAL","COMM","DEPTNO"
- Prebuilt Table:** -- None --
- Using Index:**  Using No Index
- For Update:**
- Query Rewrite:** DISABLE
- Categories:** -- None --
- Generate:**  **Generate SQL Only:**

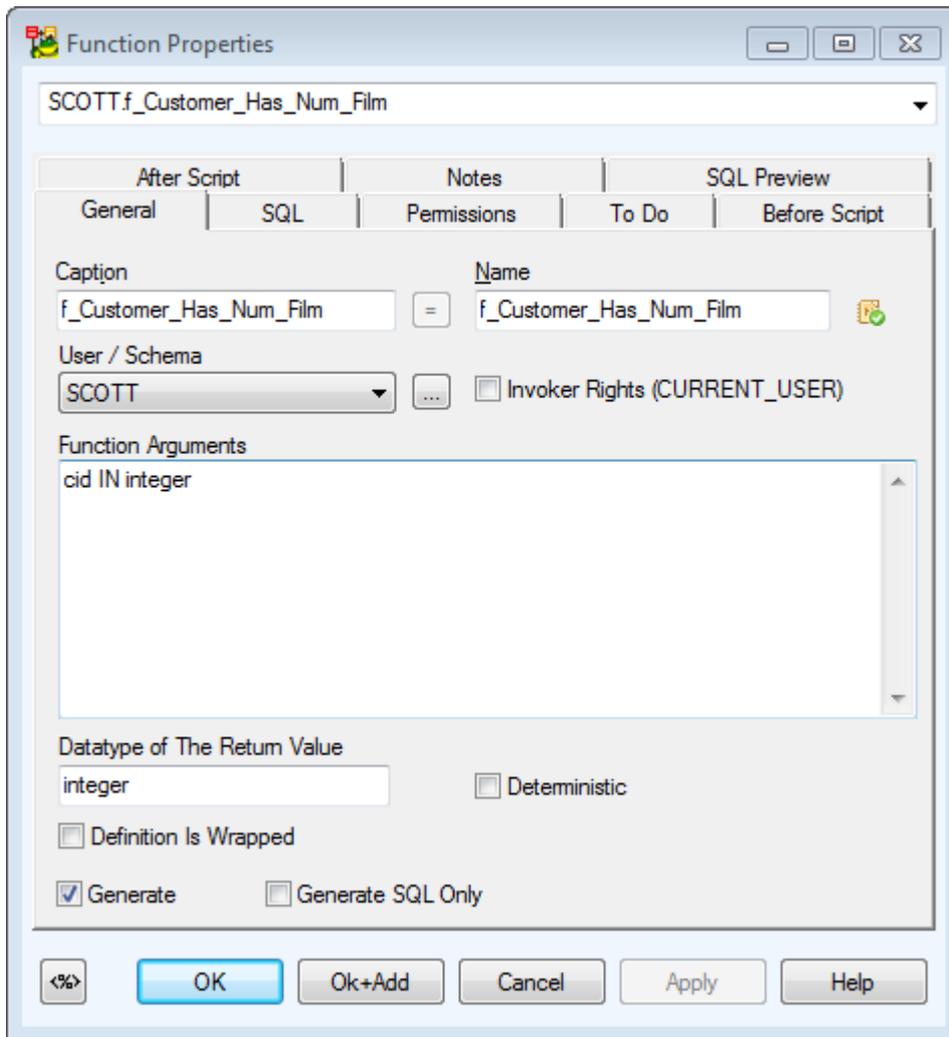
At the bottom, there are buttons for '<%' (Help), 'OK', 'Ok+Add', 'Cancel', and 'Apply'.

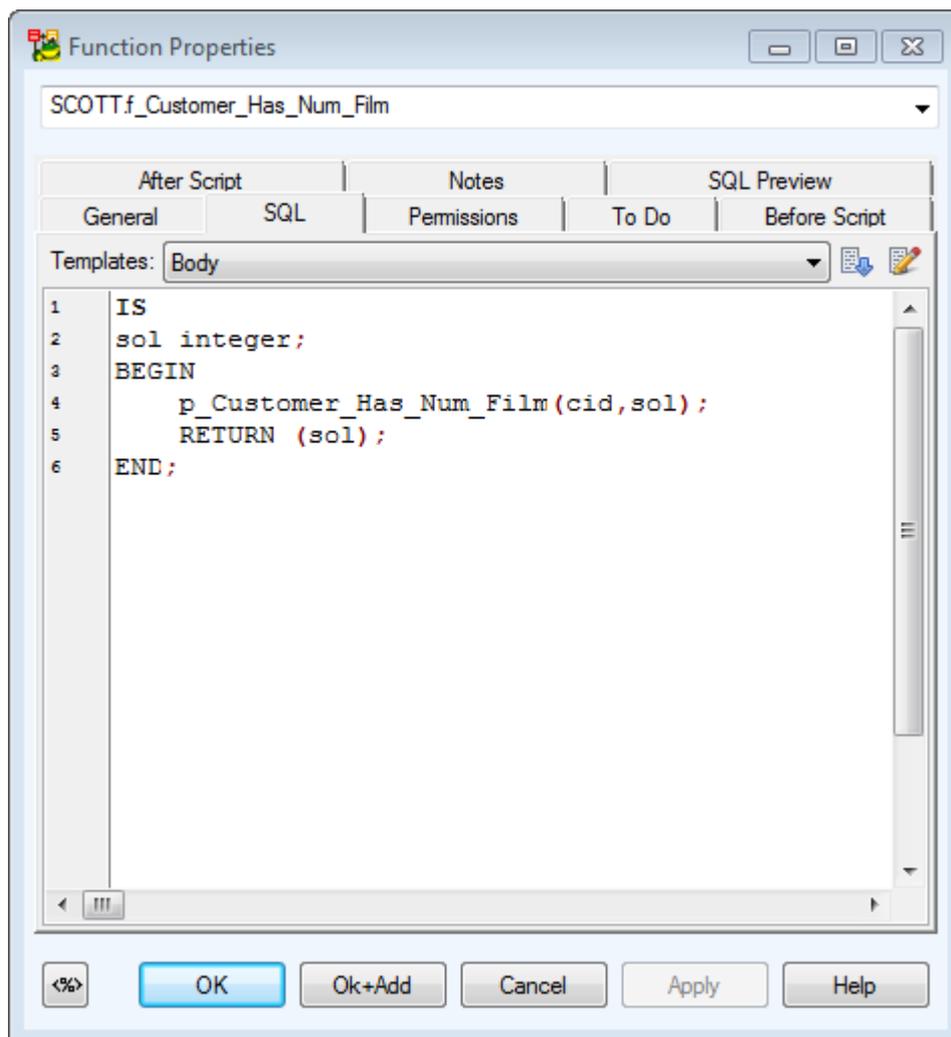
# Procedure



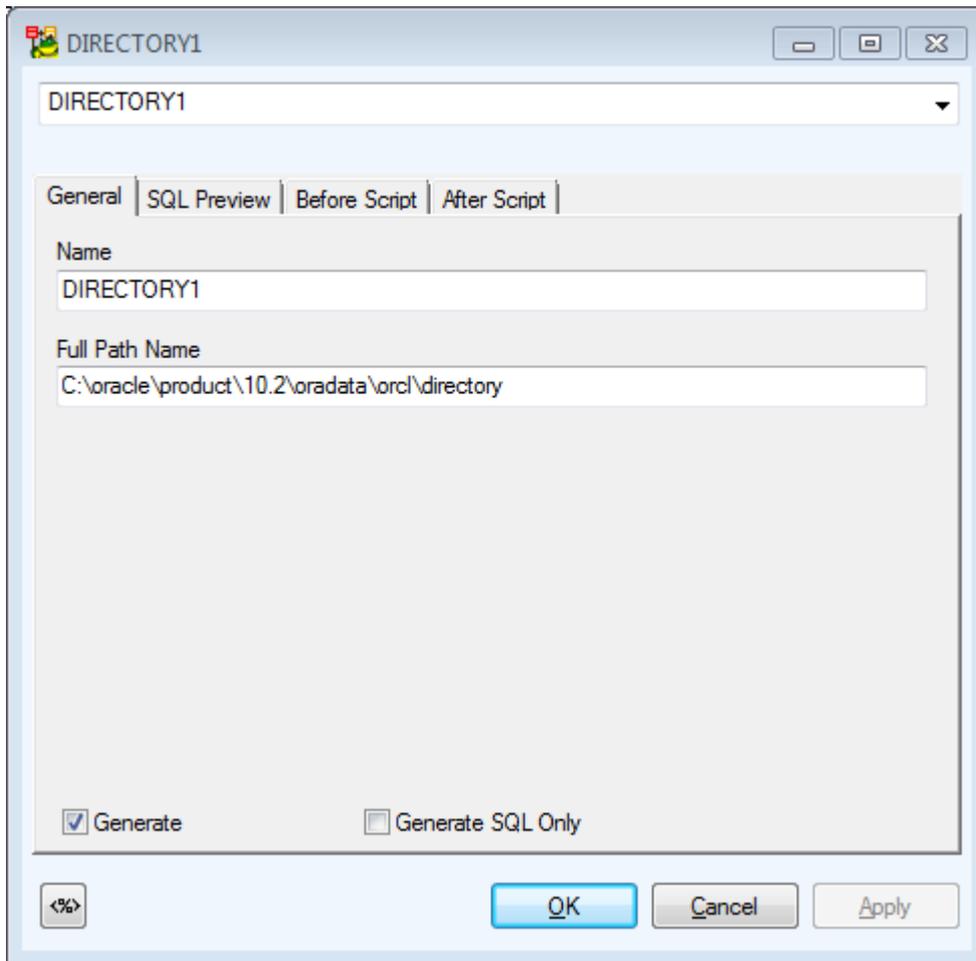


# Function

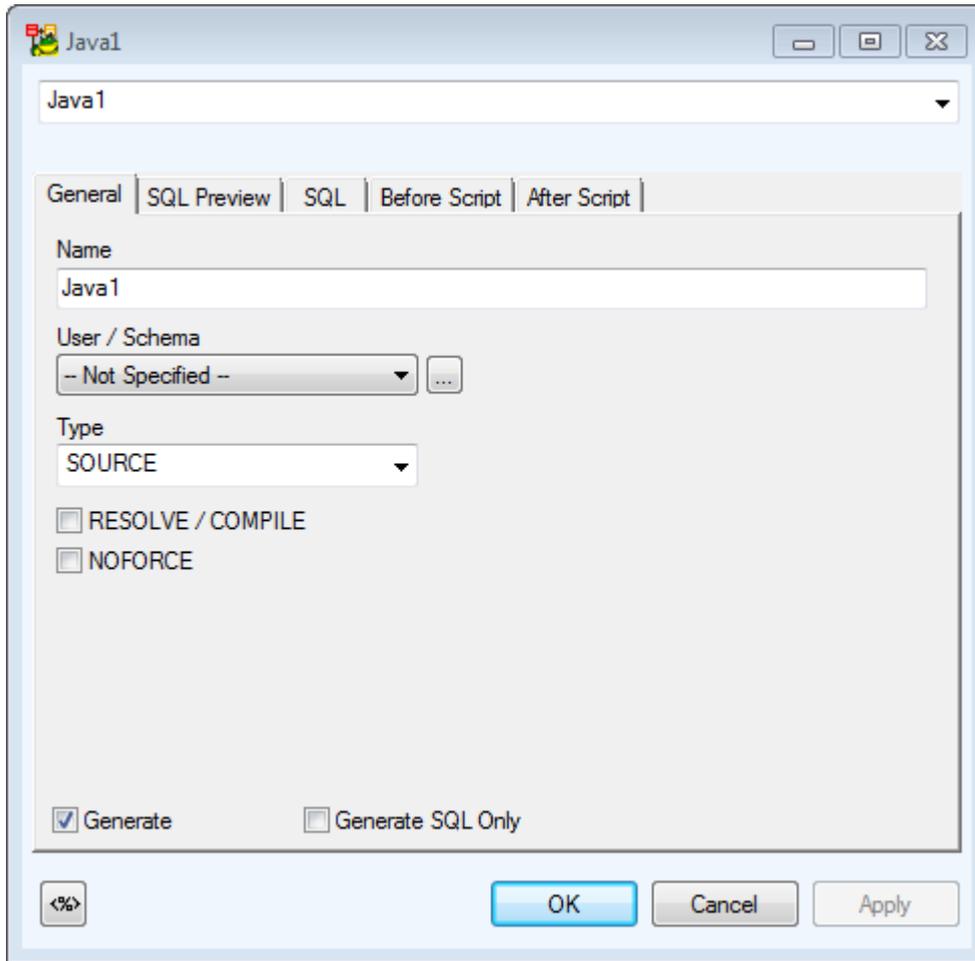




# Directory



# Java



Support for Oracle Java Source, including reverse engineering: All the three types of Java schema objects (SOURCE, CLASS, RESOURCE) should be written to this object. - During reverse engineering only SOURCE types are loaded, and from them only sequences of characters for Java source. Other Java schema objects (CLASS, RESOURCE) are loaded only as a name. Checkbox **Generate** is unselected for all the Java schema objects (SOURCE, CLASS, RESOURCE).

# Sequence

MY\_SEQ

SCOTT.MY\_SEQ

General | SQL Preview | Before Script | After Script

Name  
MY\_SEQ

User / Schema  
SCOTT

Increment By  
2

Start With

Maxvalue  
 NOMAXVALUE

Minvalue  
 NOMINVALUE

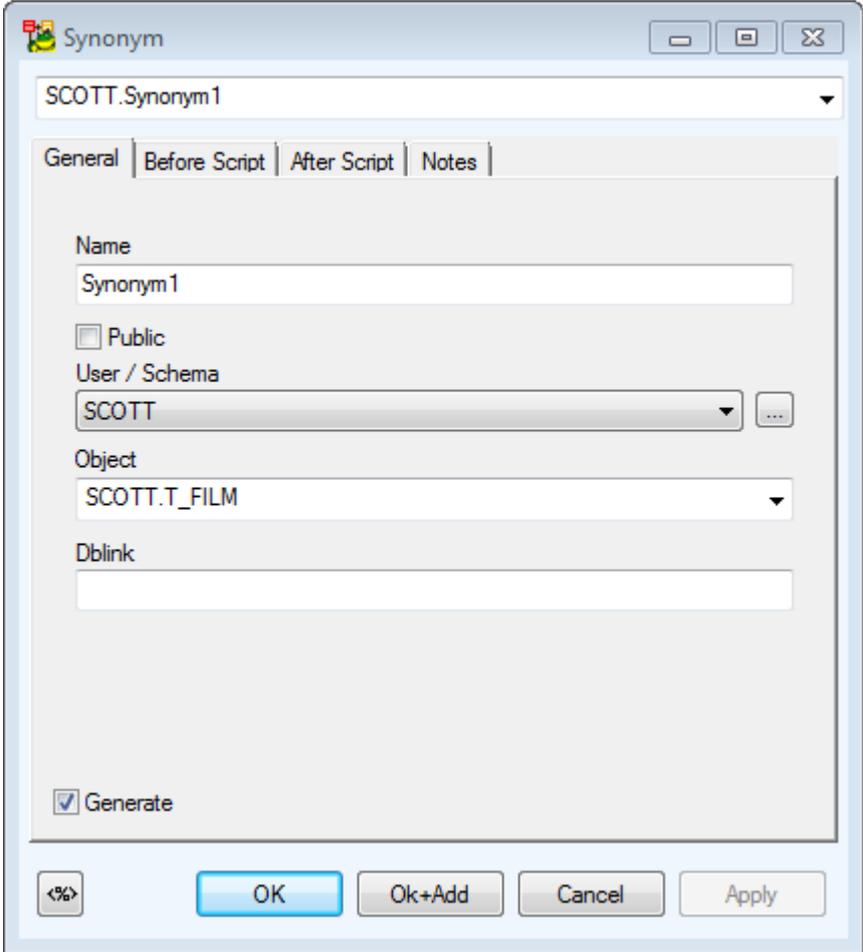
Cache  
 NOCACHE  
20

NOCYCLE  
 NOORDER

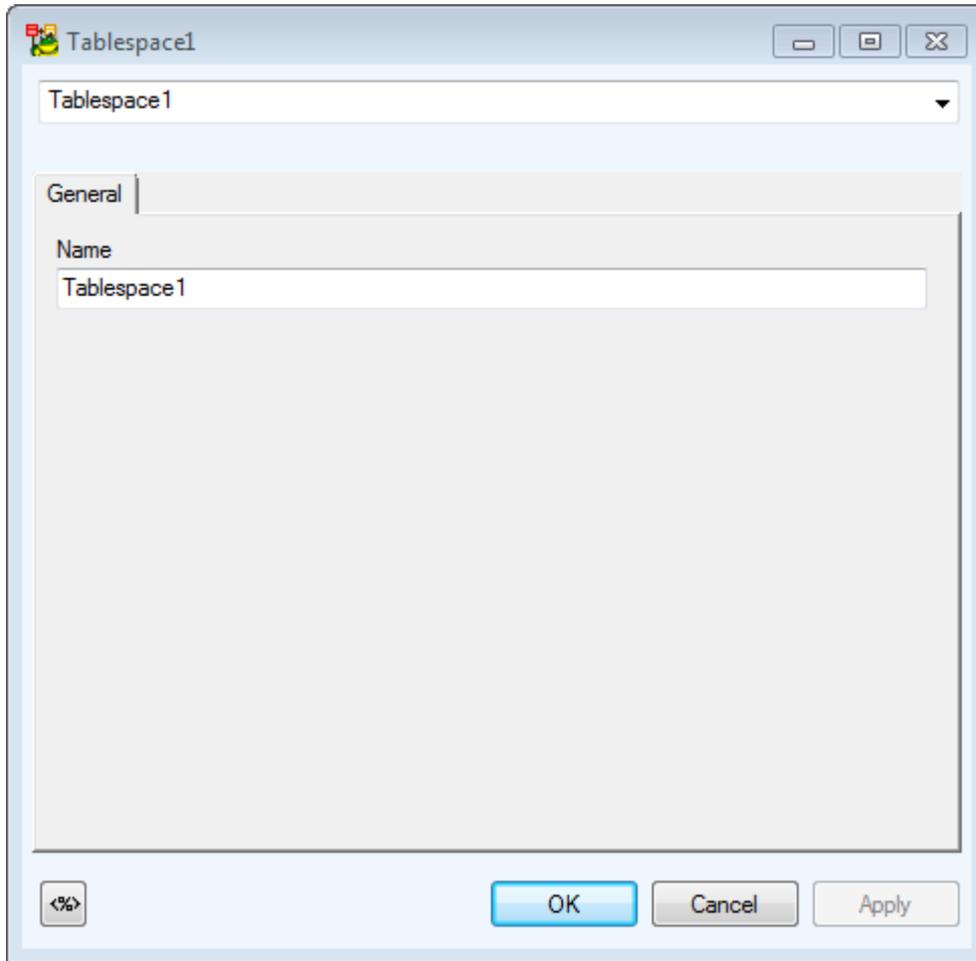
Generate  Generate SQL Only

<%> OK Cancel Apply

# Synonym



# Tablespaces

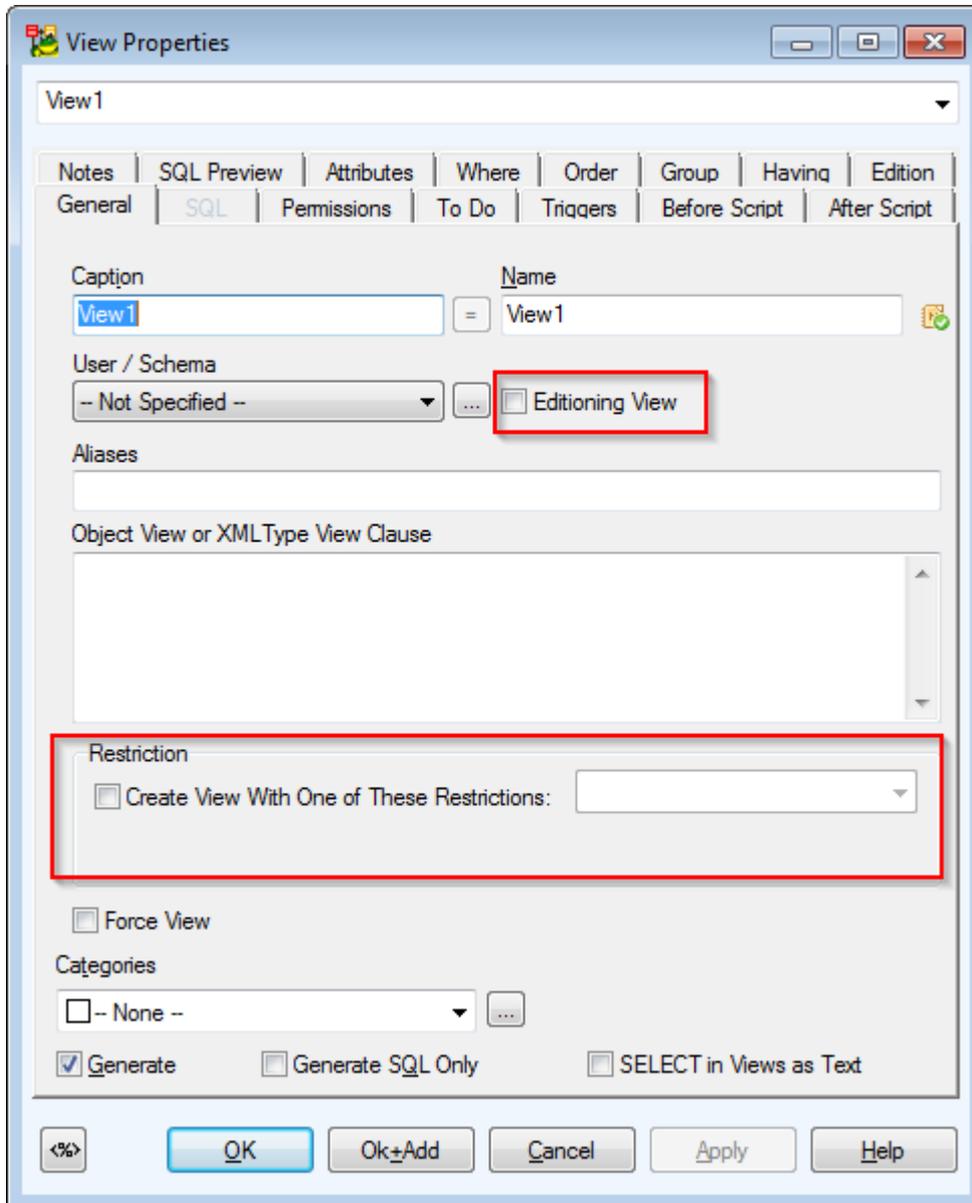


## Reverse Engineering - Oracle 11g Release 1

See [Reverse Engineering - Oracle 21c](#) for more information.

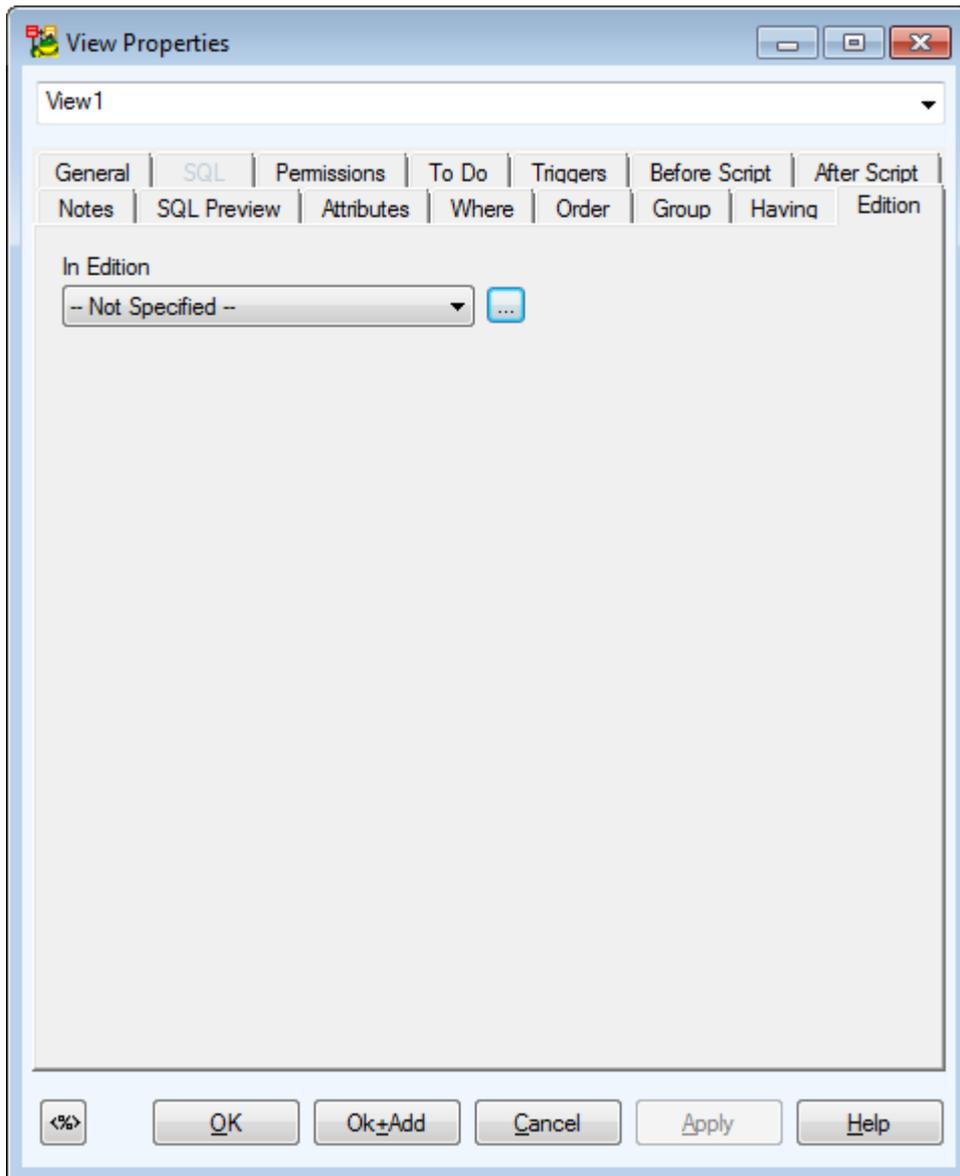
# Specifics - Oracle 11g Release 2

## View

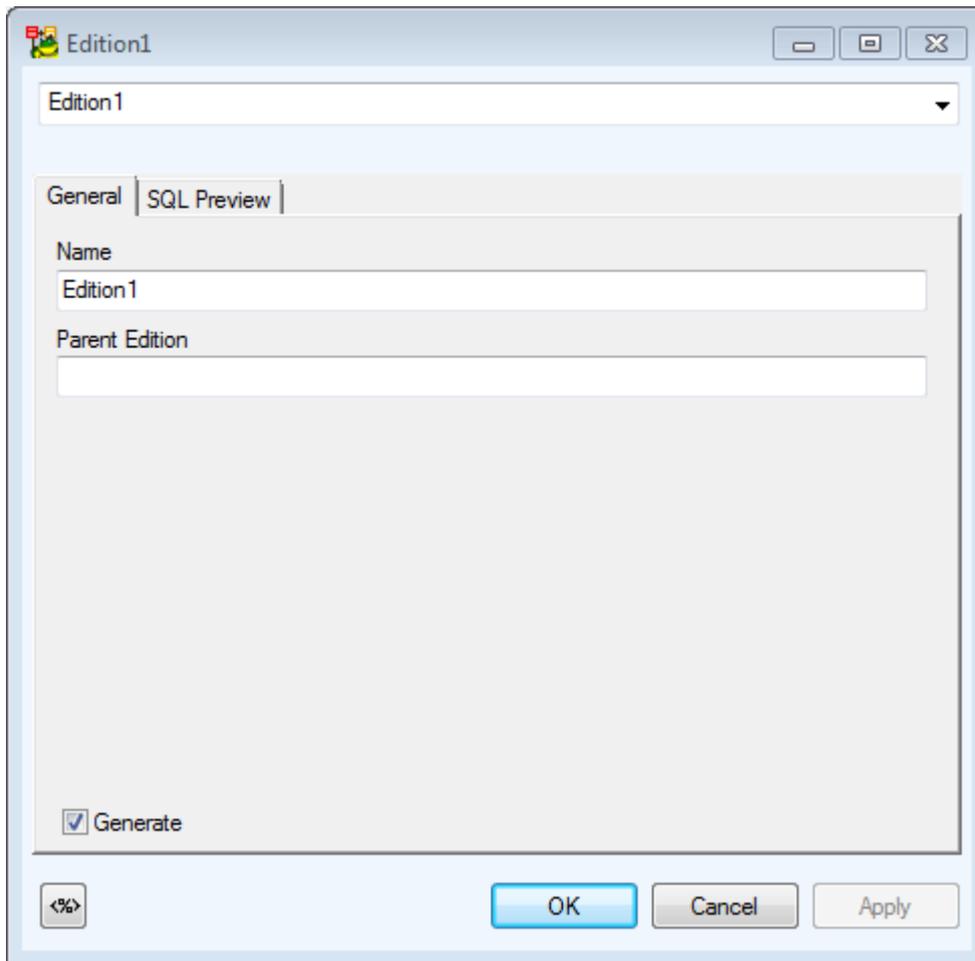


New parameter Editioning - **Edition View** checkbox.

**Restriction** area - You can define a constraint for restriction READ ONLY or WITH CHECK OPTION.



# Edition



The Edition object can be generated in DDL script and also loaded during reverse engineering.

The following objects can refer to the extra object **Edition**: Synonym, View, Function, Procedure, Package, Type, Trigger. See their **Properties** dialog and the new tab **Edition** | **In Edition** box.

## Edition - Reverse Engineering

**RE Wizard** | page **Options** | checkbox **Load Objects\* Only from Selected Edition**: and a box where you can type name of the Edition from which you want to load objects. During RE, all objects to which the defined edition refers are loaded. In the reversed model, the Edition is written on tab **Edition** in the **Properties** dialogs of these objects that have been newly defined or modified in the database for the Edition typed in the box.

## Edition - DDL Script Generation

**DDL Script Generation** dialog | **Detail Settings** tab | checkbox **Generate Change of Edition in Session**. Select this checkbox if you want Toad Data Modeler to take into account for the script generation an edition defined on tab **Edition** in **Properties** dialog of particular object.

Example of what will be generated in DDL script:

Before object definition:

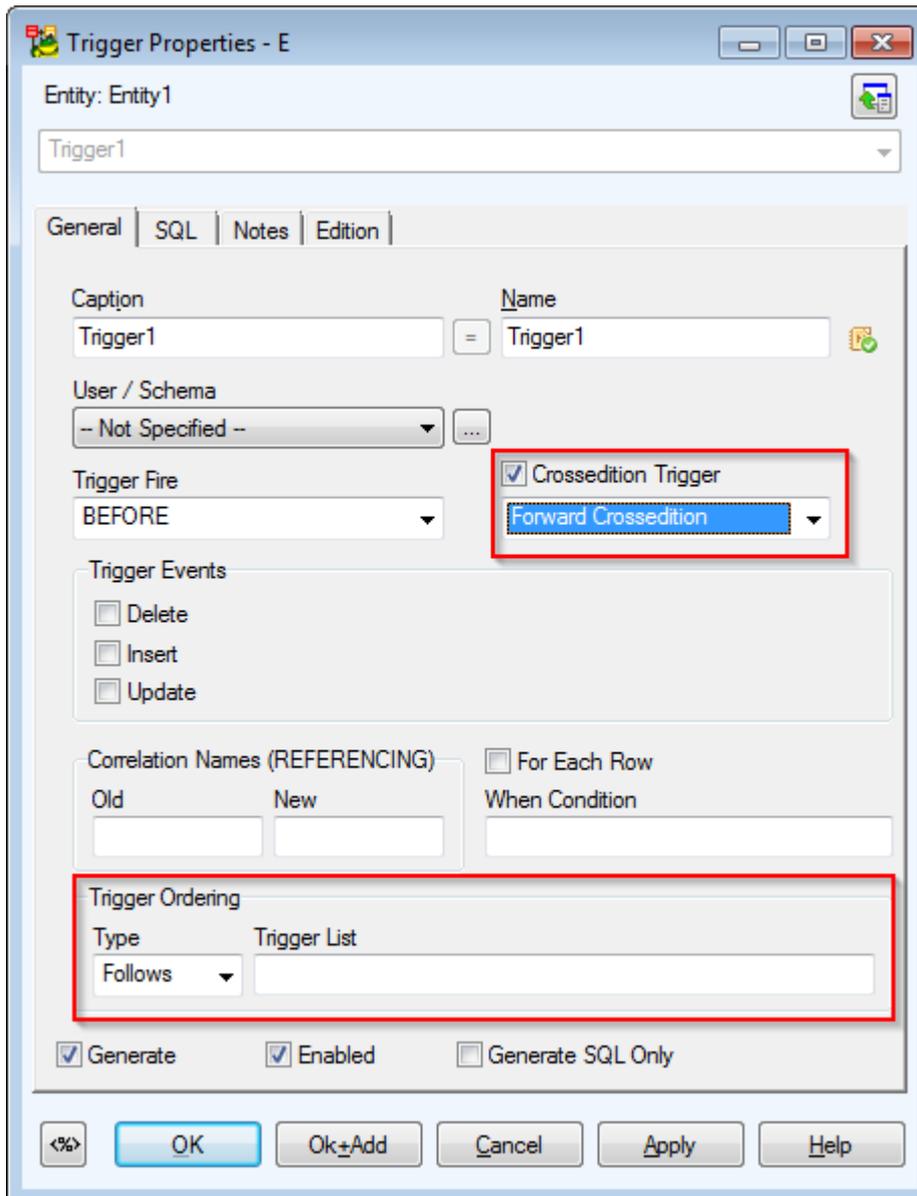
```
ALTER SESSION SET EDITION = object_edition
```

After object definition:

```
ALTER SESSION SET EDITION = ora$base
```

**i** Note: This is an example of default edition ora\$base. Default edition is defined in **Model | Model Properties** dialog | **Database Parameters** tab | **Default Edition Name** box.

## Trigger (Entity)



Parameter **CROSSEDITION** - Select the **Crossedition Trigger** checkbox to display a combo-box with options *Not Specified*, *Forward Crossedition*, *Reverse Crossedition*.

**Trigger Ordering** area - parameter *FOLLOWS* and *PRECEDES*. The **Type** combo-box offers options *Follows* or *Precedes*. In the **Trigger List** box, you can write a list of triggers (e.g. triggername1, triggername2).

## Trigger (View)

The screenshot shows the 'Trigger Properties - E' dialog box. The 'View' is set to 'View 1' and the selected trigger is 'Trigger1'. The 'General' tab is active, showing the following configuration:

- Caption:** Trigger1
- Name:** Trigger1
- User / Schema:** -- Not Specified --
- Trigger Fire:** COMPOUND
- Trigger Events:** Update (checked), Delete (unchecked), Insert (unchecked). The 'Column List' field is empty.
- Nested Table Column:** (empty)
- Correlation Names (REFERENCING):** For Each Row (checked), When Condition (empty). Old, New, and Parent fields are empty.
- Trigger Ordering:** Type: Follows, Trigger List (empty).
- Options:** Generate (checked), Enabled (checked), Generate SQL Only (unchecked).

From the **Trigger Fire** box, you can select items *COMPOUND*, *INSTEAD OF*, *BEFORE* and *AFTER*.

**Column List** edit box is available when the **Update** checkbox is selected in **Trigger Events** section and **Trigger Fire** option is set to either *COMPOUND*, *AFTER* or *BEFORE*.

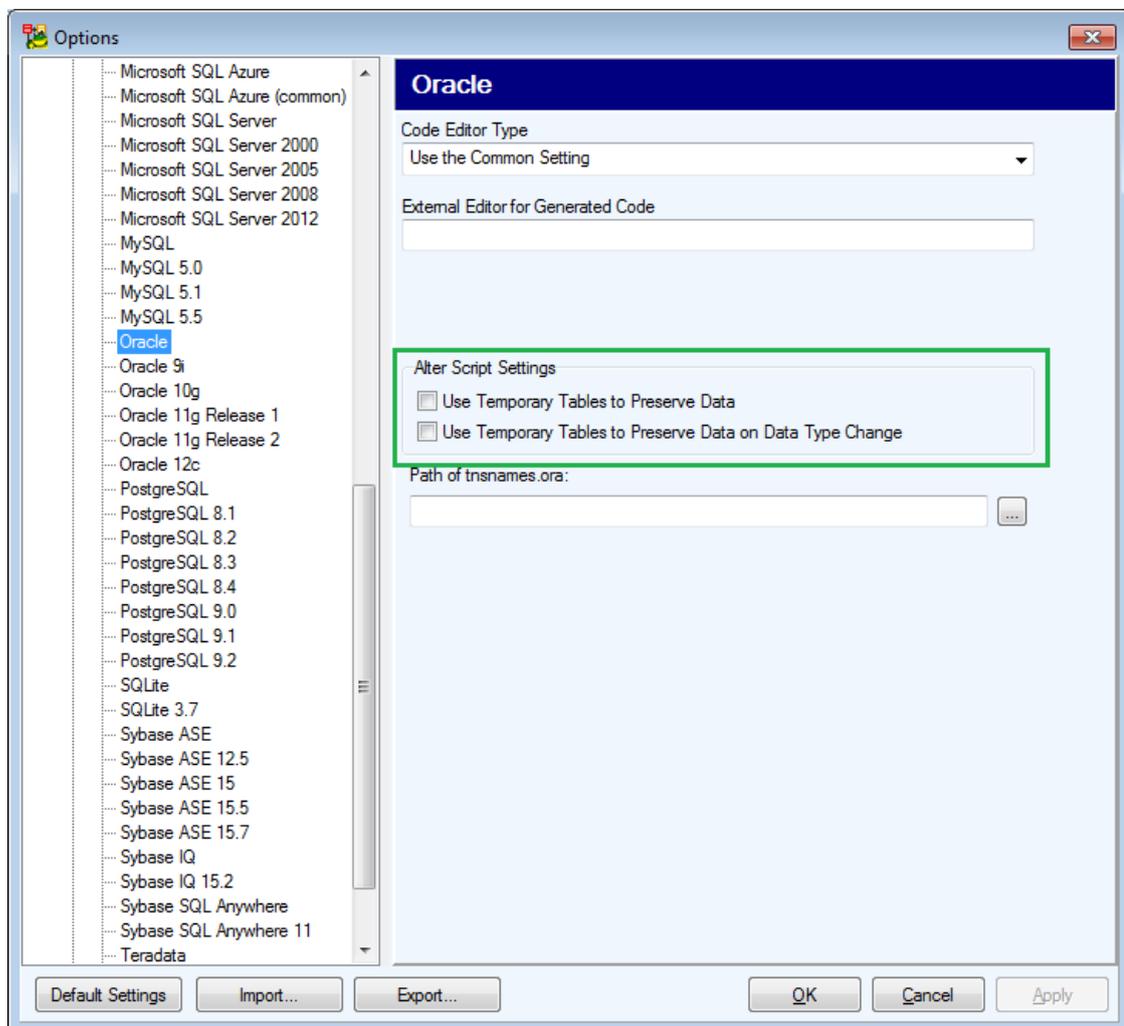
**For Each Row** checkbox and **When Condition** edit box are enabled based on the **Trigger Fire** option.

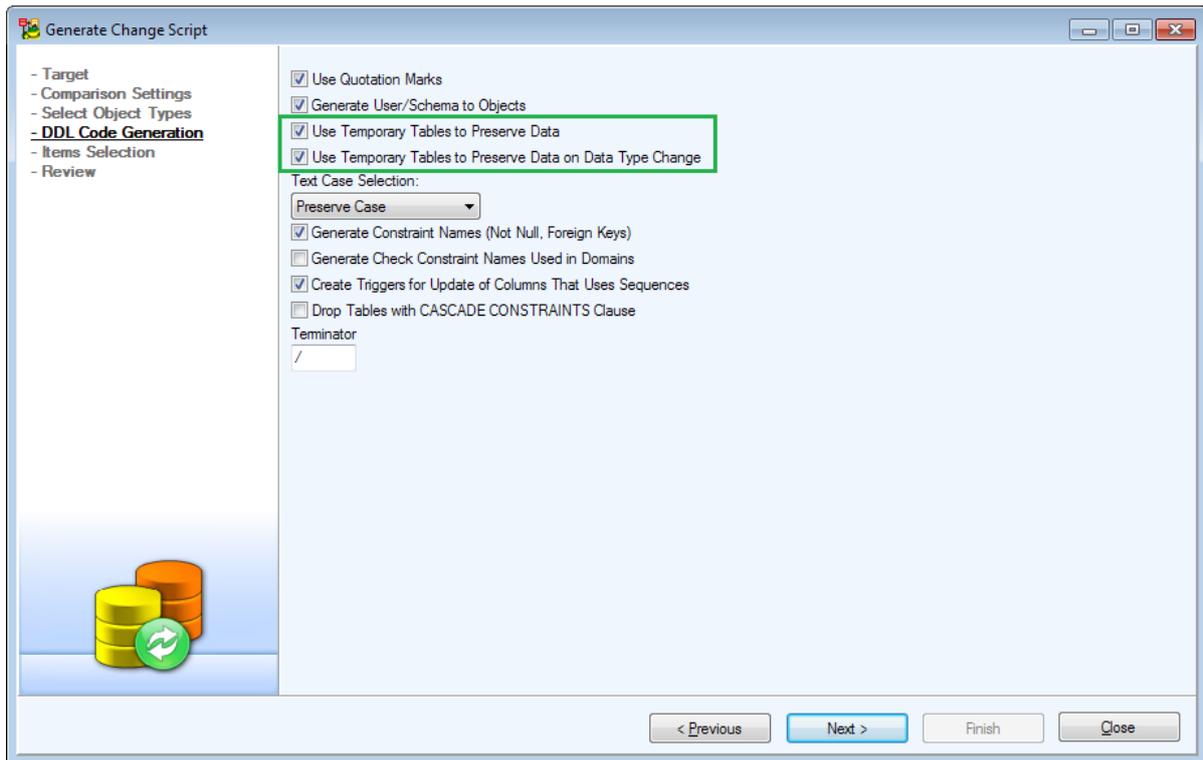
Other objects in Model Explorer:

- Edition
- Sequences
- Packages
- Directories
- Tablespaces
- ExtraObjects
- Java

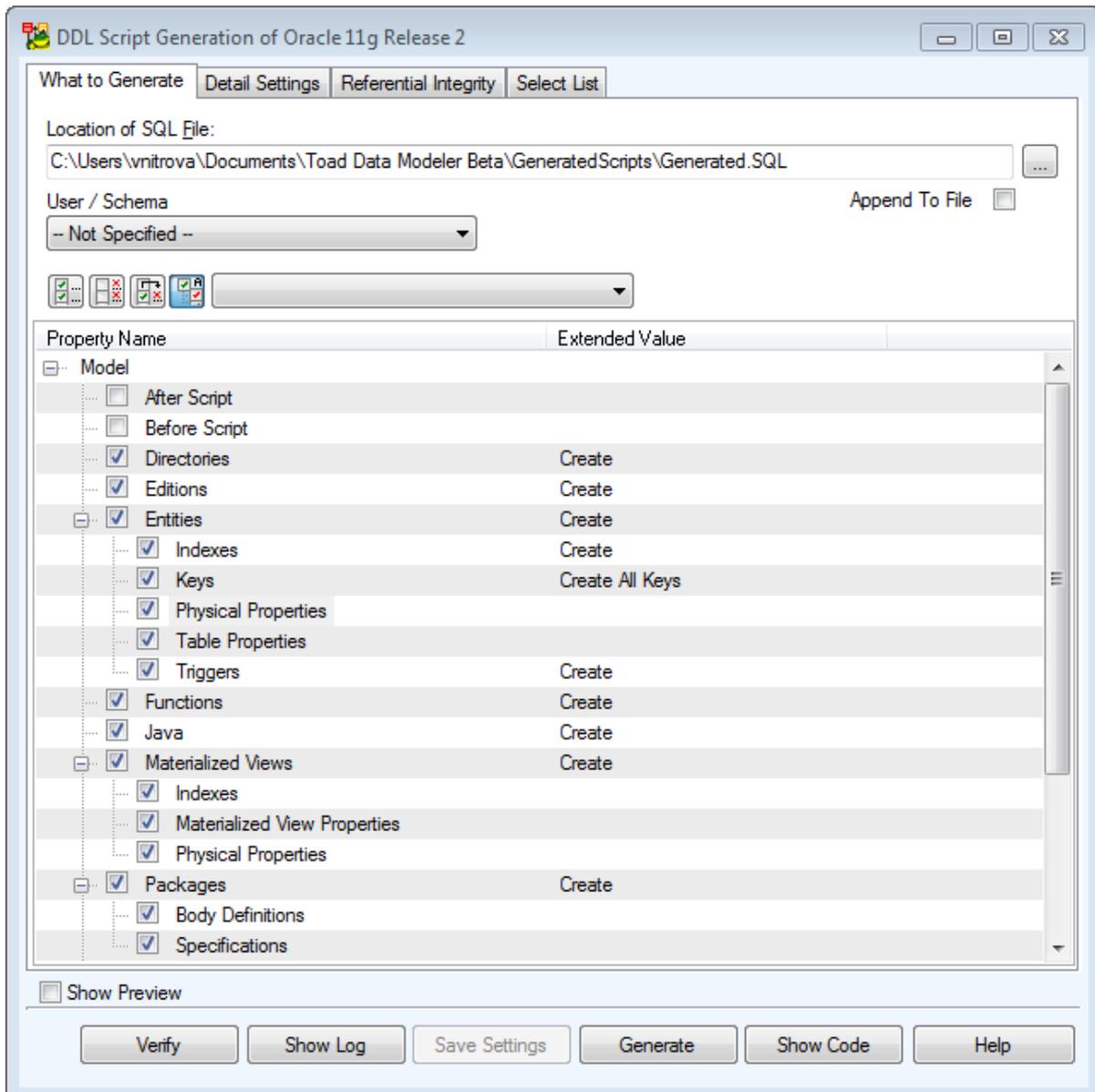
## Change Script

- **Change Script generation-** Use **Temporary Tables** options are now available for all Oracle Databases in Settings and in Change Script generator.





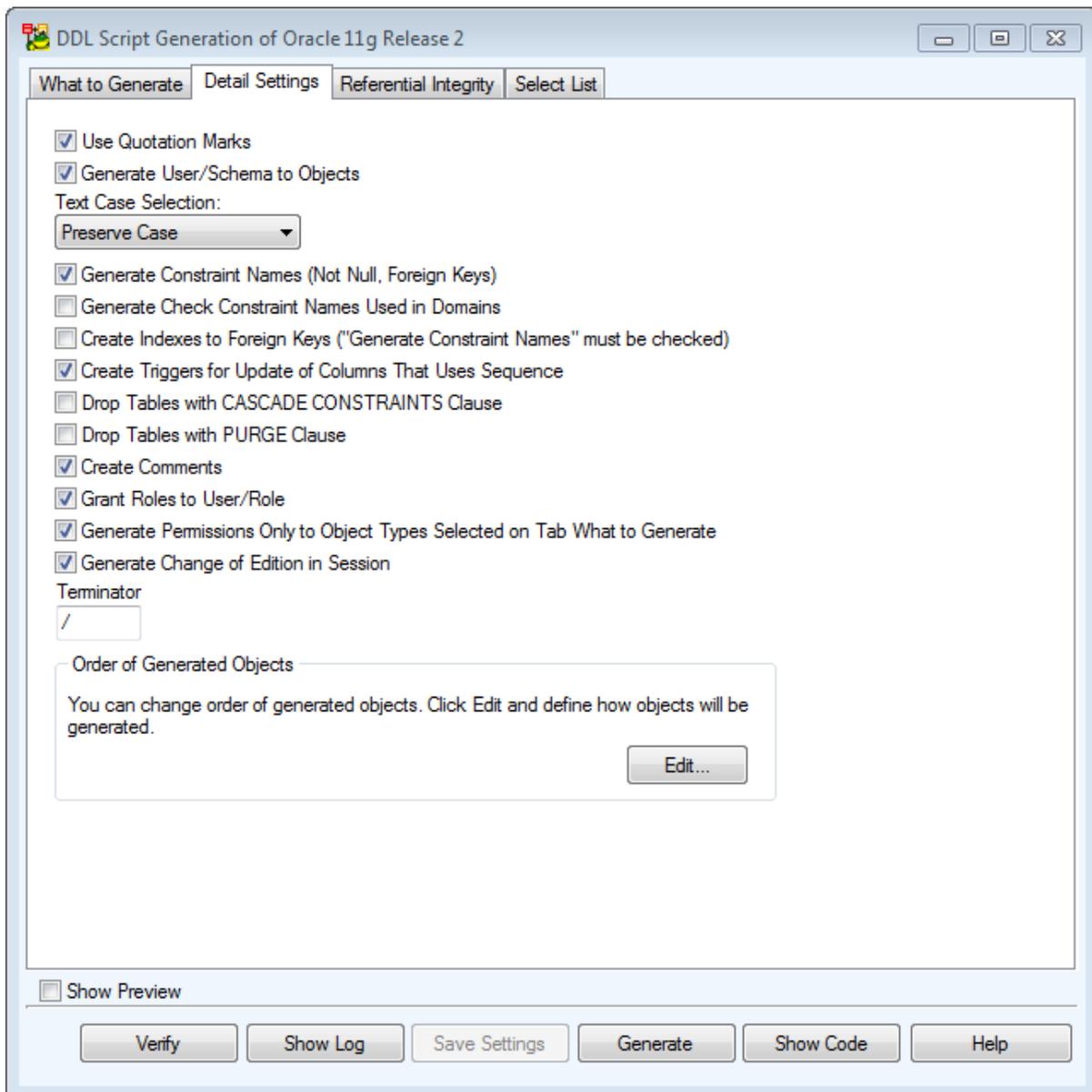
# Script Generation - Oracle 11g Release 2



On tab **What to Generate** see the **Editions** object.

User Data Types - Click the **Extended Values** column and see option CREATE OR REPLACE ... FORCE.

**i** Note: The option is used for CREATE OR REPLACE type. It permits to overwrite a type on which another type depends. If only CREATE OR REPLACE is selected, it leads to error notification.



**Detail Settings** tab | checkbox **Generate Change of Edition in Session**. Select this checkbox if you want Toad Data Modeler to include an edition defined on tab **Edition** in **Properties** dialog of particular object for the DDL script generation.

Example of what will be generated in DDL script:

Before object definition:

```
ALTER SESSION SET EDITION = object_edition
```

After object definition:

```
ALTER SESSION SET EDITION = ora$base
```

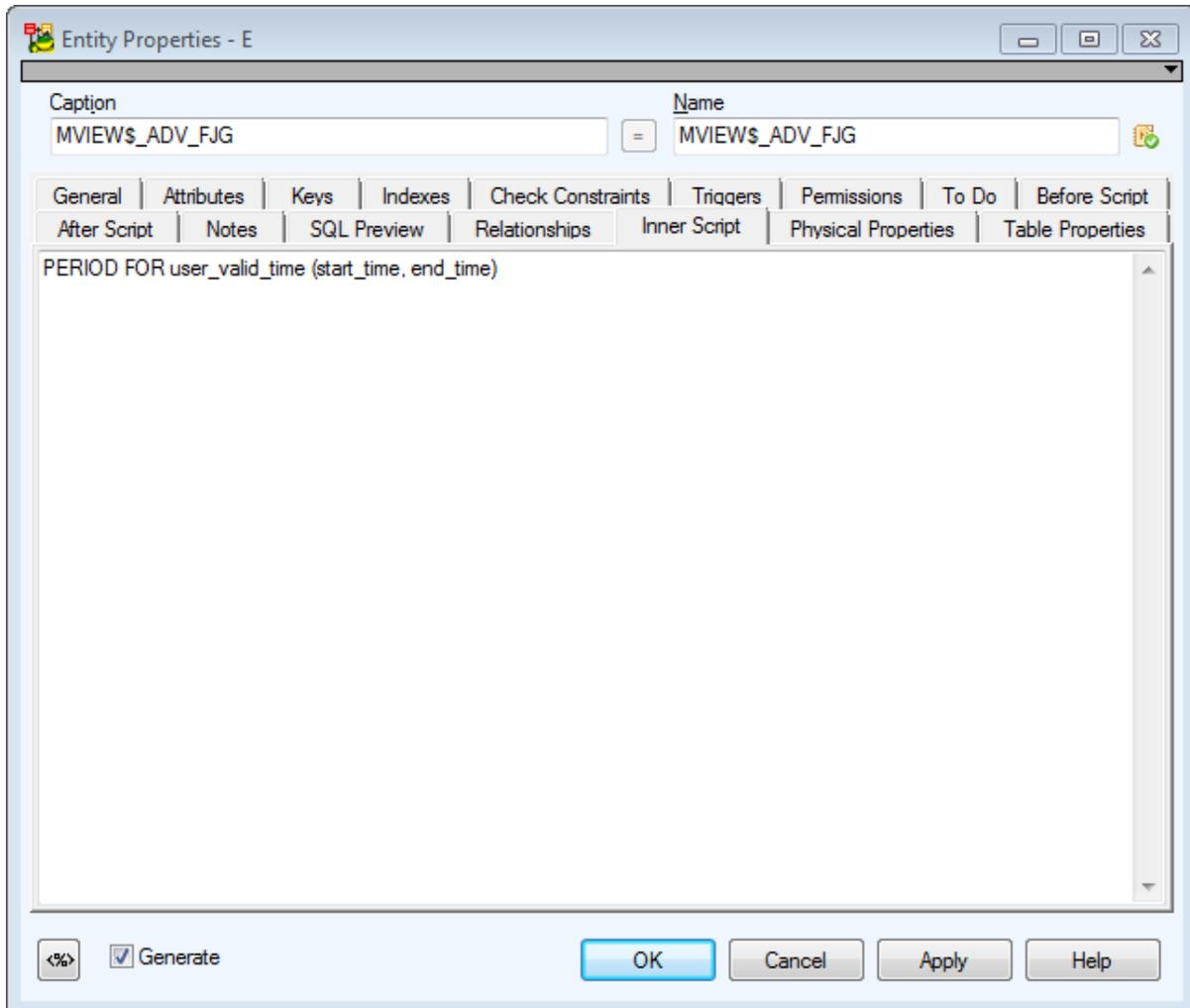
**i** Note: This is an example of default edition ora\$base. Default edition is defined in **Model | Model Properties** dialog | **Database Parameters** tab | **Default Edition Name** box.

# Reverse Engineering - Oracle 11g Release 2

See [Reverse Engineering - Oracle 21c](#) for more information.

# Specifics - Oracle 12c Release 1

## Entity



- **Temporal Validity** support for table  
**Entity Properties** dialog | **Inner Script** tab  
e.g. PERIOD FOR user\_valid\_time (start\_time, end\_time)
- *INDEXING ON/OFF* option for partitioned tables, for both whole table and individual partitions. **Table Properties** tab is used for this option. It is also loaded there during reverse engineering.
- Several *COMPRESS* parameters have been renamed.
- **Materialized Zonemap, Clustering, INMEMORY** and **[NO] ROW LEVEL LOCKING** options supported

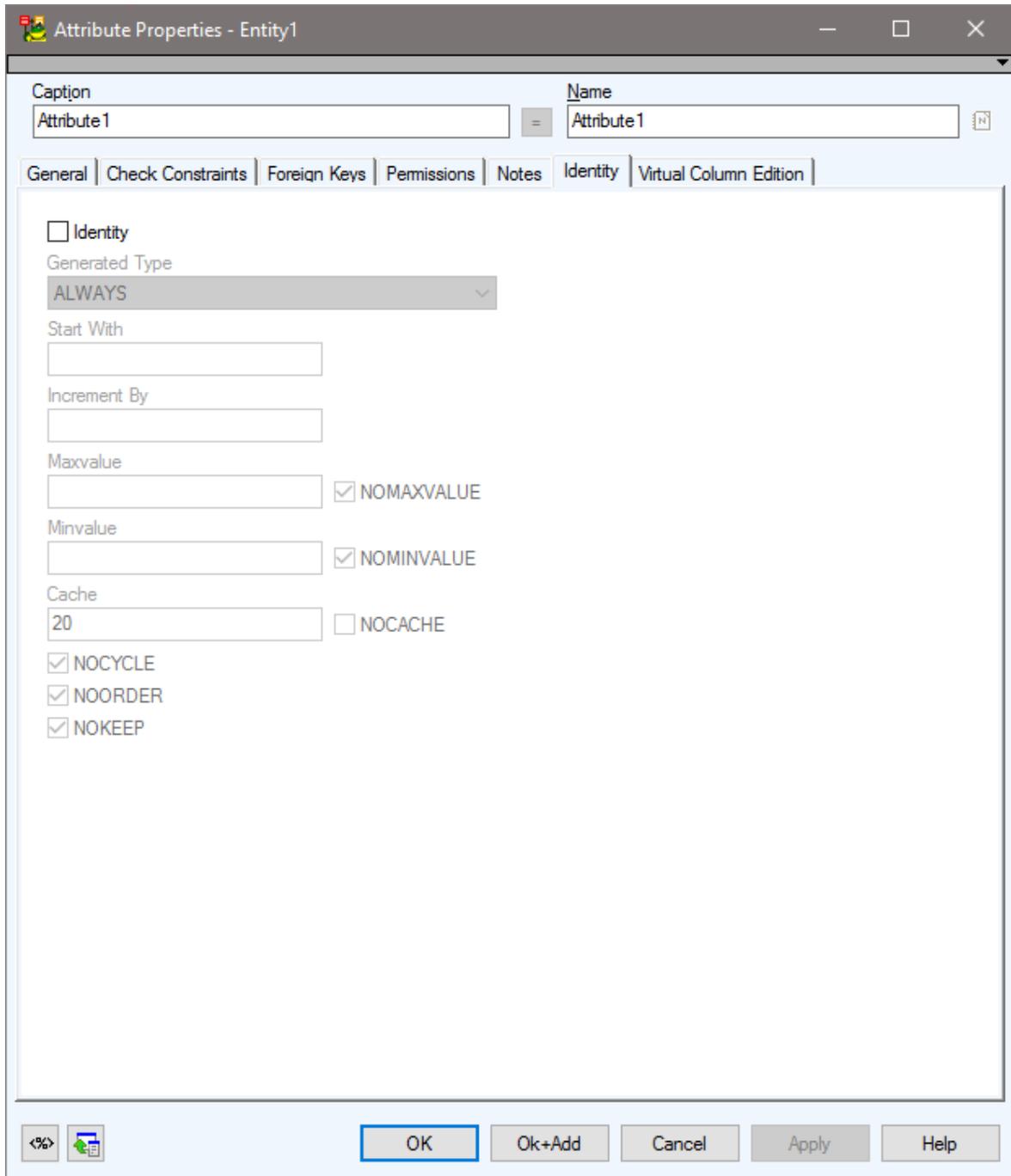
(added in **Oracle patch 12.1.0.2**)

- **Read permission supported (Oracle patch 12.1.0.2).**

## Attribute

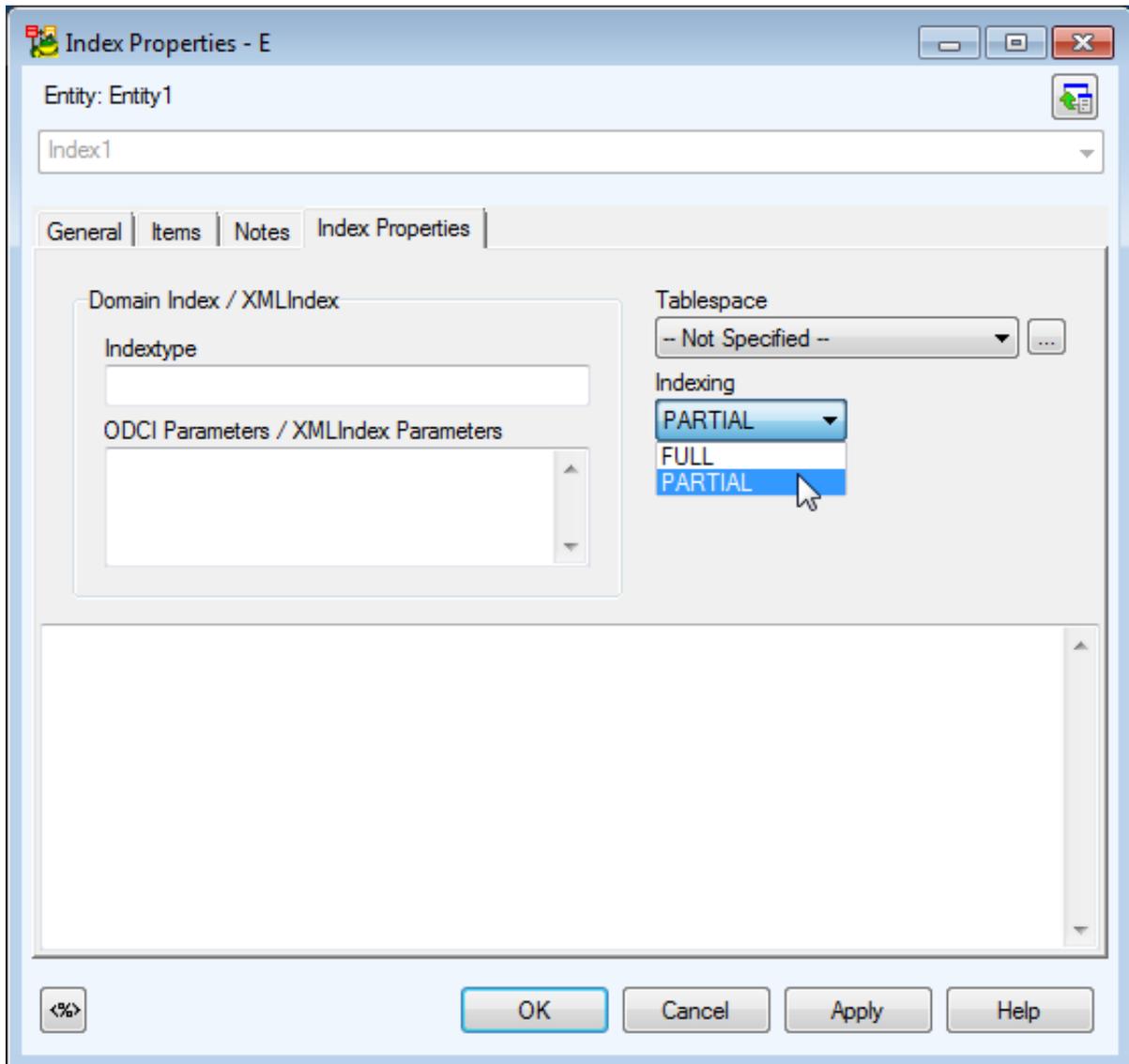
The screenshot shows the 'Attribute Properties - Entity1' dialog box. The 'Caption' is 'Attribute1' and the 'Name' is 'Attribute1'. The 'Data Type' is 'Char(x)' with a length of 20. The 'Column Length in' is set to 'Default'. The 'Default Value' is empty, and the 'On Null' checkbox is checked and highlighted with a red box. The 'Default Rule' is '-- None --'. The 'Primary Key', 'Not Null', and 'Unique' checkboxes are unchecked. The 'Comment' field is empty. The 'Not Null Constraint' section has a 'Constraint Name' field and several unchecked checkboxes: 'Deferrable', 'Initially Deferred', 'Disable', 'No Validate', and 'Rely'. The 'Virtual Column' section has an unchecked 'Is Virtual Column' checkbox and an empty 'Expression' field. The 'Other Database Specifics' section has a 'Used Sequence (trigger)' dropdown set to '-- None --', an empty 'Encryption Specification' field, and an unchecked 'Invisible' checkbox highlighted with a red box. The 'REF Type' section has an unchecked 'Use REF Type' checkbox and an empty 'Inline REF Constraint' field. The bottom of the dialog has buttons for '<%', 'OK', 'Ok+Add', 'Cancel', 'Apply', and 'Help'.

- **Visible/Invisible** option
- *DEFAULT ON NULL* expression - **On Null** checkbox
- Data Types VARCHAR2(x), NVARCHAR2(x) and RAW(size) can acquire length up to 32767 bytes (formerly only 4000).



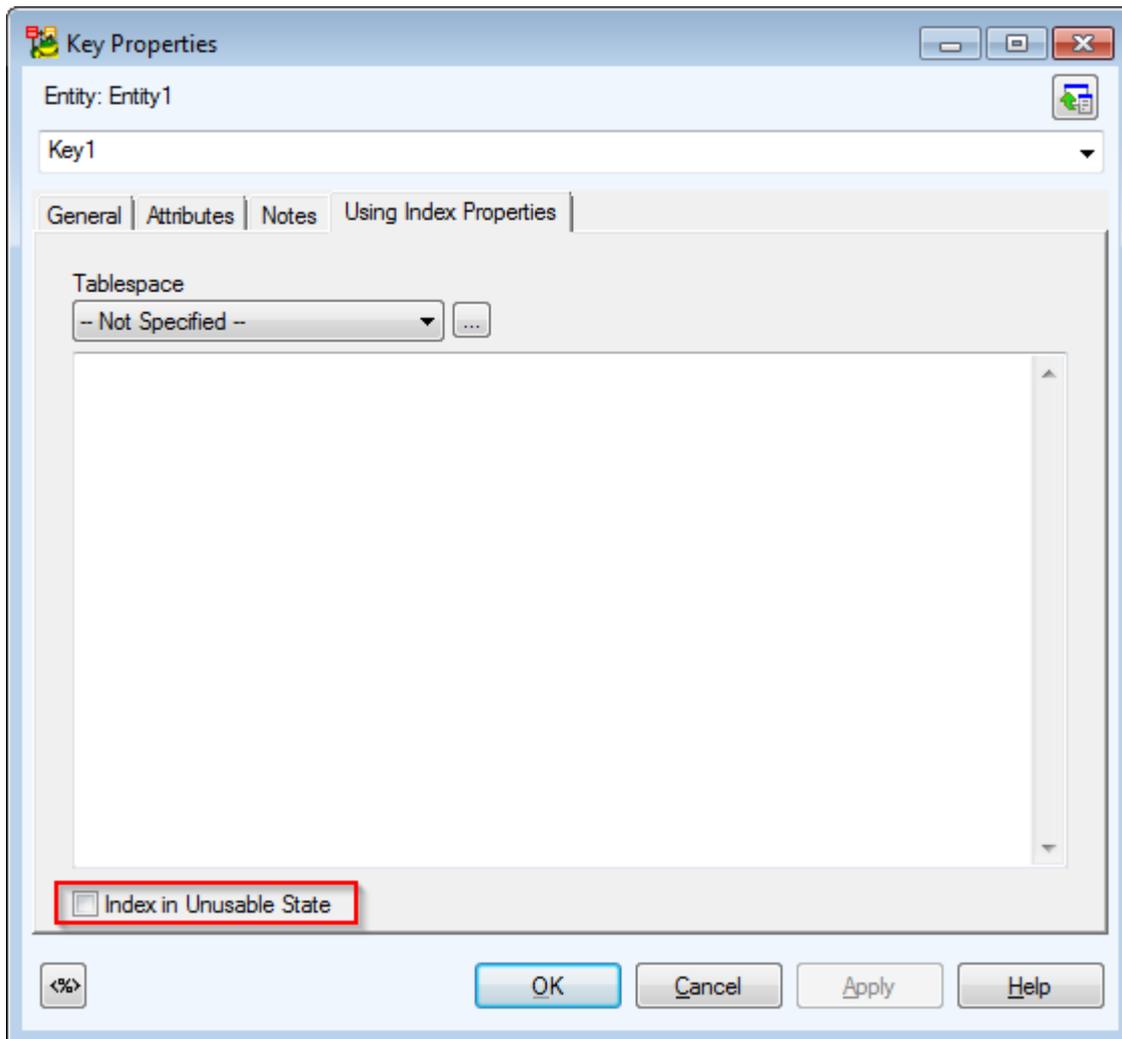
- New **Identity** tab
- Set NOKEEP in **Attribute Properties | Identity**
- **Virtual Column Edition** tab where you can define *EDITION* for virtual column. See the comboboxes **Unusable Before Edition** and **Unusable Beginning Edition**.

## Index



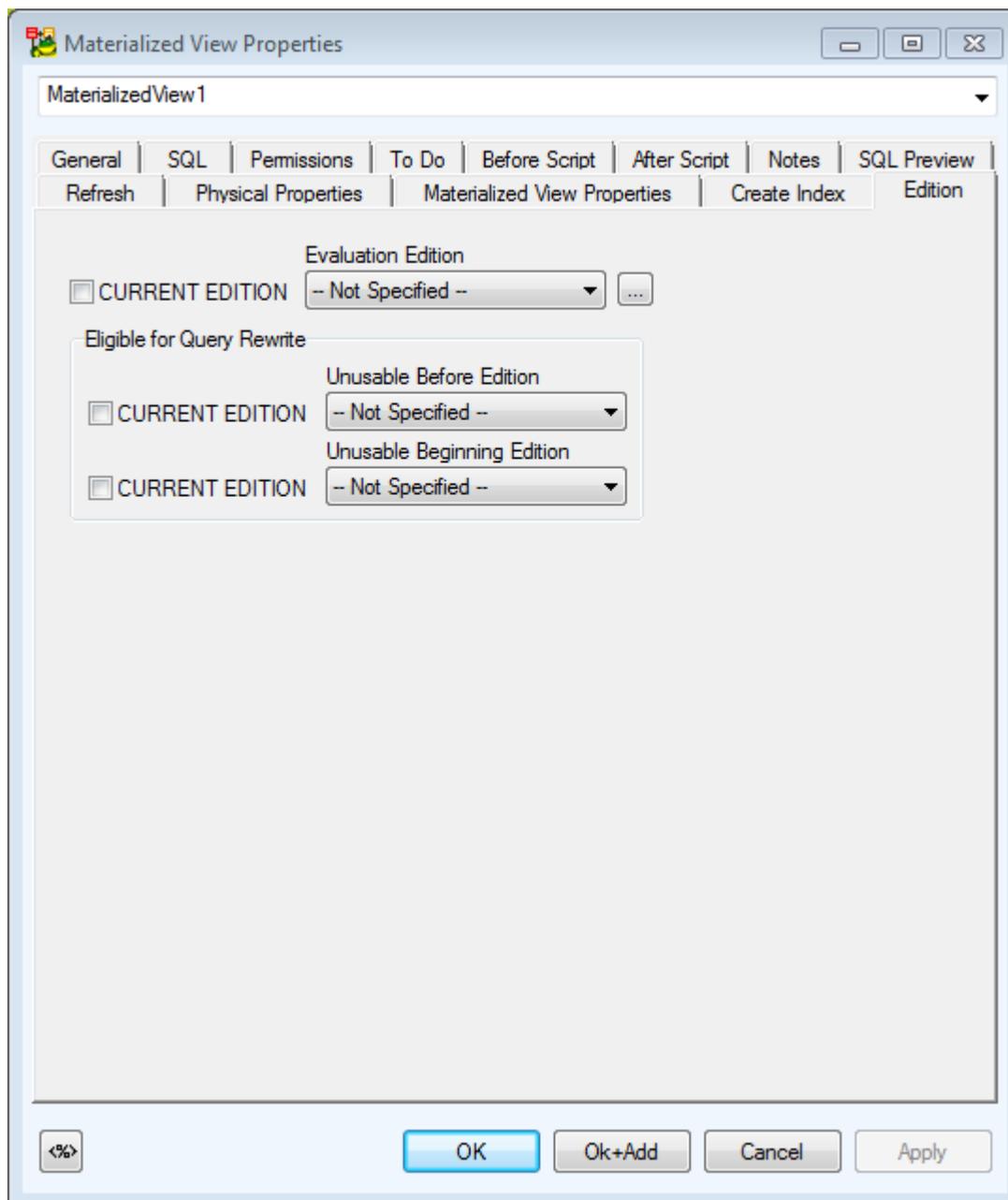
- New option *INDEXING PARTIAL/FULL* for indexes of partitioned tables - **Indexing** combobox.
- For index and its partitions it is possible to set *USABLE* (besides *UNUSABLE*) in the text field.

# Keys



**Key Properties** dialog | **Using Index Properties** tab | **Index in UNUSABLE state** checkbox where you can set **USABLE** (besides UNUSABLE) option.

# Materialized View

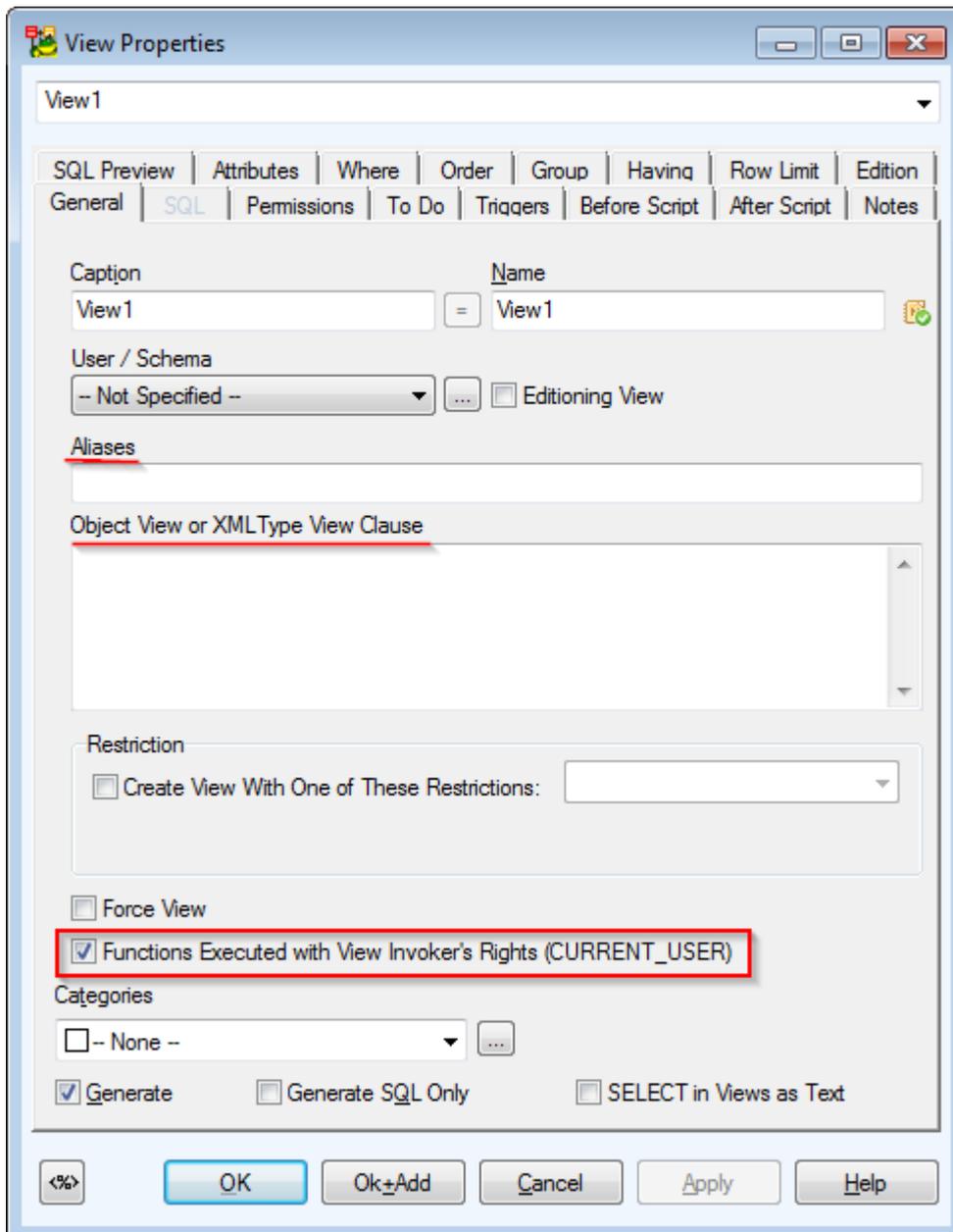


- **UNUSABLE EDITION** option can be defined in two new comboboxes **Unusable Before Edition** and **Unusable Beginning Edition** on tab **Edition**.
- **INMEMORY** and **[NO] ROW LEVEL LOCKING** options supported (added in **Oracle patch 12.1.0.2**)
- **Read** permission supported (**Oracle patch 12.1.0.2**).

# Function, Procedure, Package, User Data Type, Synonym, View, Trigger

- **Edition** tab where you can define a function as *NONEDITIONABLE* in checkbox **Noneditionable**.
- *White List (ACCESSIBLE BY)* supported

## View



- *VISIBLE/INVISIBLE* option can be defined for alias items (textually in **Aliases** box)
- Select the **Functions Executed with View Invoker's Rights (CURRENT\_USER)** checkbox to define the *BEQUEATH CURRENT\_USER/DEFINER* option to specify whether functions referenced in the view are executed using the view invoker's rights or the view definer's rights.
- *STORE ALL VARRAYS AS LOBS/TABLES* option for XMLType views are loaded to **Object View or XMLType View Clause** box on tab **General**.
- **Read** permission supported (**Oracle patch 12.1.0.2**).
- **JSON functions** now supported (**Oracle patch 12.1.0.2**).

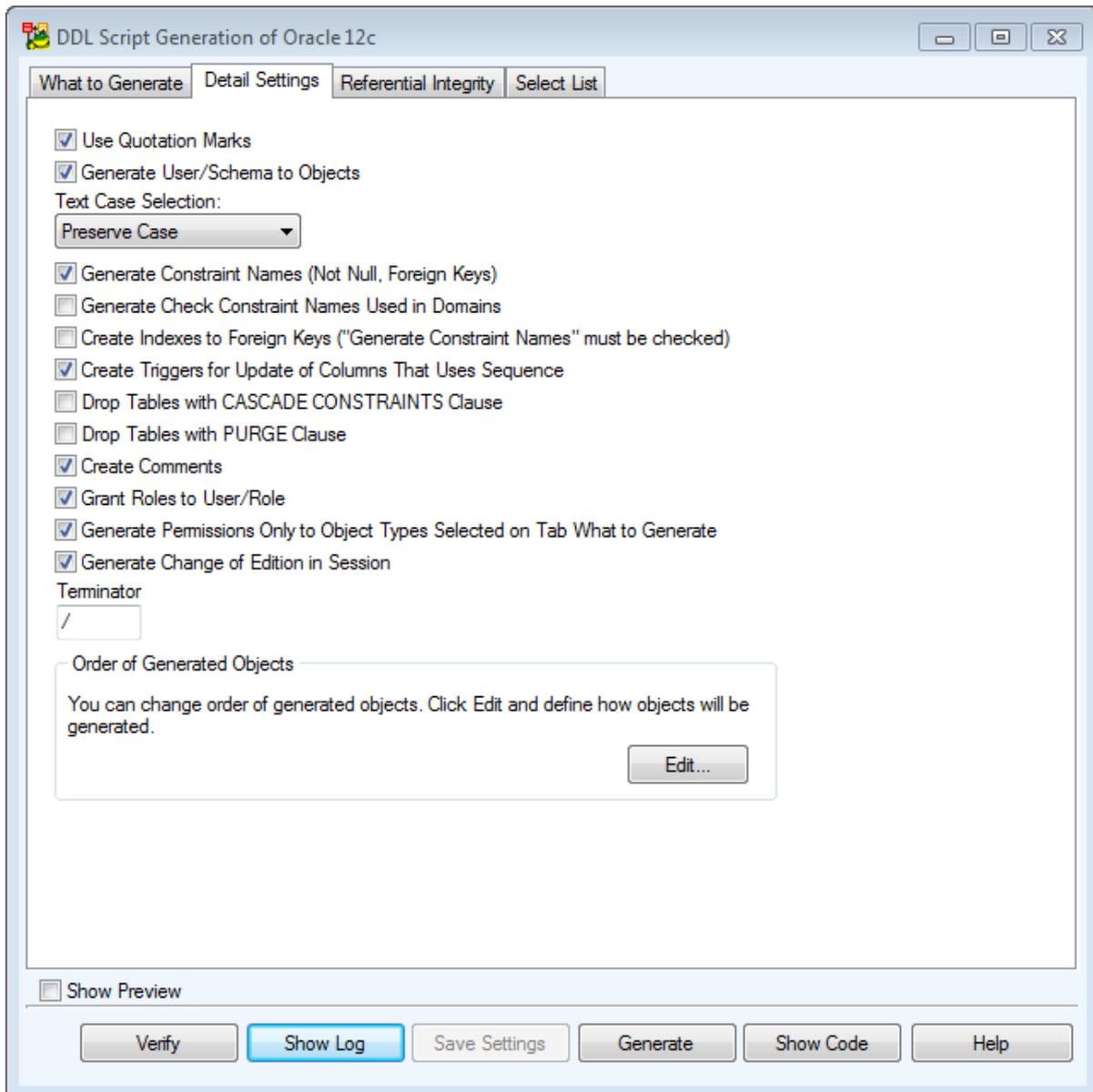
## Sequence

- *KEEP/NOKEEP* select the **NOKEEP** checkbox.

# Reverse Engineering - Oracle 12c Release 1

See [Reverse Engineering - Oracle 21c](#) for more information.

# Script Generation - Oracle 12c Release 1

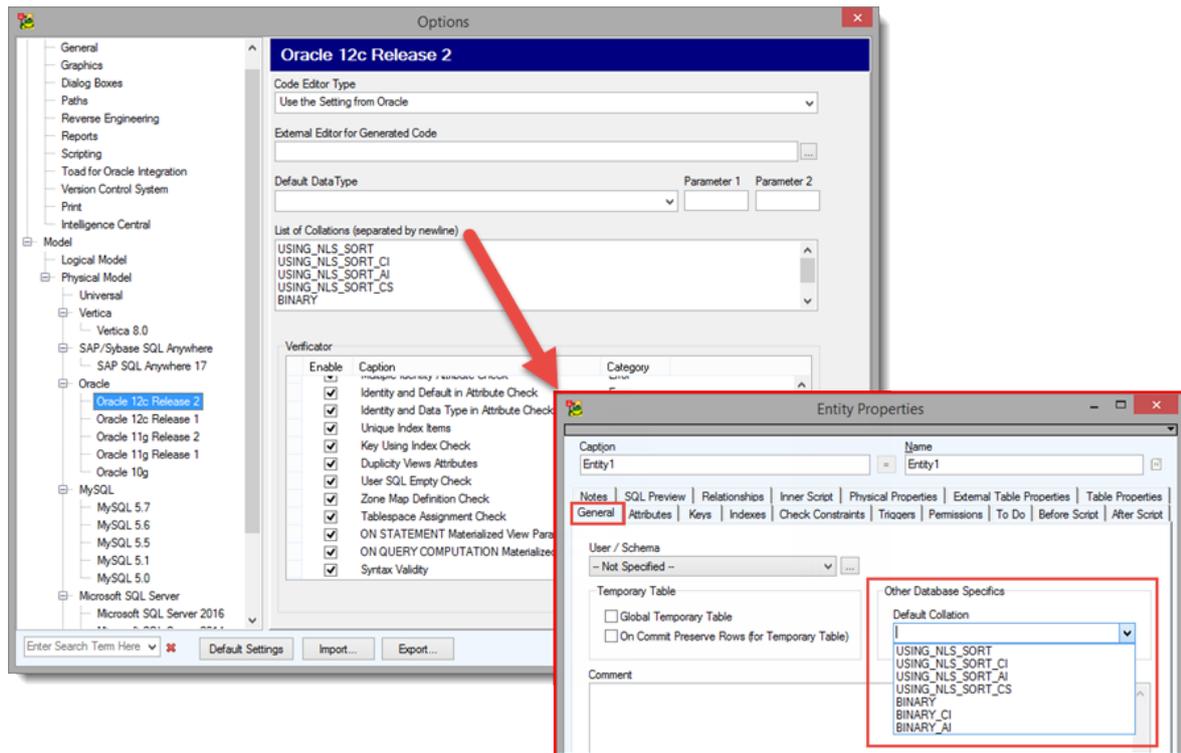


# Specifics - Oracle 12c Release 2

- Options are available in **Options | Model | Physical Model | Oracle | Oracle 12c Release 2**
- Identifiers now can be 128 characters long (exception - Tablespaces)

## Support for Collation

- New property COLLATION can now be used for tables (column collation), views, materialized views, users, packages, procedures, functions and types
- **List of Collations** displays objects



## Tables

### Table Properties and Physical Properties Tab

- In **Entity Properties | Physical Properties | Segment Creation** select Not Specified, Deferred or Immediate

The following properties' definitions need to be entered manually into the text field and are loaded automatically during **Reverse Engineering**:

- FOR SERVICE property of INMEMORY tables - can be defined for the whole tables, for their partitions and subpartitions

- Inmemory\_column\_clause for NO INMEMORY tables - you now can specify it to enable or disable specific table columns for the IM column store, and specify the data compression method for specific columns
- READ ONLY or READ WRITE property - can be defined for the whole tables, for their partitions and subpartitions
- New In-Memory Column Store policy
- Compression policy - new option COLUMN STORE COMPRESS FOR QUERY ROW AFTER ilm\_time\_period OF NO MODIFICATION
- New properties COMPRESSION and INDEXING can be used for subpartition templates
- Automatic list partitions can now be used for partitions
- Multi-Column List Partitioning can now be used for partitions and subpartitions
- List of value lists can be specified for multi-column list partitions

### External Tables

- Now you can use NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY constraints for external tables
- Can use ENCRYPT property, Virtual Column, range/list partitions and subpartitions
- Do not have to use DEFAULT and LOCATION in their definitions
- In **Entity Properties | External Table Properties** you can use one of the following drivers:
  - ORACLE\_LOADER, ORACLE\_DATAPUMP, ORACLE\_HDFS, and ORACLE\_HIVE

### Misc.

- New Encrypt Algorithms are used for columns

### Indexes

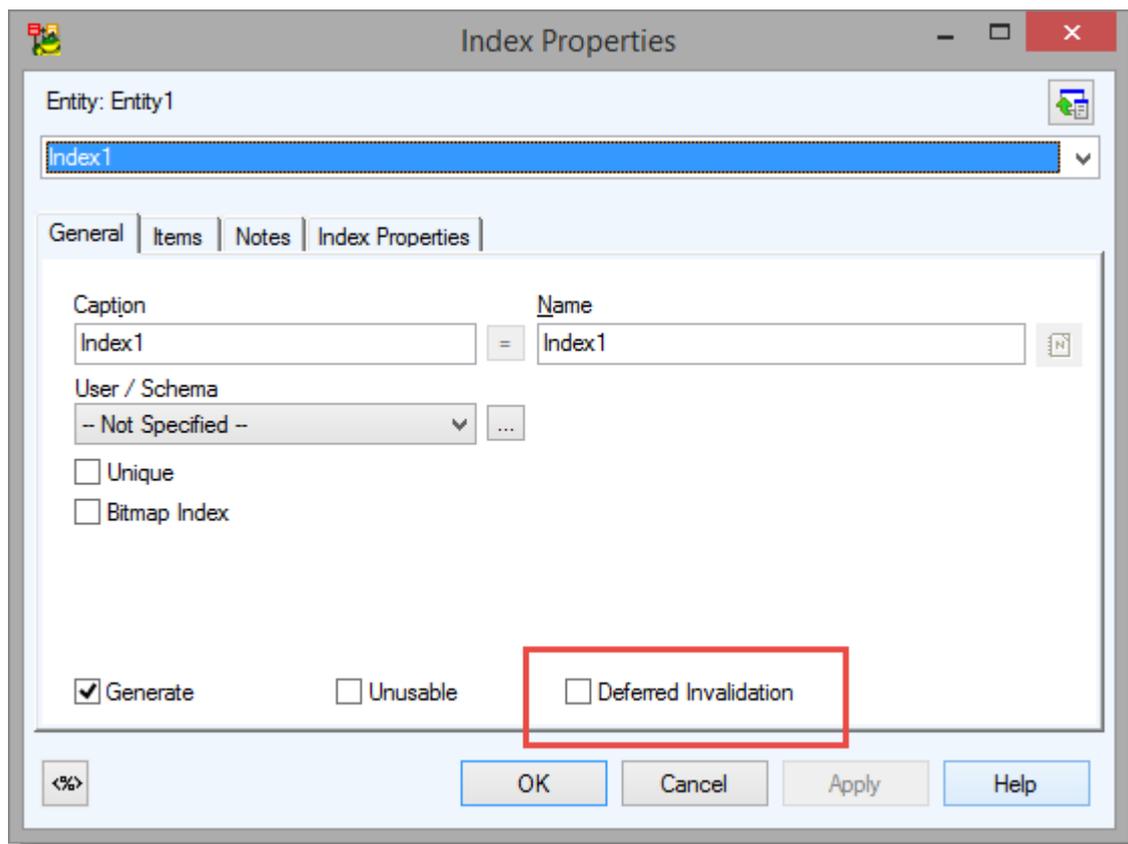
#### Index Properties Tab

The following property definition needs to be entered manually into the text field and is loaded automatically during **Reverse Engineering**:

- New Compress parameters COMPRESS ADVANCED and COMPRESS ADVANCED HIGH - can be defined for an index and its individual partitions (global and local)

### Misc.

- New option DEFERRED INVALIDATION is available. *Default is IMMEDIATE INVALIDATION*



- Column COLLATION is recognized during Reverse Engineering

## Keys

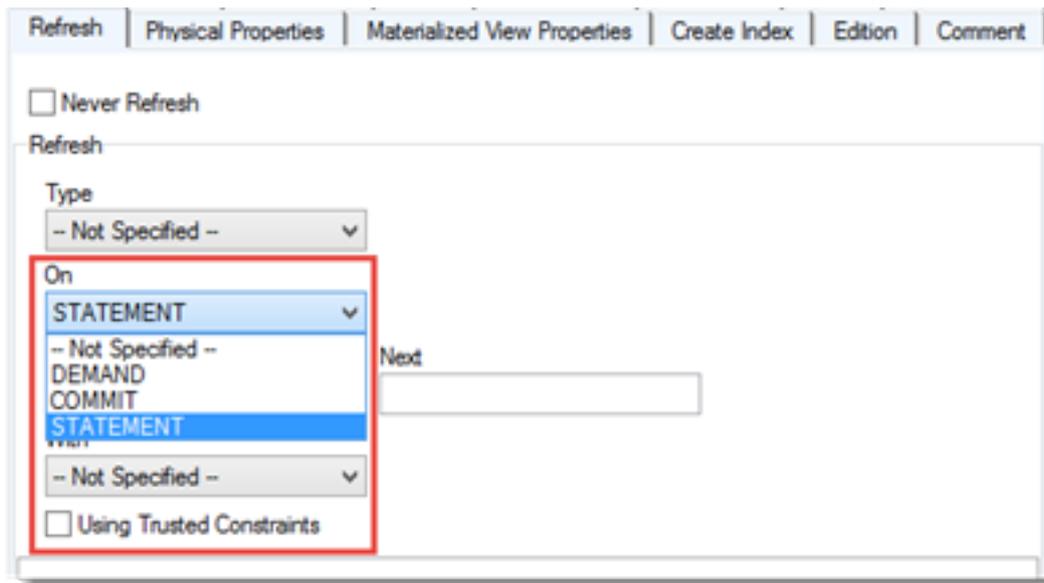
### Using Index Properties Tab

The following property definition needs to be entered manually into the text field and is loaded automatically during **Reverse Engineering**:

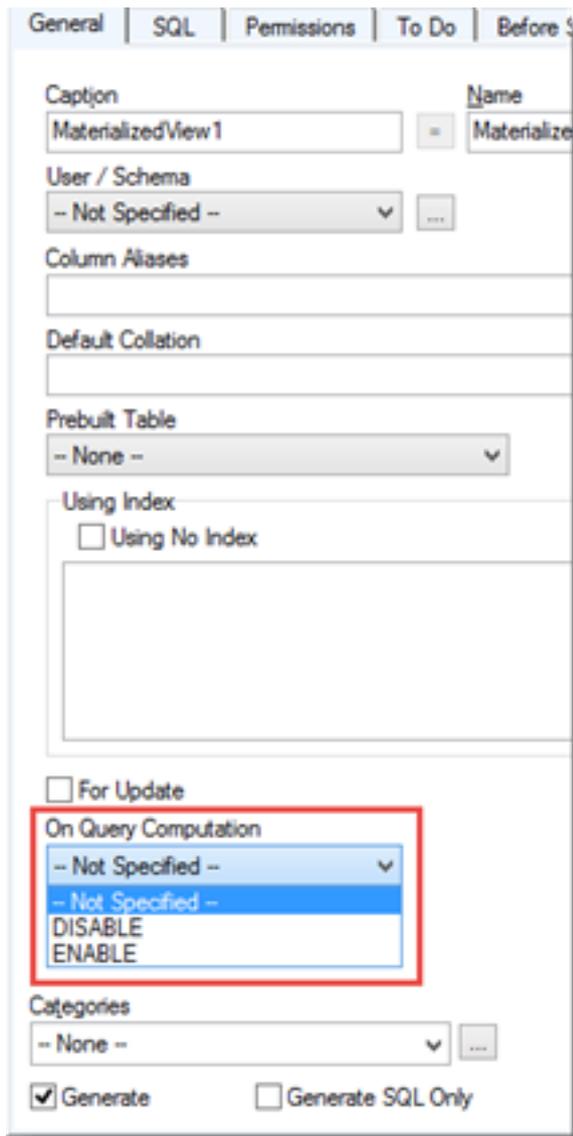
- New Compress parameters COMPRESS ADVANCED and COMPRESS ADVANCED HIGH

### Materialized Views

- New property FOR SERVICE - can be defined for the whole tables, for their partitions and subpartitions; the definition needs to be entered manually into the text field (**Physical Properties Tab**) and is loaded automatically during **Reverse Engineering**:
- New Encrypt Algorithms - can be set in the **Column Aliases** field
- New ON STATEMENT mode:
  - Refresh type needs to be FAST
  - The materialized view's defining query needs to include the ROWID column of the fact table.
  - ON STATEMENT mode cannot be converted to a different mode using ALTER MATERIALIZED VIEW but instead DROP/CREATE is performed during **Change Script Generation**



- New property ENABLE ON QUERY COMPUTATION - *by default: DISABLE ON QUERY COMPUTATION*
  - If enabled, the refresh mode COMMIT cannot be used



### **User**

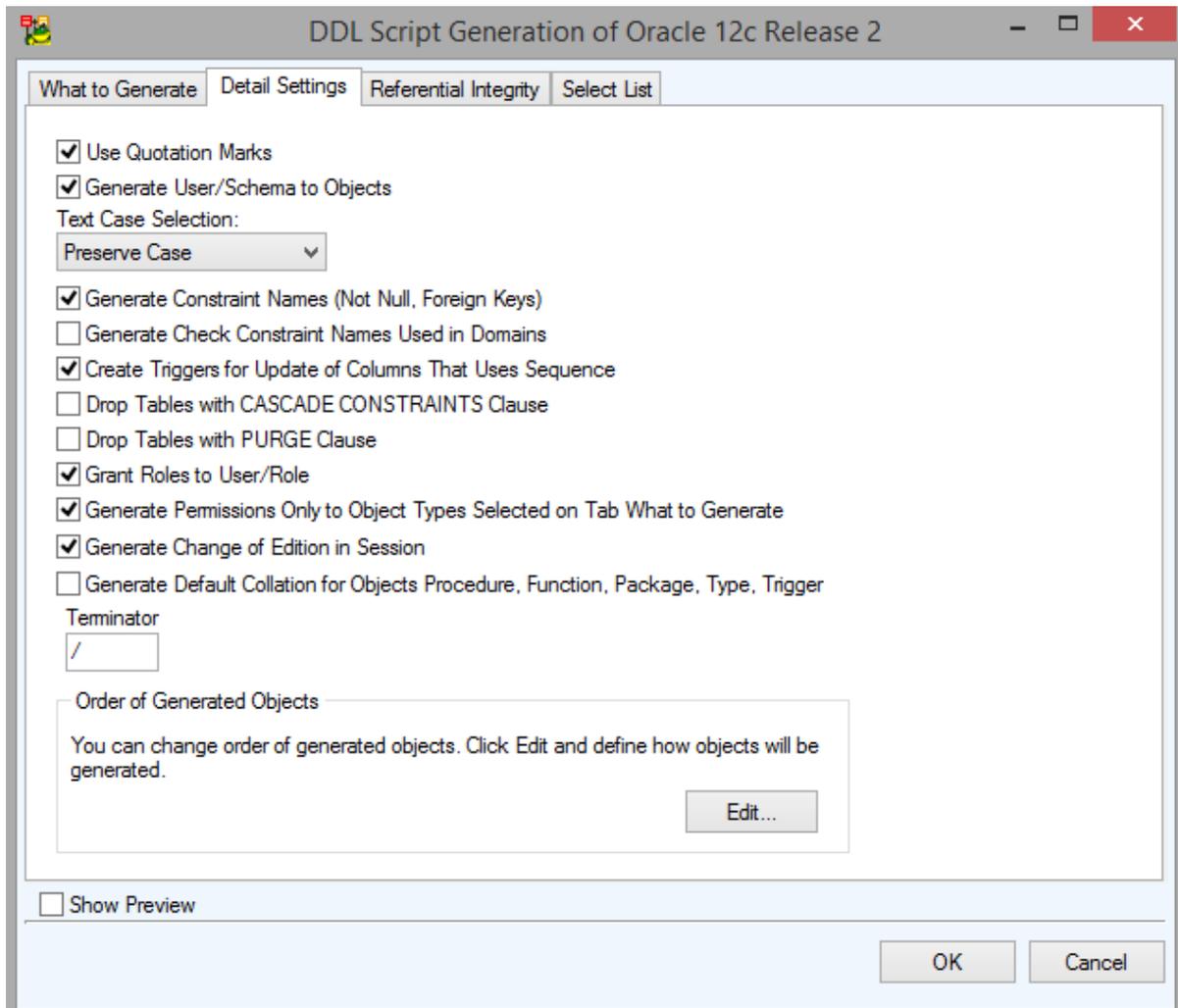
The following properties' definitions need to be entered manually into the text field (as CREATE USER sentence) and are loaded automatically during Reverse Engineering:

- New property HTTP DIGEST ENABLE
- New property LOCAL TEMPORARY TABLESPACE

## **Reverse Engineering - Oracle 12c Release 2**

See [Reverse Engineering - Oracle 21c](#) for more information.

# Script Generation - Oracle 12c Release 2

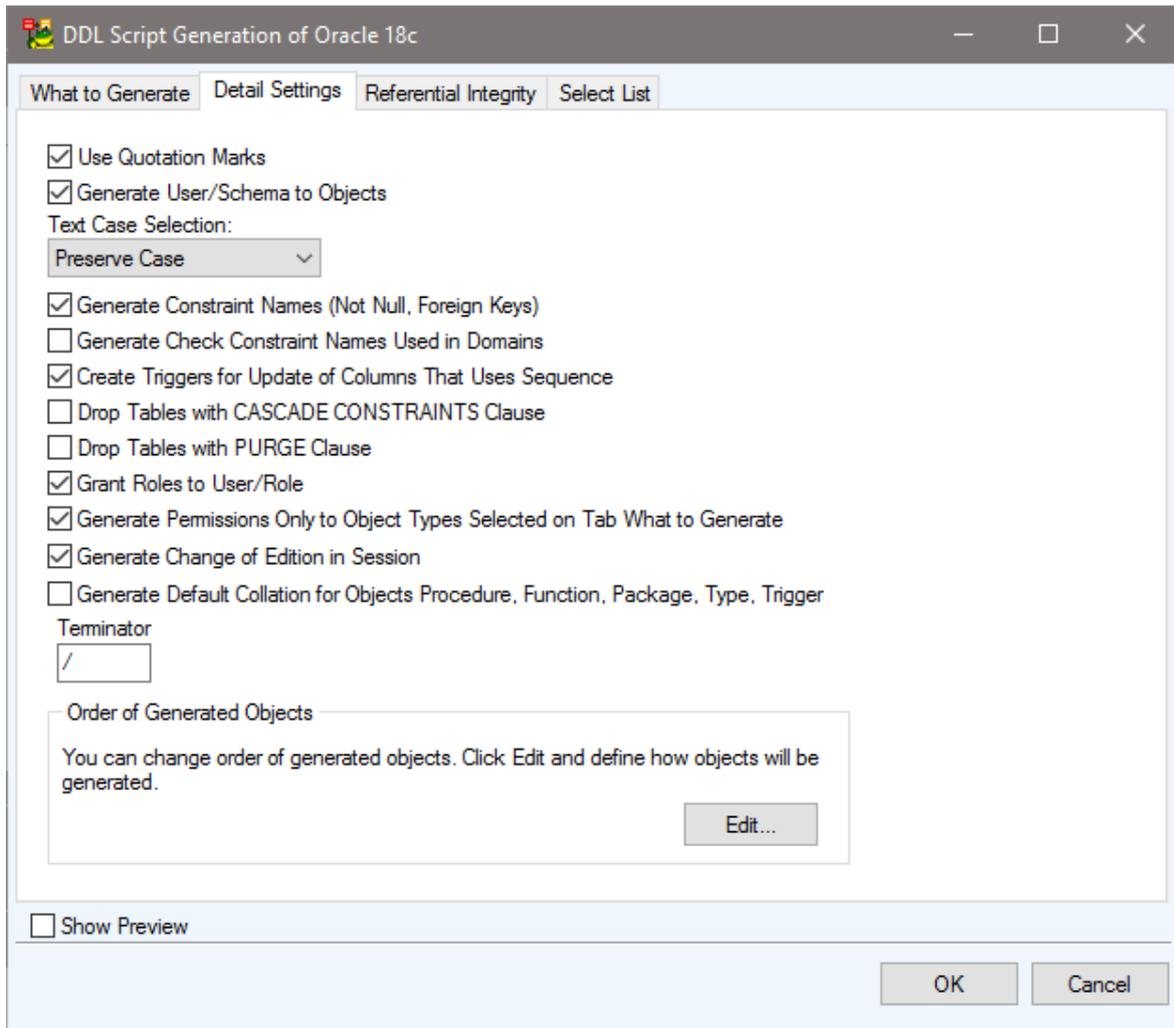


# Specifics - Oracle 18c

## Reverse Engineering - Oracle 18c

See [Reverse Engineering - Oracle 21c](#) for more information.

## Script Generation - Oracle 18c



# Specifics - Oracle 19c

## Reverse Engineering - Oracle 19c

See [Reverse Engineering - Oracle 21c](#) for more information.

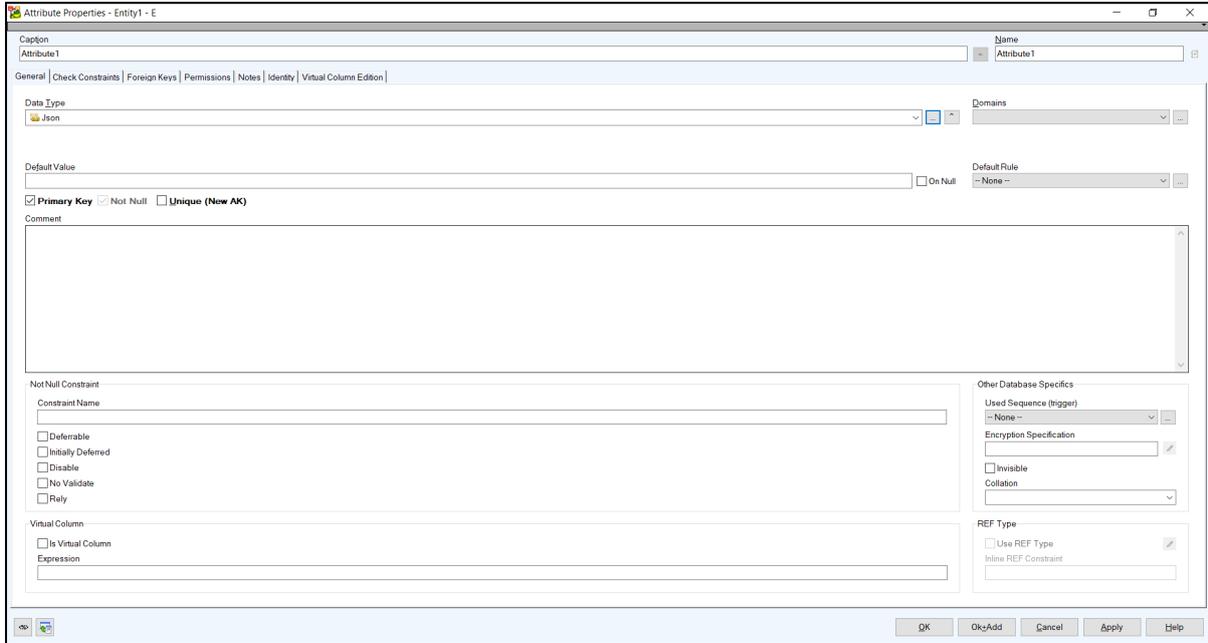
## Script Generation Oracle 19c

See [Script Generation - Oracle 9i](#) for more information.

# Specifics - Oracle 21c

## Attribute Properties

### JSON Data Type



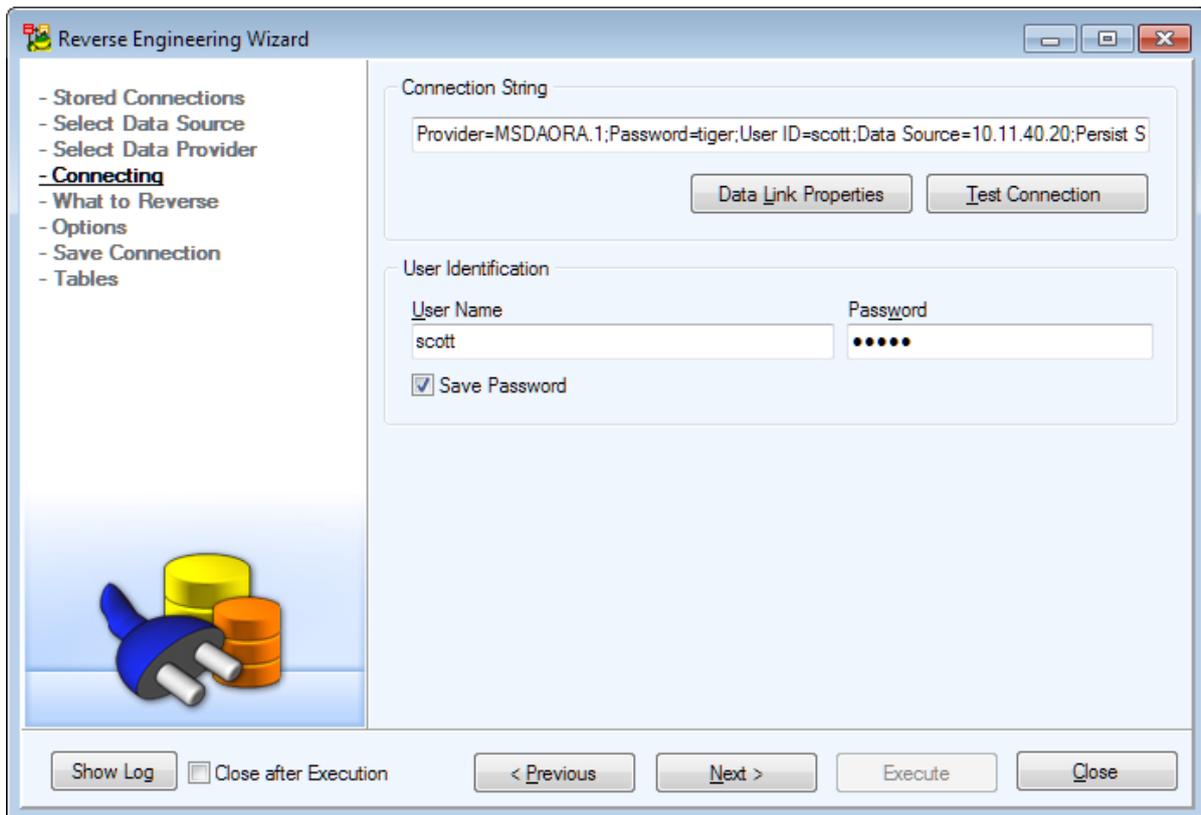
The JSON data type was introduced in the Oracle 20c preview release to provide native JSON support and improve JSON processing performance. It became generally available in Oracle 21c.

# Reverse Engineering - Oracle 21c

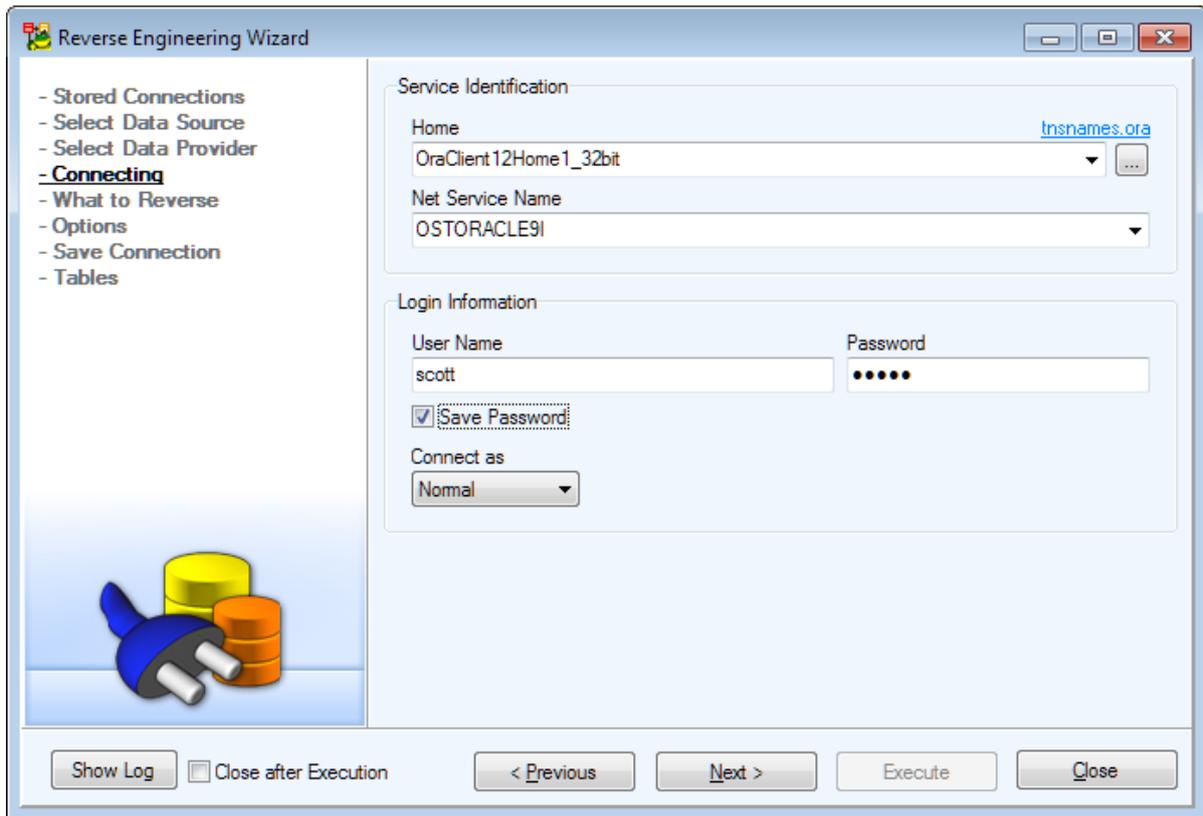
Available **Data Providers** are:

- **Connection via ADO**
- **Native Connection**
- **Connection via TCP/IP**

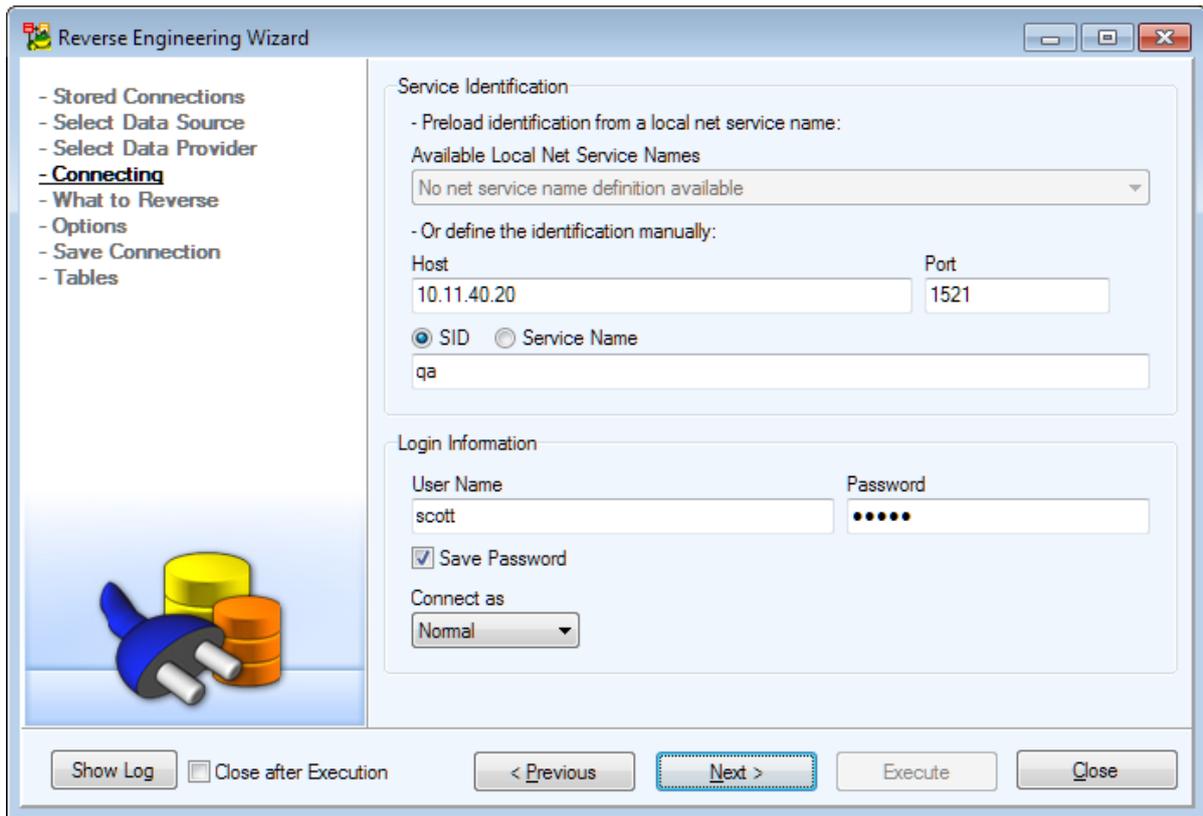
**Connection via ADO:**



**Native Connection:**



**Connection via TCP/IP**

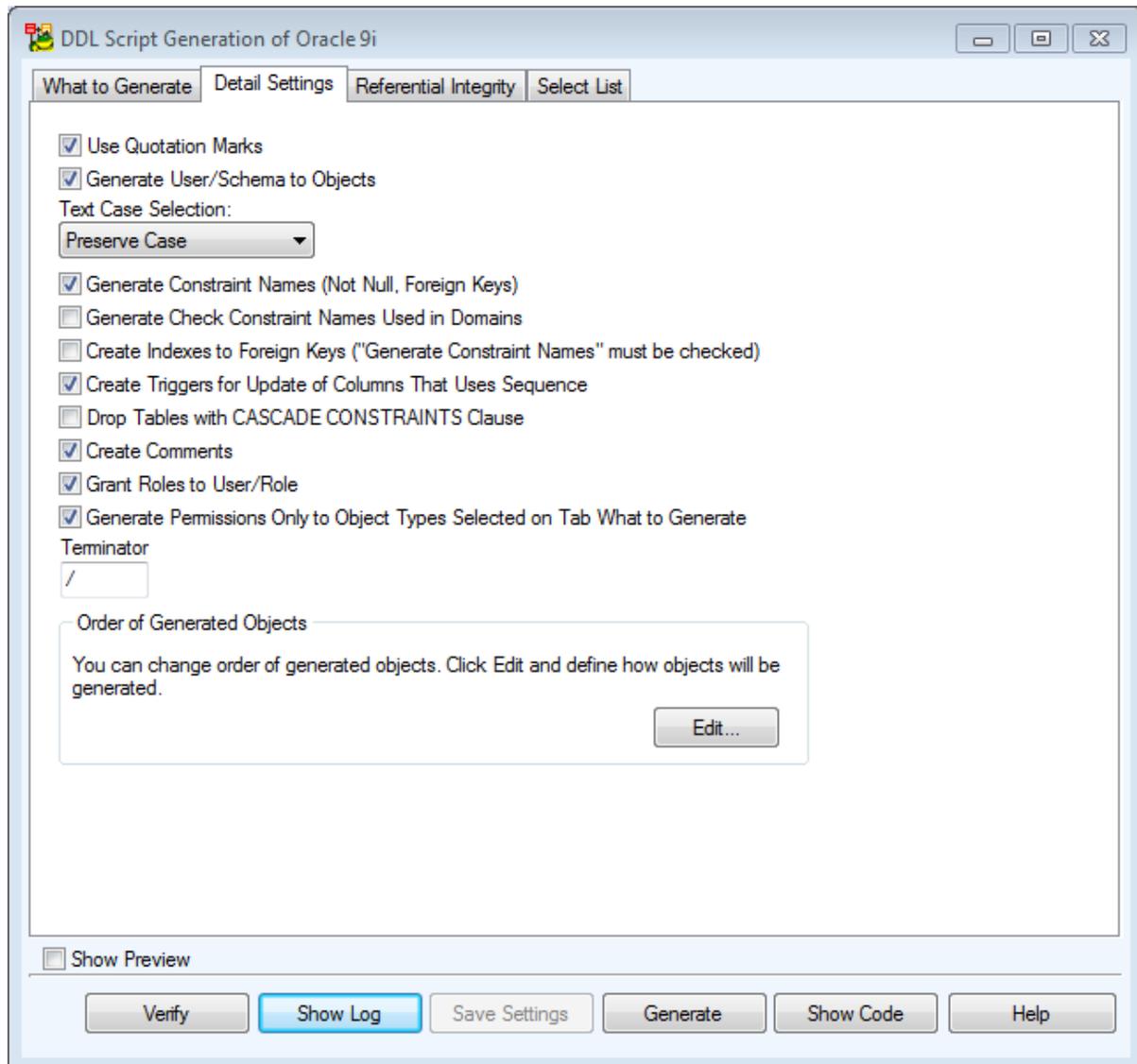


**i** Note:

1. If the following error message occurs in the Log area: "Unable to reverse users, roles and permissions. You haven't assigned the SELECT\_CATALOG\_ROLE role!", it means you have not all the necessary rights to load users, roles and permissions.
2. To load these items successfully, you need to have the SELECT\_CATALOG\_ROLE role assigned or need to set a right for user to SELECT tables DBA\_USERS, DBA\_ROLES, DBA\_ROLE\_PRIVS, DBA\_TAB\_PRIVS.
  - Missing access to system table ALL\_TABLES.
  - Missing privilege SELECT on system table ALL\_TABLES.

**i** Note: Connection via TCP/IP does not support Oracle native encryption.

# Script Generation - Oracle 21c



## Changes since Toad Data Modeler v. 3.5

Automatically generated trigger (trigger for an attribute sequence – see the **Attribute Properties** dialog | **General** tab | **Used Sequence/Trigger** box) is generated as a standard trigger (e.g. entity trigger).

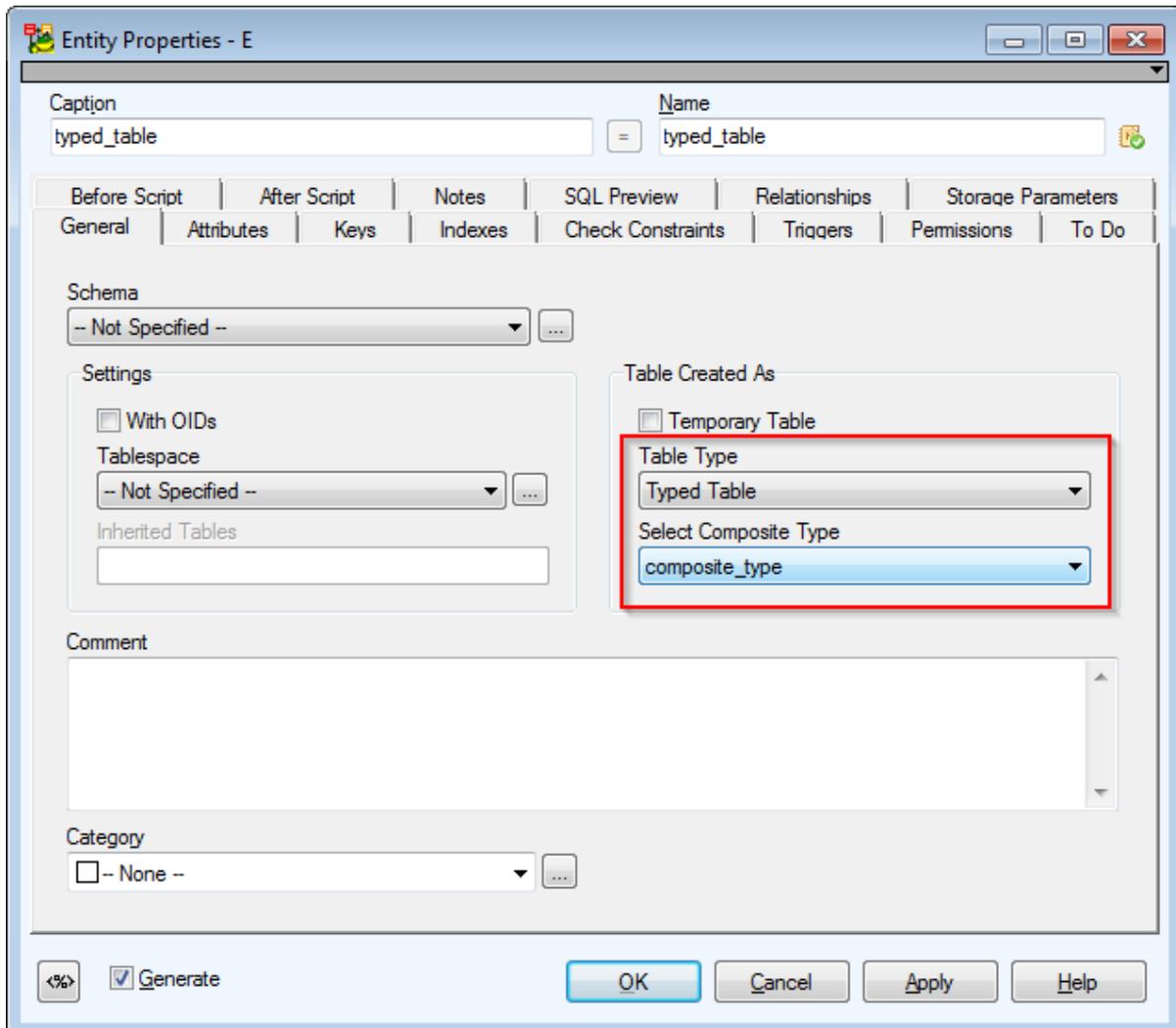
This trigger is generated provided that the *Triggers* item is selected in the **DDL Script Generation** dialog | **What to Generate** tab. It is NOT necessary to select also the *Entity* item.

In the past, this trigger was generated only provided that the *Entities* item was selected (the *Triggers* item was not taken into consideration).

# Specifics - PostgreSQL 9.5

## Entity

### Table Type checkbox

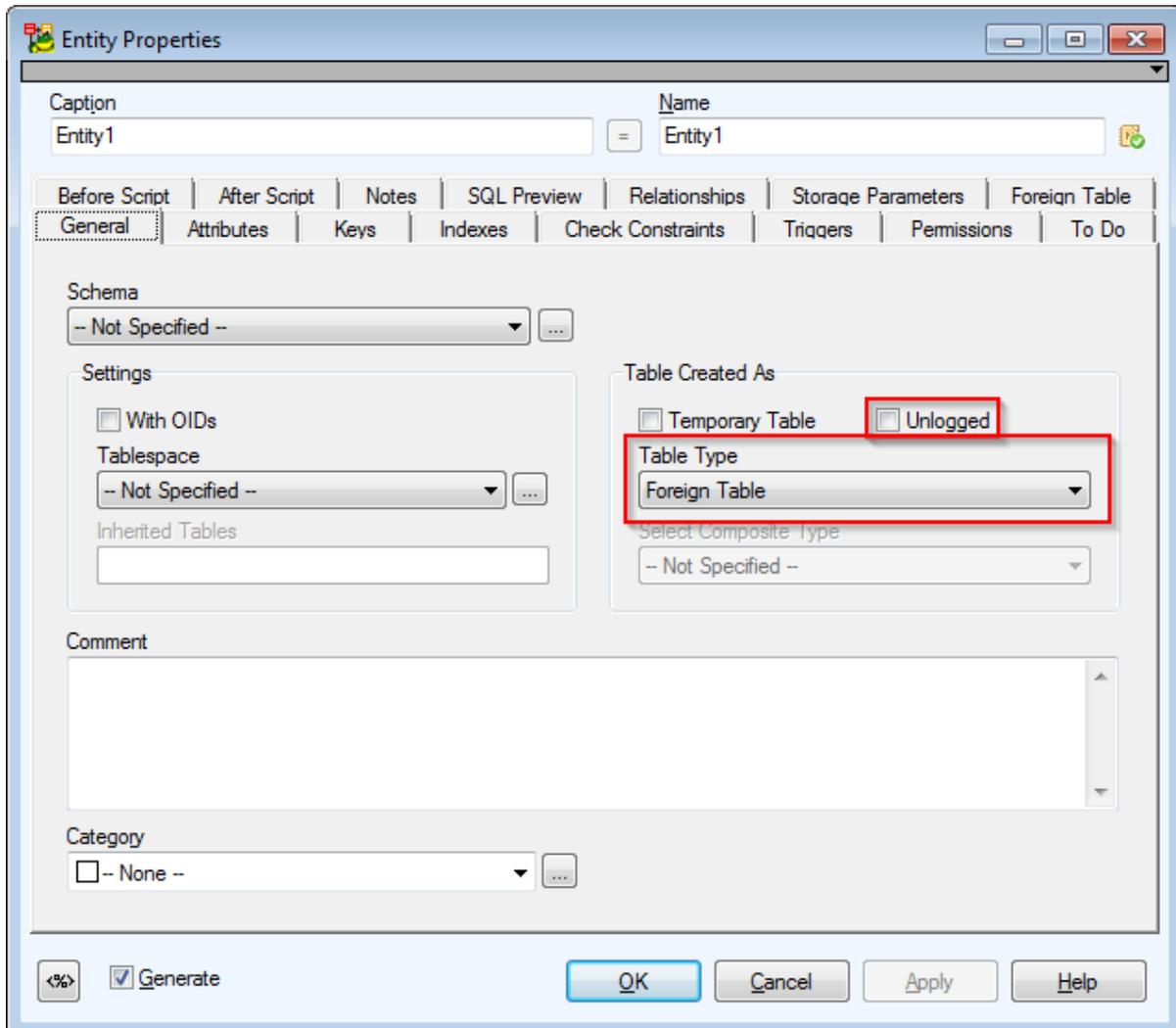


New combo box **Table Type** available in **Entity Properties** dialog, tab **General**. When the **Typed Table** option is selected, the new combo box **Select Composite Type** appears below the **Table Type** combo box. **Select Composite Type** combo box lists all existing **User Data Types**.

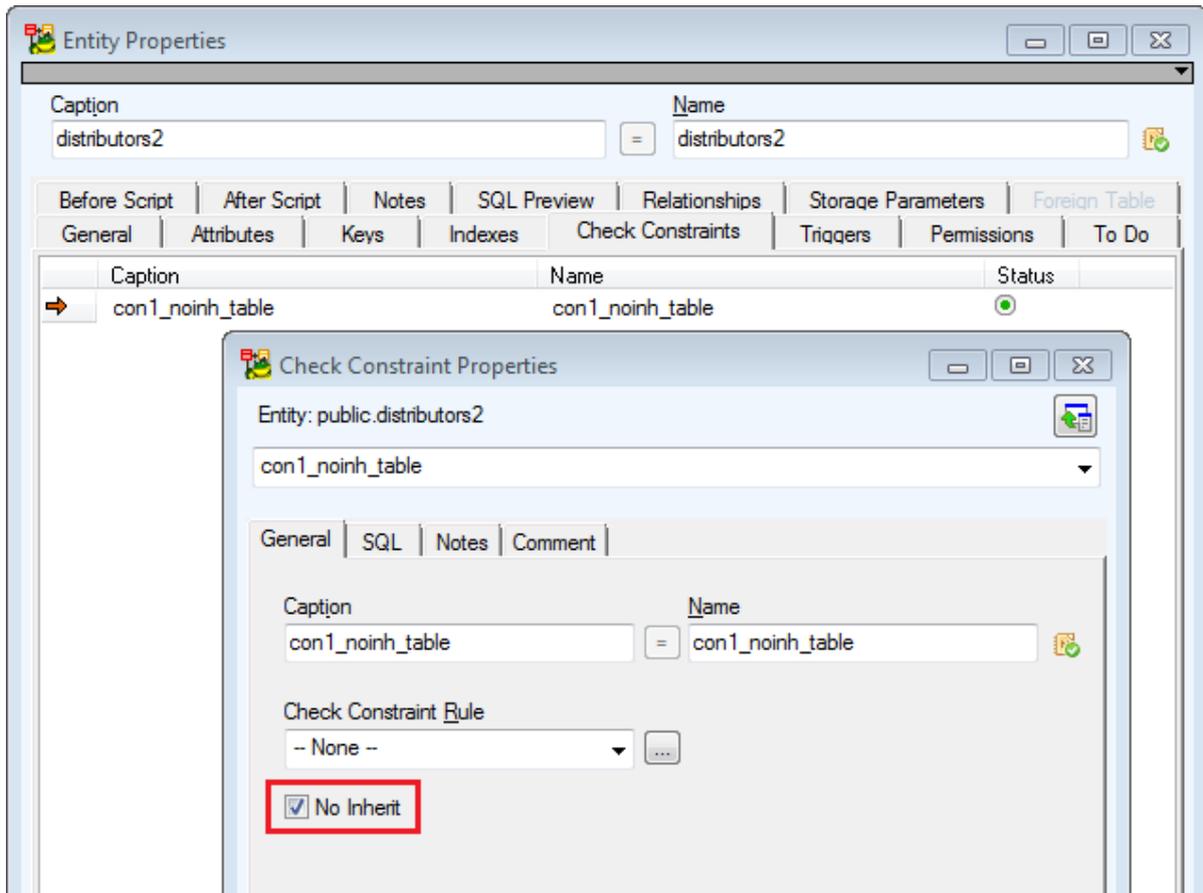
- i** Note: Make sure you select **Composite type table** from the combo box or run verification, which will return an Error message in case a wrong type is used.

# Unlogged checkbox

New checkbox **Unlogged** added (verification changed accordingly).



## Table, Attribute - Check Constraint



**No Inherit** - Allows not inherited constraints for tables and attributes.

Generated SQL:

```
create table distributors2
(
  did integer,
  name varchar (40),
  CONSTRAINT con1_noinh_table CHECK (did > 100 AND name <> ") NO INHERIT
)
```

## User Data Type

**General** tab, **Type** box - New type *Range Type*.

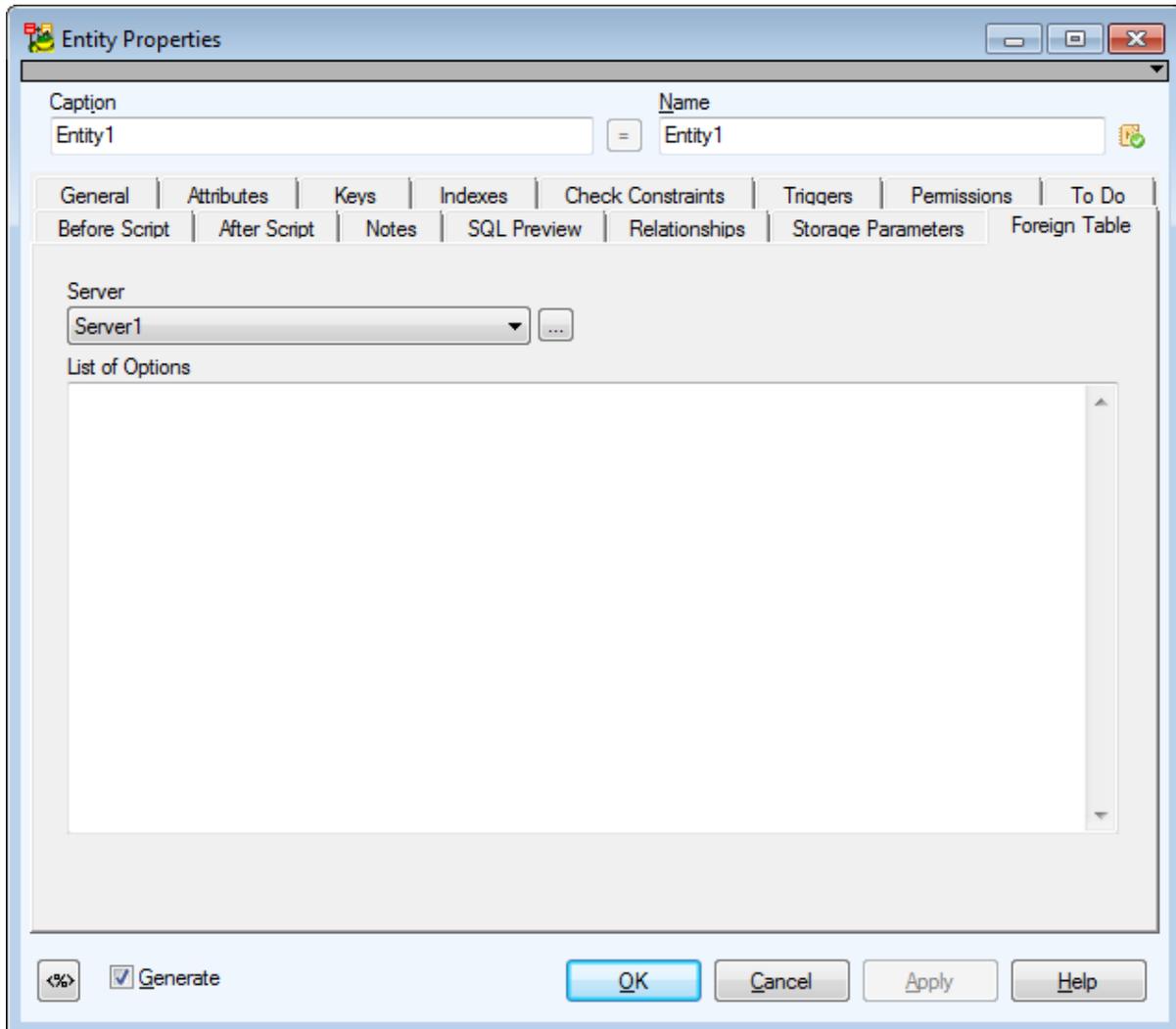
See other objects in Model Explorer:

- Aggregates
- Collations
- Foreign Servers

- Tablespaces
- Sequences
- Rewrite Rules

## Foreign Table

When *Foreign Table* is selected in the **Table Type** combobox, a new **Foreign Table** tab displays.



**List of Options** box should contain the part of CREATE FOREIGN TABLE command written in brackets in OPTIONS—e.g. filename 'c:\file.txt'.

Although common entity object is used for *Foreign Table* entity in Toad Data Modeler, only the following features can be used:

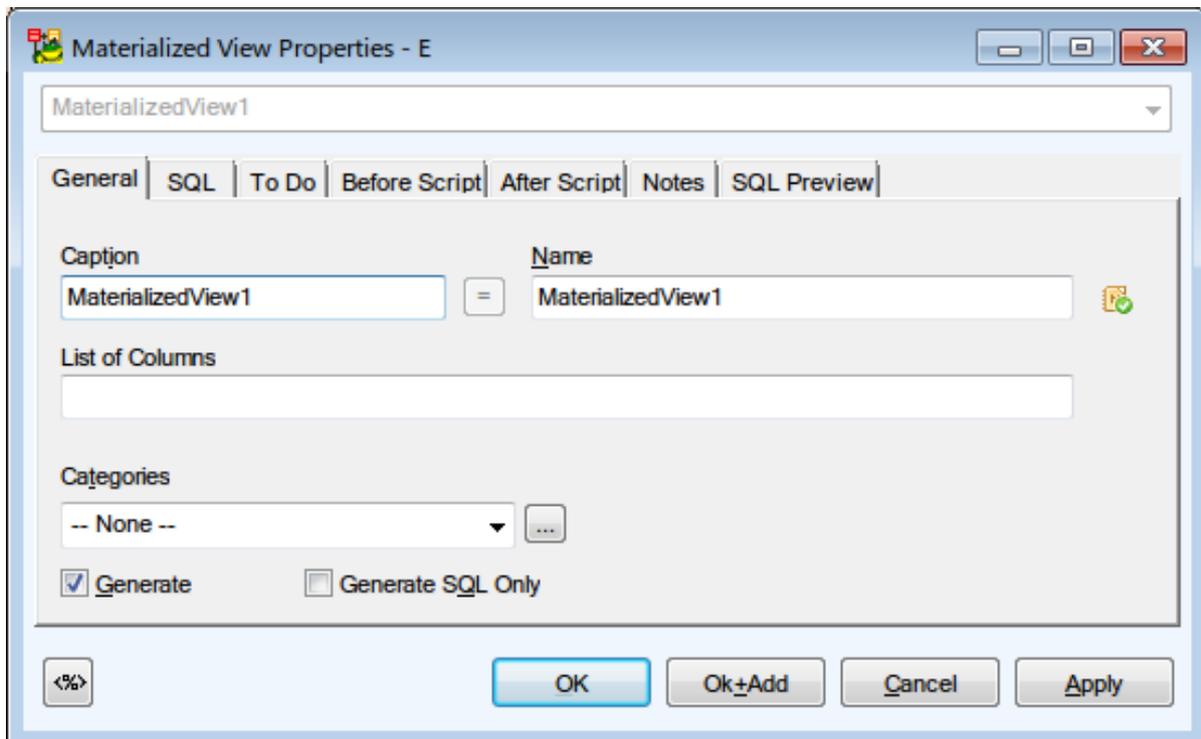
- Name
- Schema
- Attributes list

- Comment
- Permission
- Foreign Server combobox
- Foreign Table Options tab

For Attributes, only the following features can be used:

- Name
- Data Type
- Null

## Materialized Views



## Views

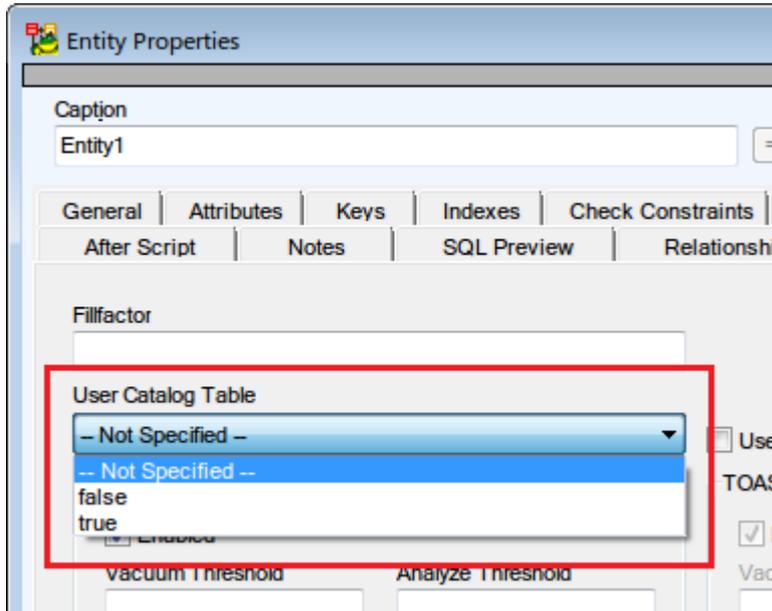
*RECURSIVE* parameter

- *CREATE RECURSIVE VIEW name (columns) AS SELECT ...;*

*WITH, CASCADED, LOCAL, CHECK OPTION* parameters

# Tables

*USER\_CATALOG\_TABLE* parameter for **Table Storage Parameters**

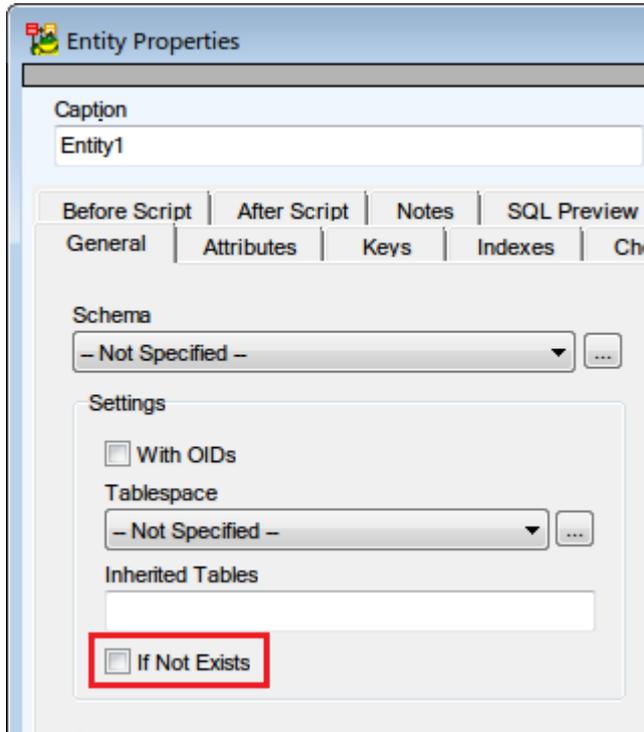


# Aggregates

*SSPACE, FINALFUNC\_EXTRA, INITCOND, MSFUNC, MINVFUNC, MSTYPE, MSSAPCE, MFINALFUNC, MFINALFUNC\_EXTRA, MINITCOND, ORDER BY, HYPOTHETICAL* parameters

# Schemas

*IF NOT EXISTS* parameter



COLLATE, CONSTRAINT parameters

MULTIXACT FREEZE MIN/MAX/TABLE AGE for Autovacuum and TOAST Autovacuum Table Storage Parameters

Entity Properties - E

Entity1 = Entity1

General | Attributes | Keys | Indexes | Check Constraints | Triggers | Permissions | To Do  
Before Script | After Script | Notes | SQL Preview | Relationships | Storage Parameters | Foreign Table

Fillfactor

Use Storage Parameters for TOAST Table

**Autovacuum**

Enabled

Vacuum Threshold:  Analyze Threshold:

Vacuum Scale Factor:  Analyze Scale Factor:

Vacuum Cost Delay:  Vacuum Cost Limit:

Freeze Min Age:  Freeze Max Age:

Freeze Table Age:

Multixact Freeze Min Age:  Multixact Freeze Max Age:

Multixact Freeze Table Age:

**TOAST Autovacuum**

Enabled

Vacuum Threshold:

Vacuum Scale Factor:

Vacuum Cost Delay:  Vacuum Cost Limit:

Freeze Min Age:  Freeze Max Age:

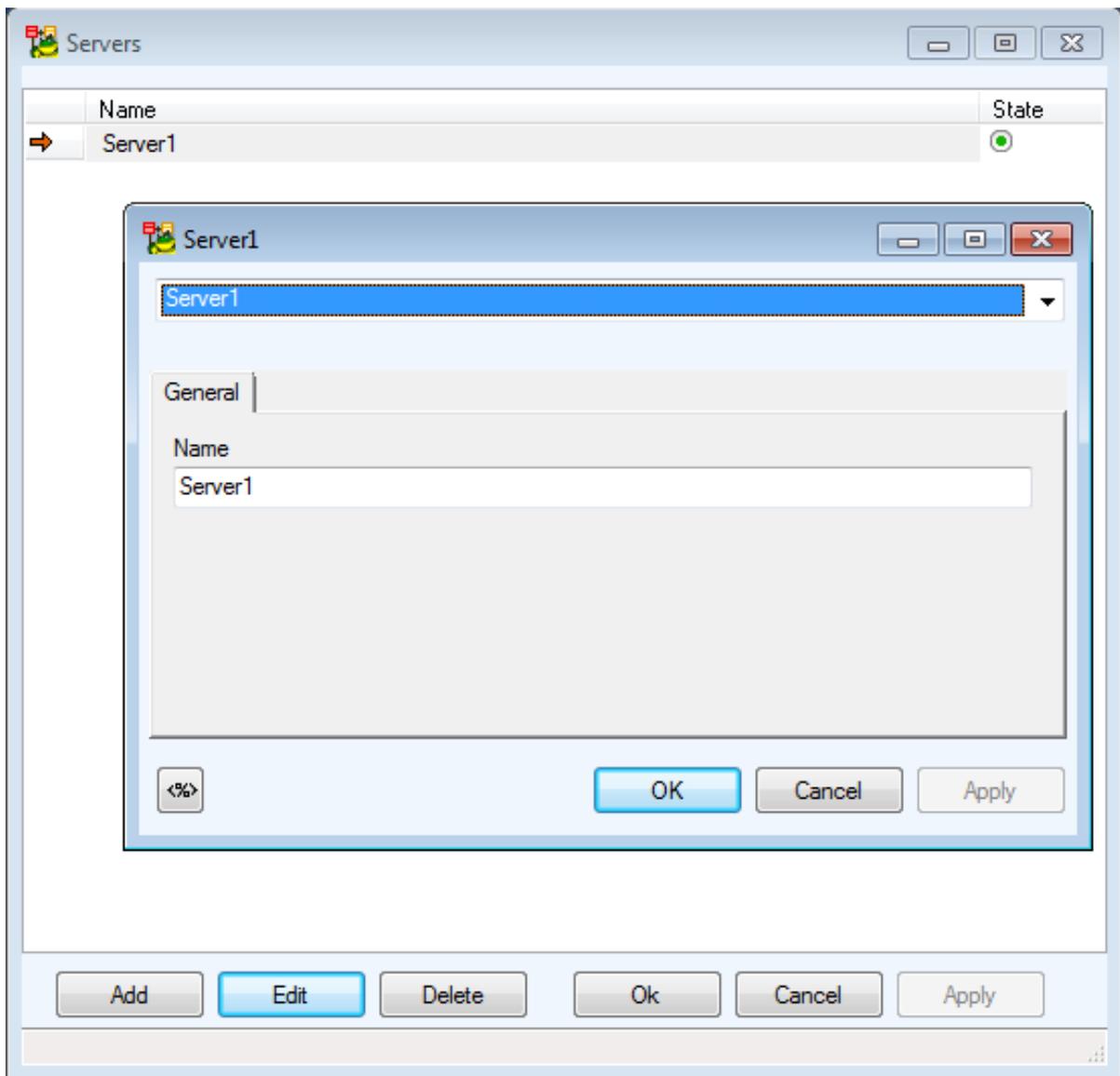
Freeze Table Age:

Multixact Freeze Min Age:  Multixact Freeze Max Age:

Multixact Freeze Table Age:

<%>  Generate OK Cancel Apply Help

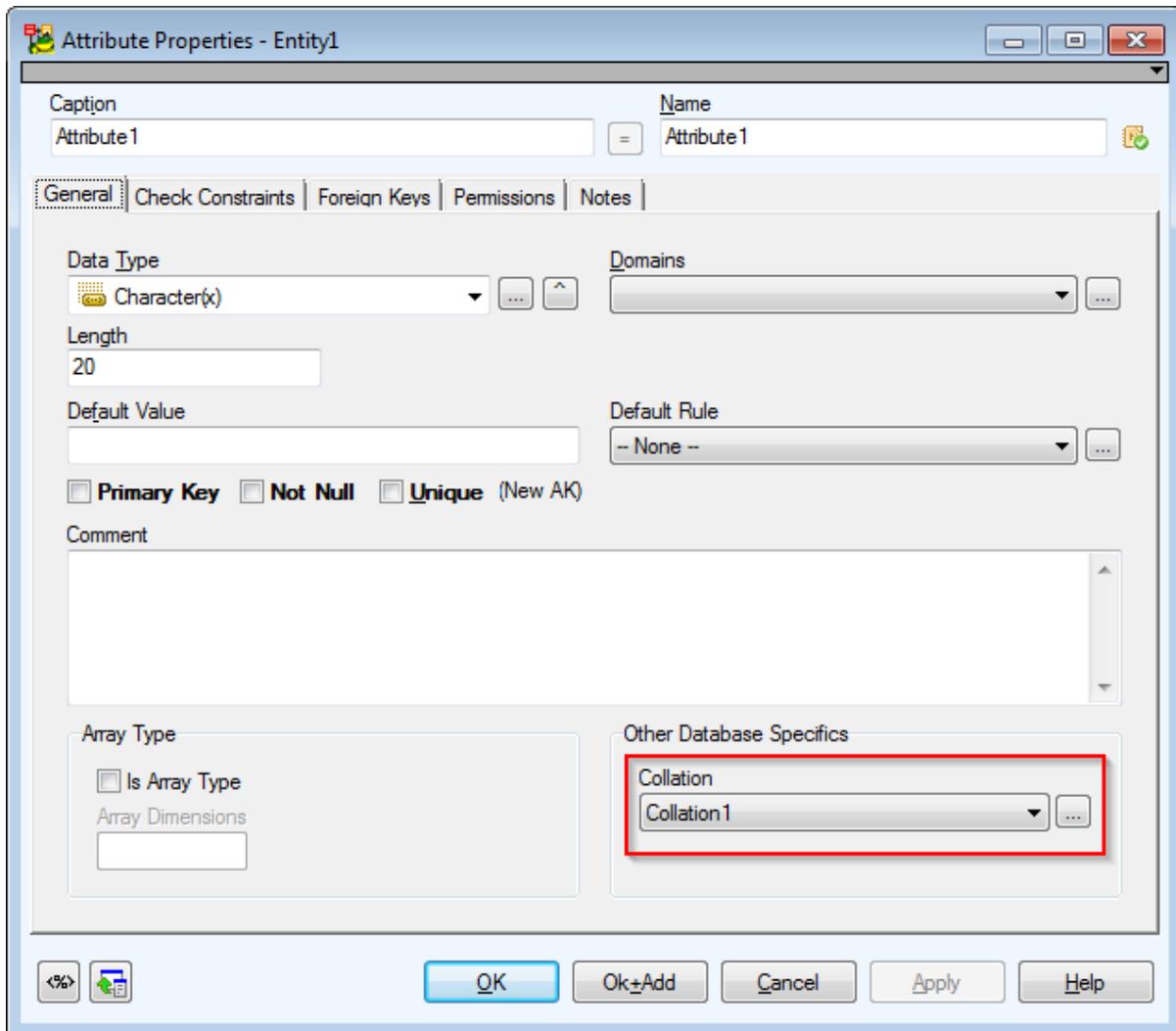
## Foreign Server



- CREATE/DROP/ALTER are not supported
- only a listing function

# Attribute

## Collation combobox



The Collation combobox is available for:

- **Data Type**— *Text, Character varying(x), Character(x), Varchar*
- **User Data Type**— **Base type** with selected **Collatable** checkbox
- **Dictionary Type**—if correct Data Type or User Data Type is selected (see above)
- **Domain**—combobox is disabled (changes not allowed)

## Attributes - Foreign Table

Attribute Properties - postgres.T\_CUSTOMER

Caption  
name

General | Check Constraints | Foreign Keys | Permissions | No

Data Type  
Character varying(x)

Length  
24

Default Value

Primary Key  Not Null  Unique (New AK)

List of Options (Foreign Table)  
columndelimiter ':'

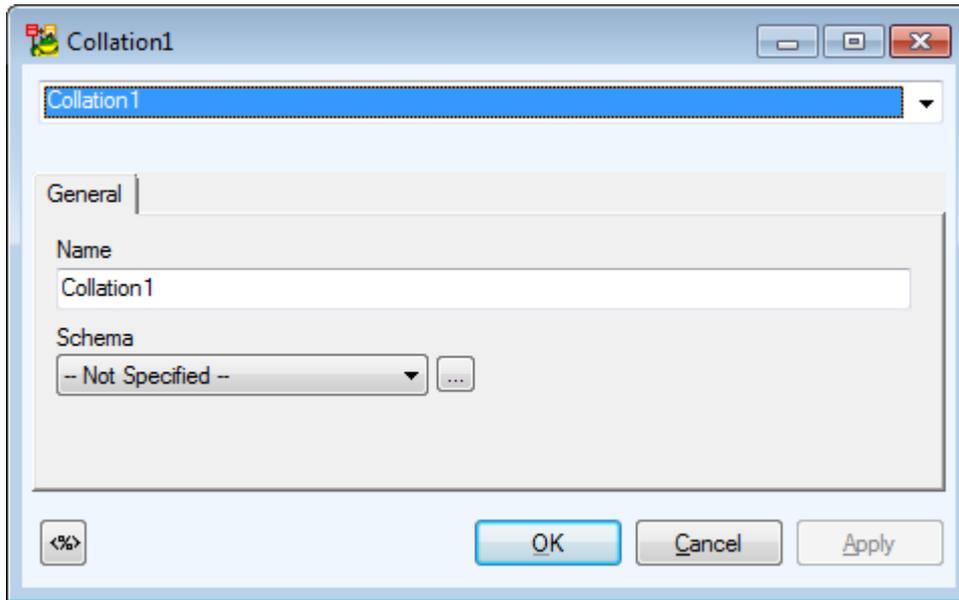
Comment  
Name column can contain First and Middle name. Surname r

Resulting in:

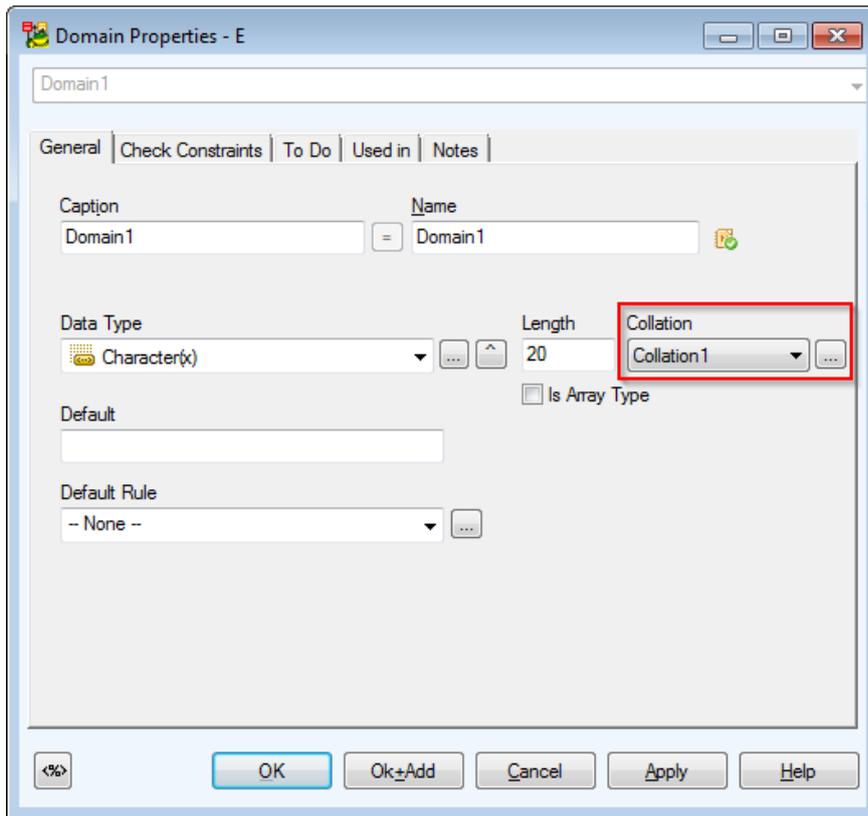
```
CREATE FOREIGN TABLE "postgres"."T_CUSTOMER"(  
  "customer_id" Integer NOT NULL,  
  "name" Character varying(24) OPTIONS (columndelimiter ':'),  
  "address" Text  
)
```

**Data Type** box - *Smallserial* and *Json* new data types.

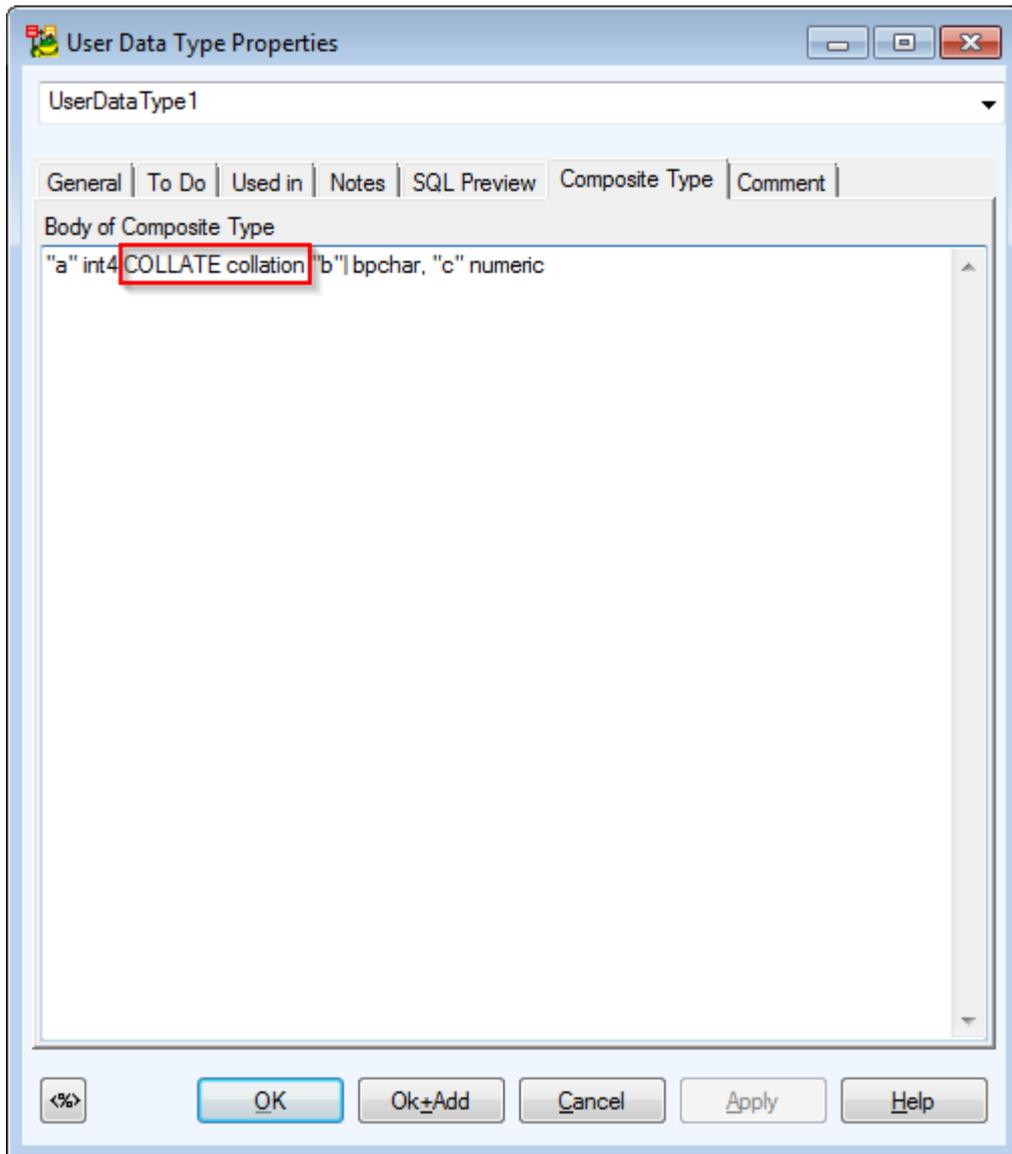
# Collation



- CREATE/DROP/ALTER are not supported
- only a listing function

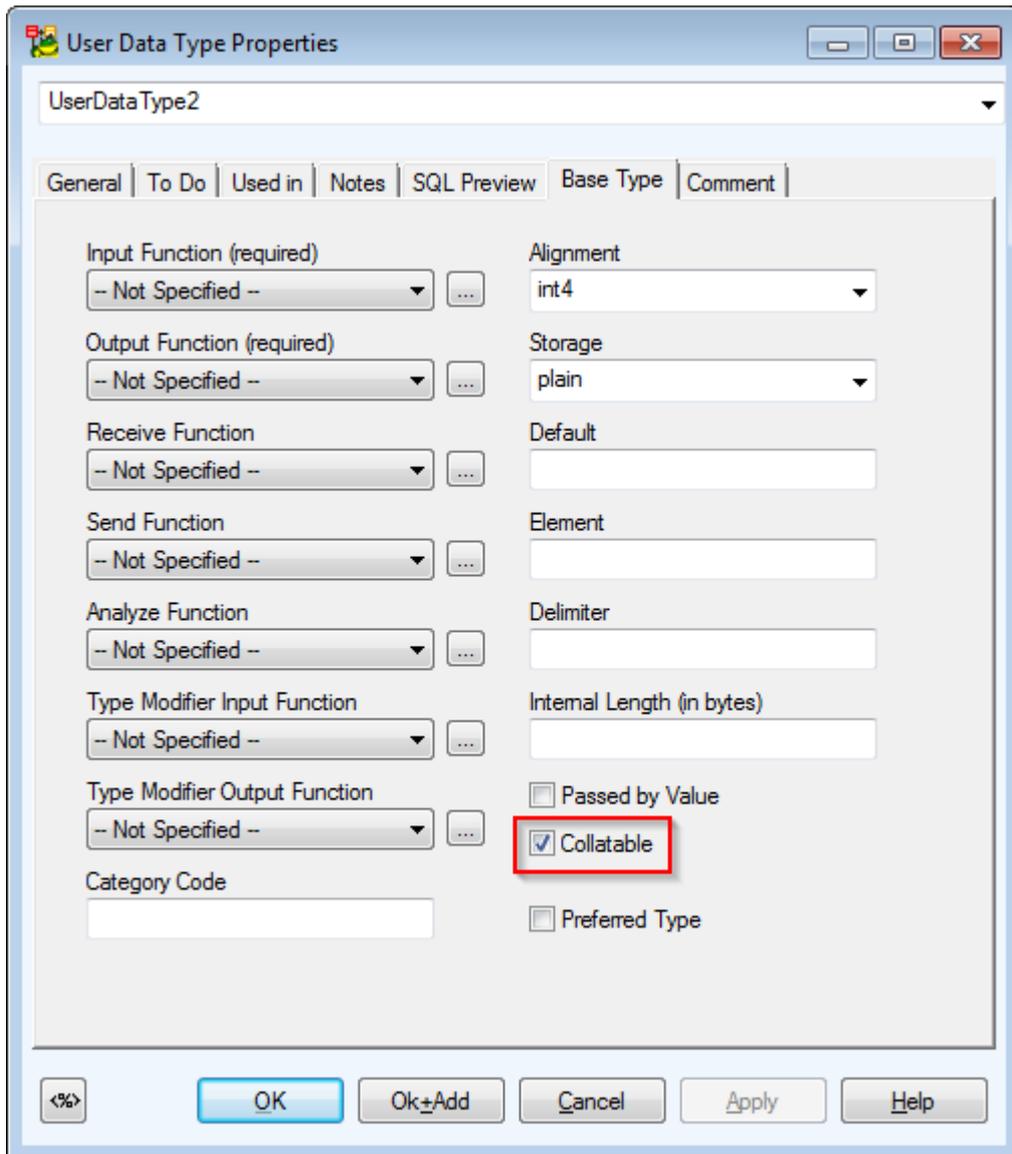


## User Data Type—Composite Type



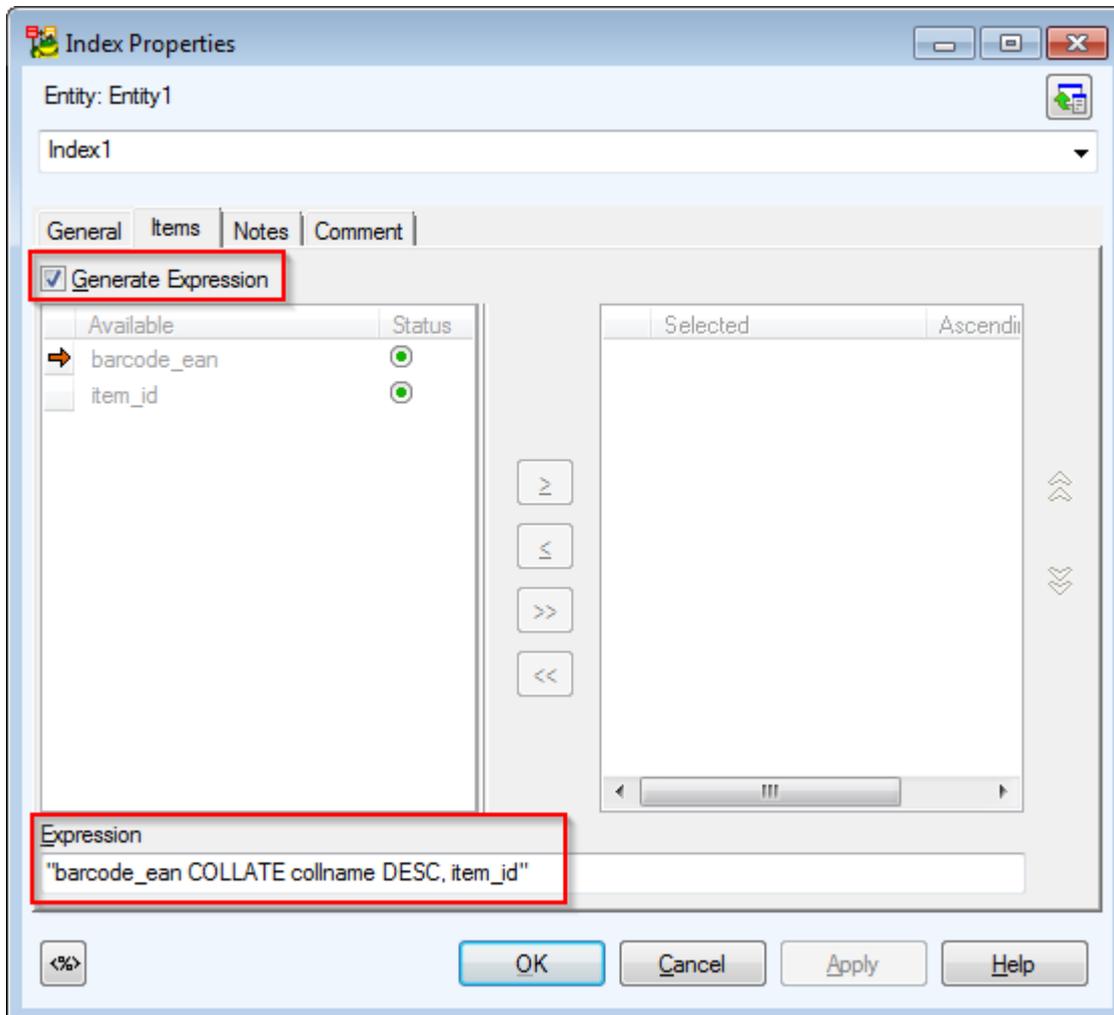
COLLATE collation definition can be used for every attribute to set its Collation. To set Collation, manually write COLLATE collation expression in **Composite Type** tab right after "attributename datatype" definition (see the screenshot above). Toad Data Modeler will load it during RE.

No attribute is necessary for Composite Type.



# Index

## Set Collation



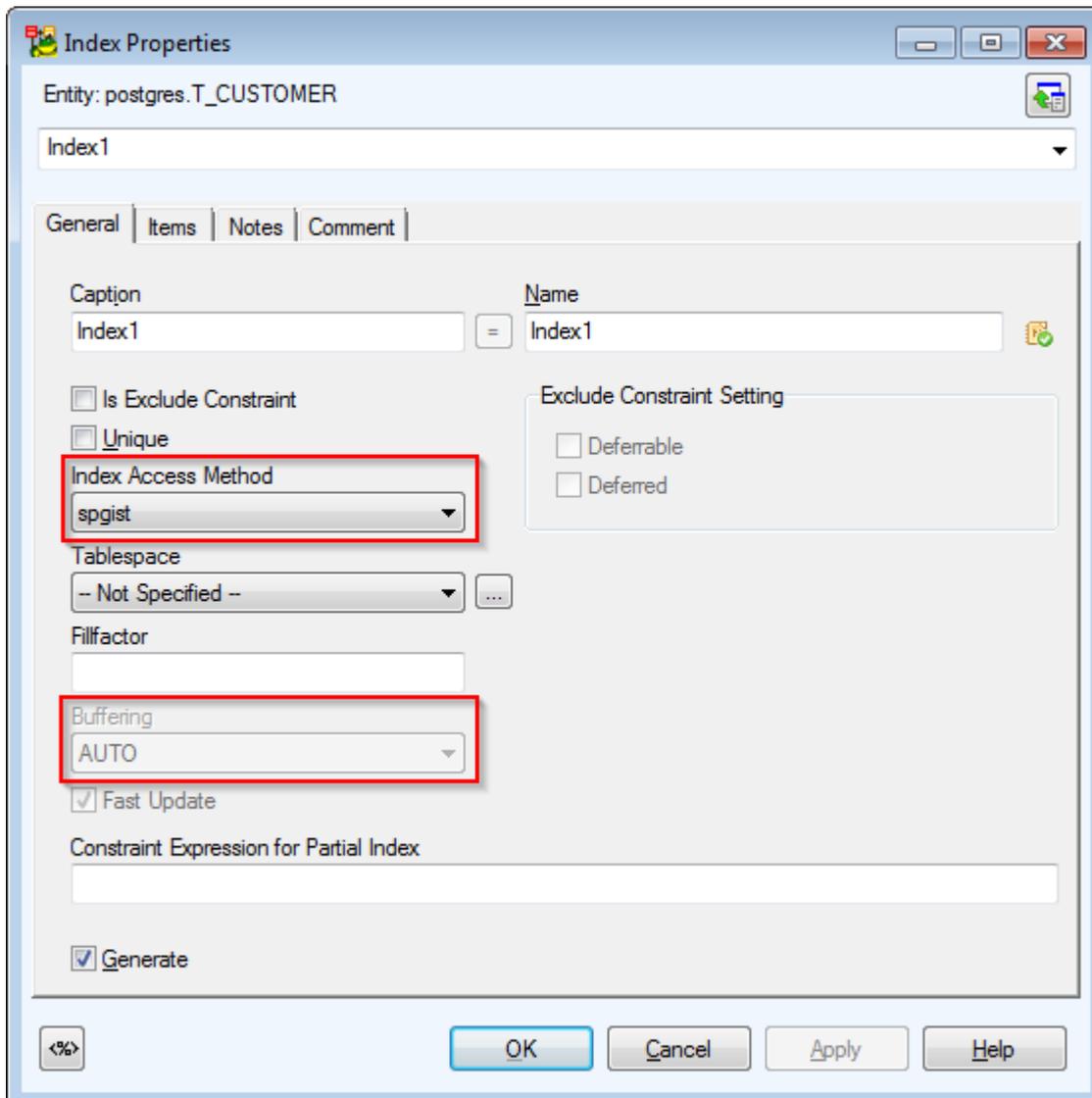
It is possible to set Collation of every column/attribute of the table.

### To set Collation

1. Select the **Generate Expression** checkbox.
2. In the **Expression** box, write e.g "atr1 COLLATE collname DESC, atr2".

Toad Data Modeler loads this index as an expression during RE.

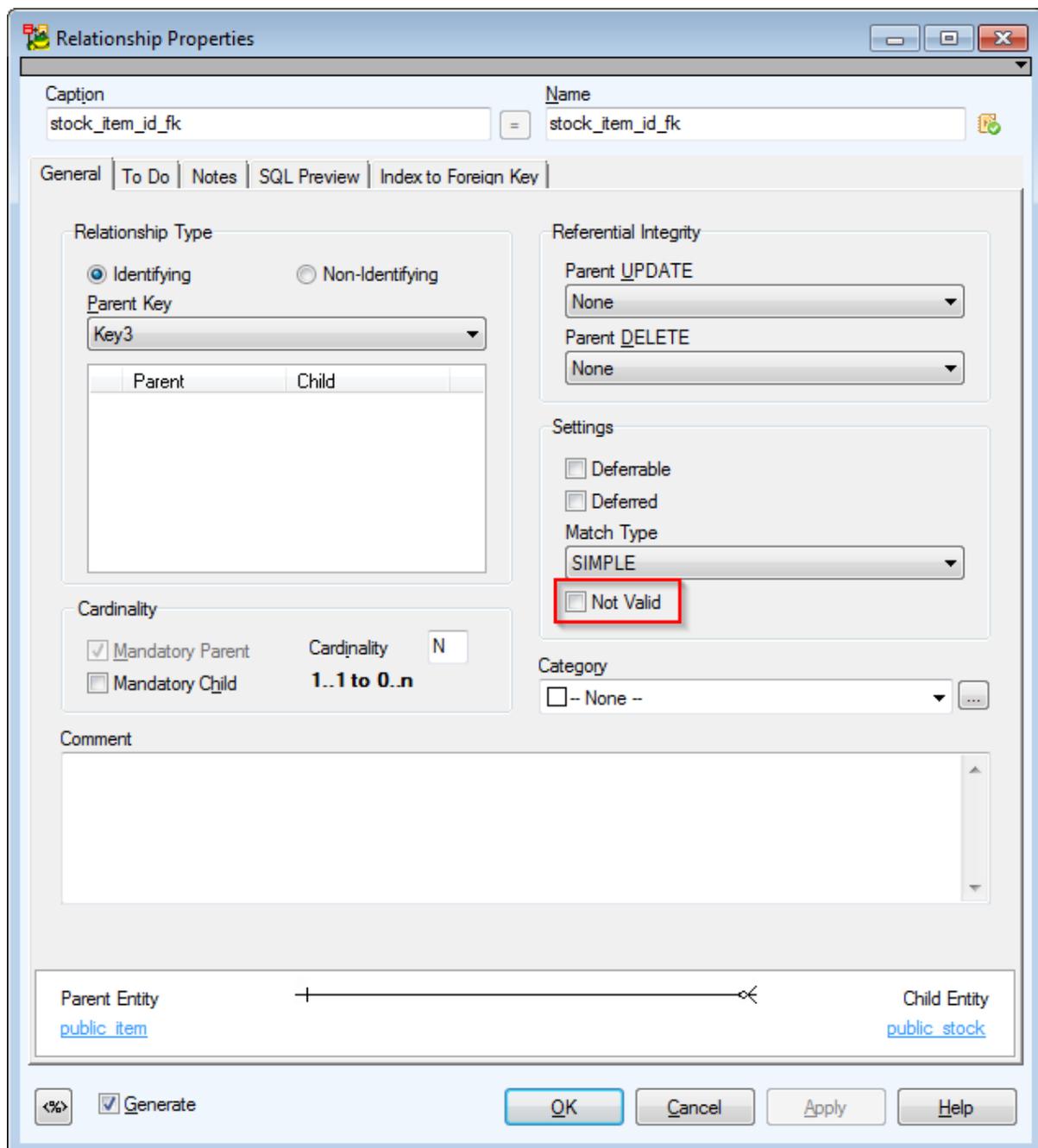
## Index Access Method



**Index Access Method**- new option *spgist*.

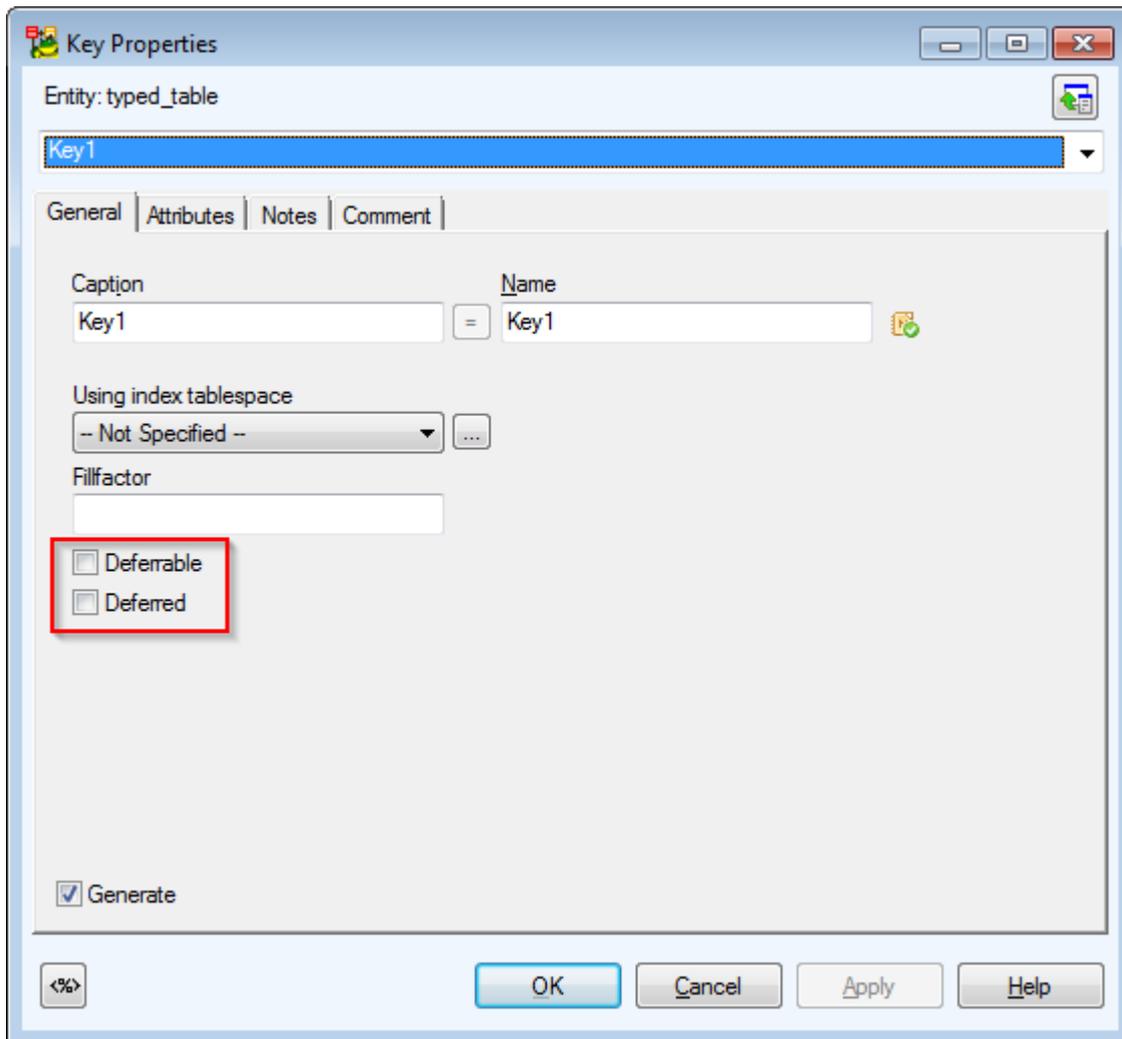
**Buffering** parameter option for *gist* index.

# Relationship



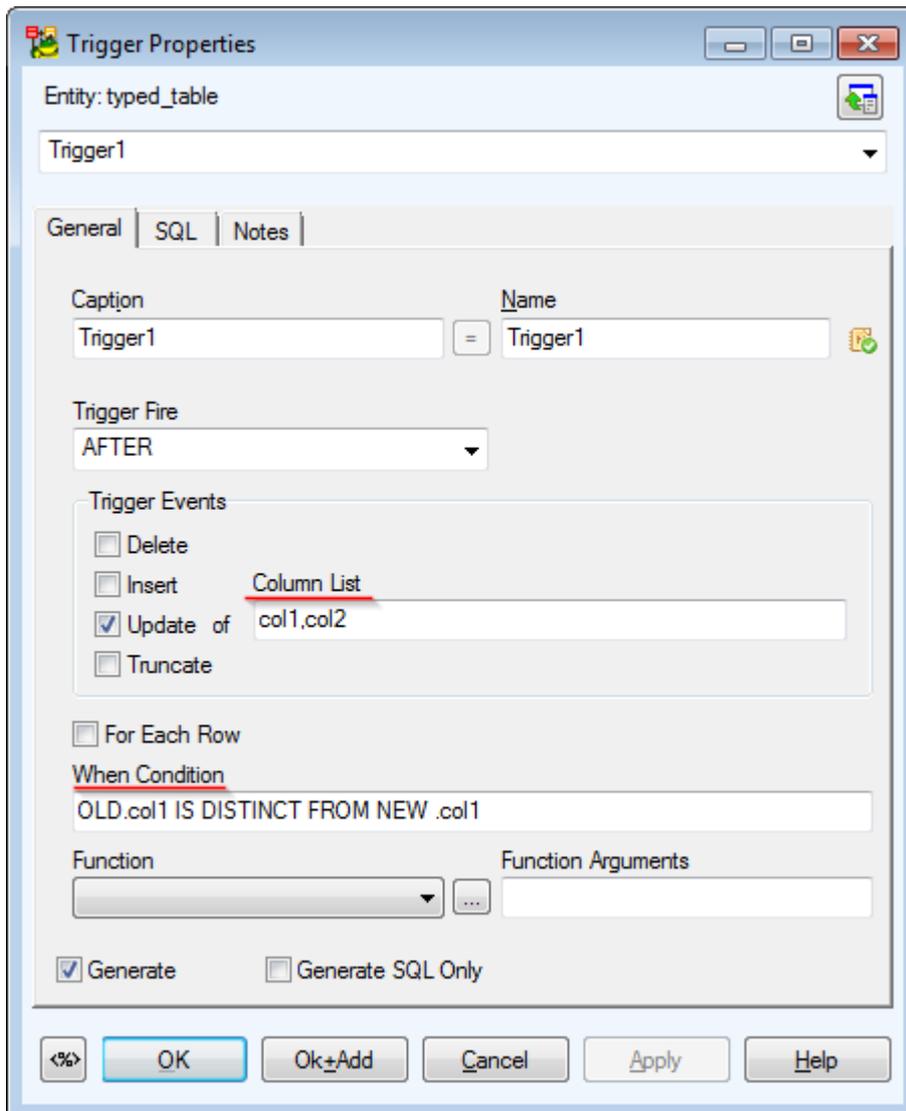
**Not Valid** checkbox—option when you add a constraint of a foreign key.

# Key Properties



Options **Deferrable** and **Deferred** available on **Key Properties** dialog, tab **General**.

# Trigger

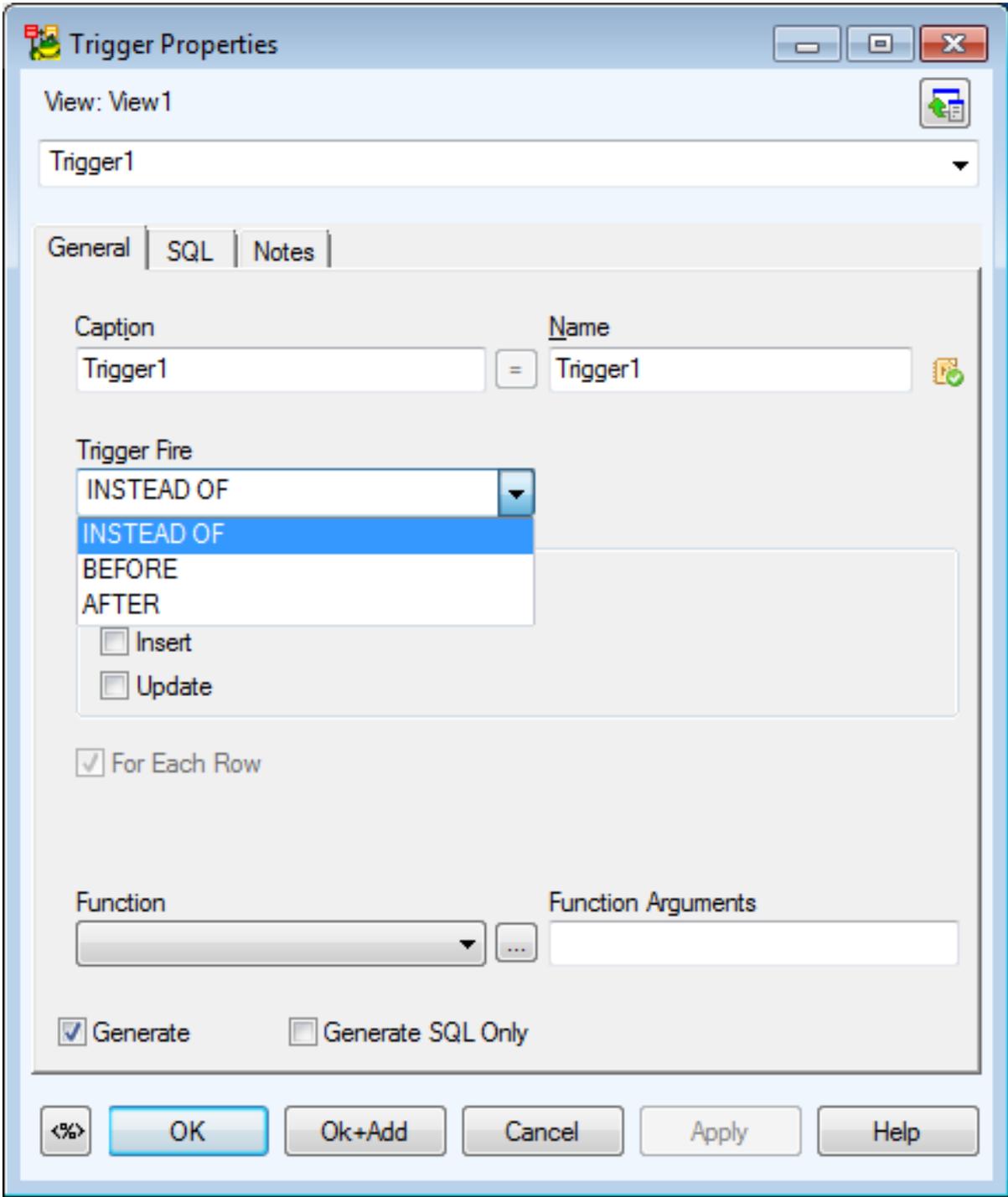


New **Column List** box available for trigger event **Update** (Update checkbox must be selected).

**i** | Note: Use comma (,) as a separator.

New **When Condition** box available on **Trigger Properties** dialog, tab **General**.

# View - Trigger



- *INSTEAD OF UPDATE* does not support columns list
- *AFTER/BEFORE UPDATE* support columns list
- if *INSTEAD OF* is used, the *FOR EACH ROW* checkbox must be applied
- if *AFTER/BEFORE UPDATE* is used, the *FOR EACH STATEMENT* checkbox must be applied
- *INSTEAD OF* does not support *WHEN*

**Other objects in Model Explorer:**

- Tablespaces
- Aggregates
- Sequences
- Rewrite Rules
- Collations
- Foreign Tables
- Foreign Servers

## User Data Type

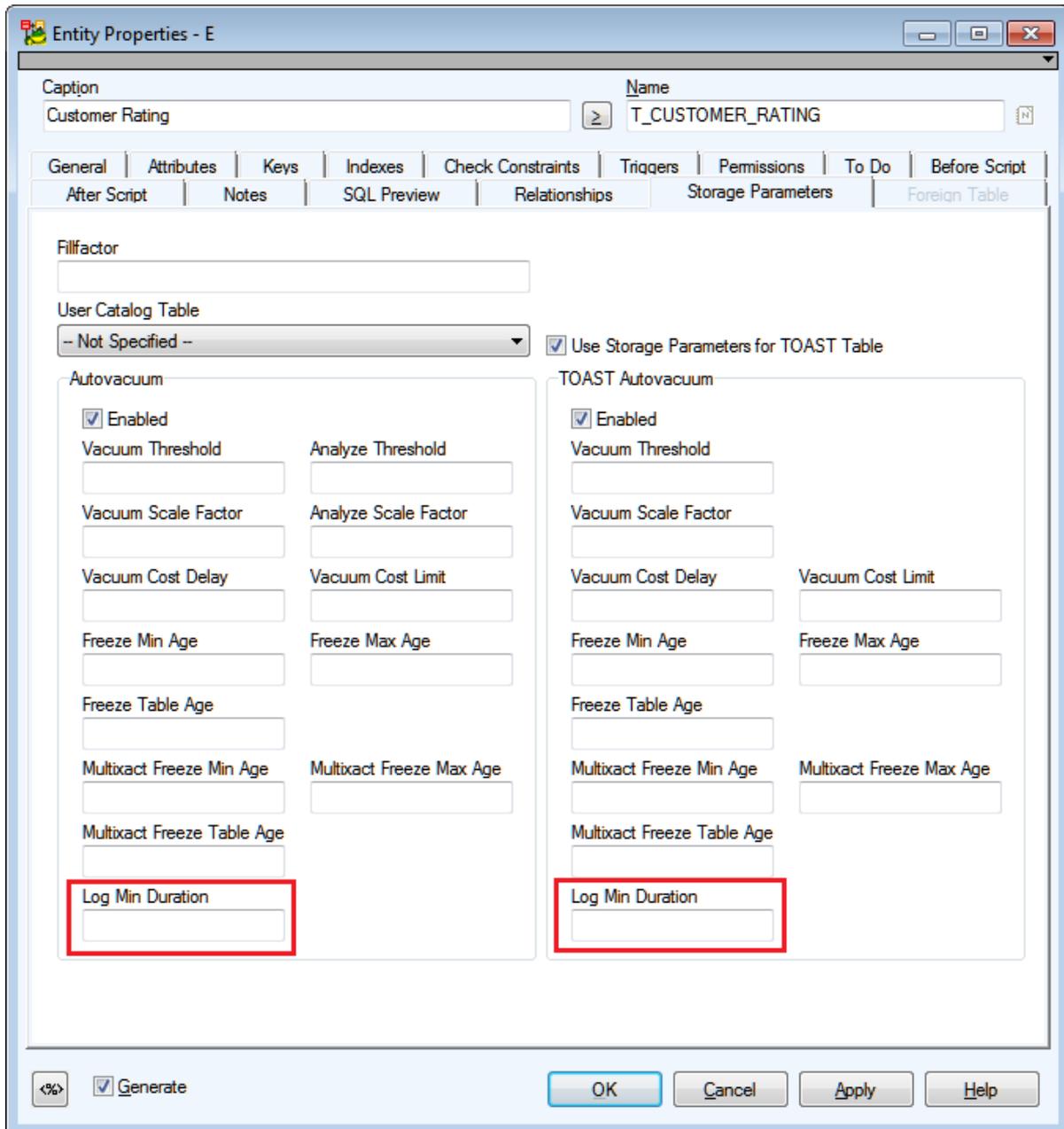
Enumerated type does not require any Label.

Other objects in Model Explorer:

- Aggregates
- Rewrite Rules
- Sequences
- Tablespaces

# Tables

## Storage Parameters - Log Min Duration for Autovacuum and TOAST Autovacuum



**Check constraint generation and inheritance** is now allowed for **foreign tables**.

```
CREATE TABLE users (id SERIAL PRIMARY KEY, username TEXT NOT NULL);  
CREATE FOREIGN TABLE users_shard_5 (CONSTRAINT us5 CHECK (id <> 0 AND username <> ''))  
INHERITS (users) server myserver options ( table_name 'users' );
```

```
CREATE FOREIGN TABLE users_shard_5a (CHECK (id <> 0 AND username <> ''))
INHERITS (users) server myserver options ( table_name 'users' );
```

## Indexes

### Index Access Method - brin

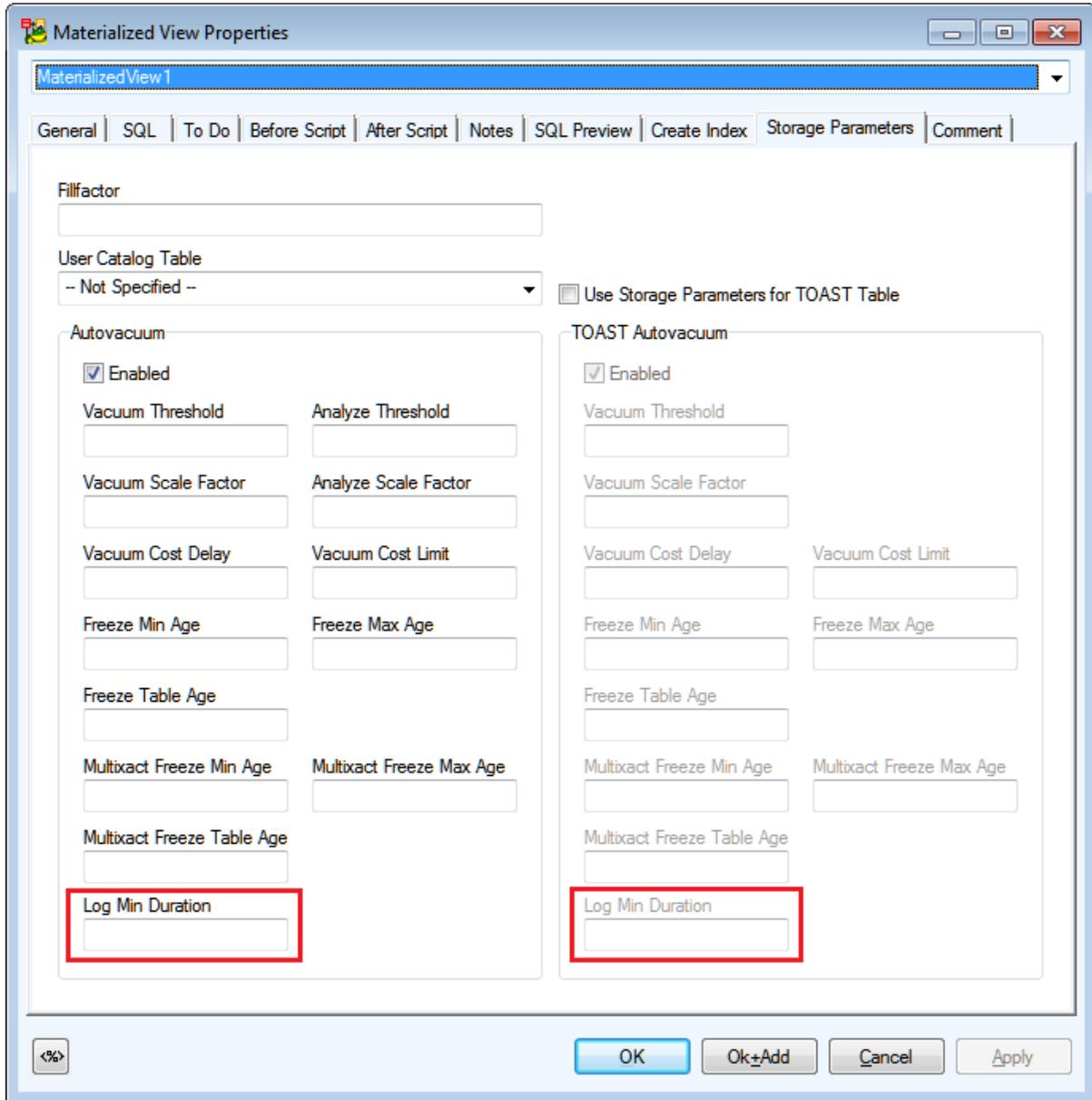
Index properties - **Pages Per Range**, **Gin Pending List Limit**

The screenshot shows the 'Index Properties' dialog for an index named 'Index1'. The 'General' tab is active. The 'Caption' is 'Index1' and the 'Name' is 'Index1'. The 'Index Access Method' is set to 'brin'. The 'Pages Per Range' and 'Gin Pending List Limit' fields are highlighted with a red box. The 'Constraint Expression for Partial Index' field is empty.

Property	Value
Caption	Index1
Name	Index1
Is Exclude Constraint	<input type="checkbox"/>
Unique	<input type="checkbox"/>
Index Access Method	brin
Exclude Constraint Setting	<input type="checkbox"/> Deferrable <input type="checkbox"/> Deferred
Pages Per Range	
Gin Pending List Limit	
Buffering	AUTO
Fast Update	<input checked="" type="checkbox"/>
Constraint Expression for Partial Index	

# Materialized Views

## Storage Parameters - Log Min Duration for Autovacuum and TOAST Autovacuum



# User Groups

## Role Options - *BYPASSRLS*, *NOBYPASSRLS*

```
CREATE ROLE tangerine3 BYPASSRLS;
```

```
CREATE ROLE tangerine5 WITH LOGIN PASSWORD 'jw8s0F4' VALID UNTIL '2005-01-01'  
BYPASSRLS;
```

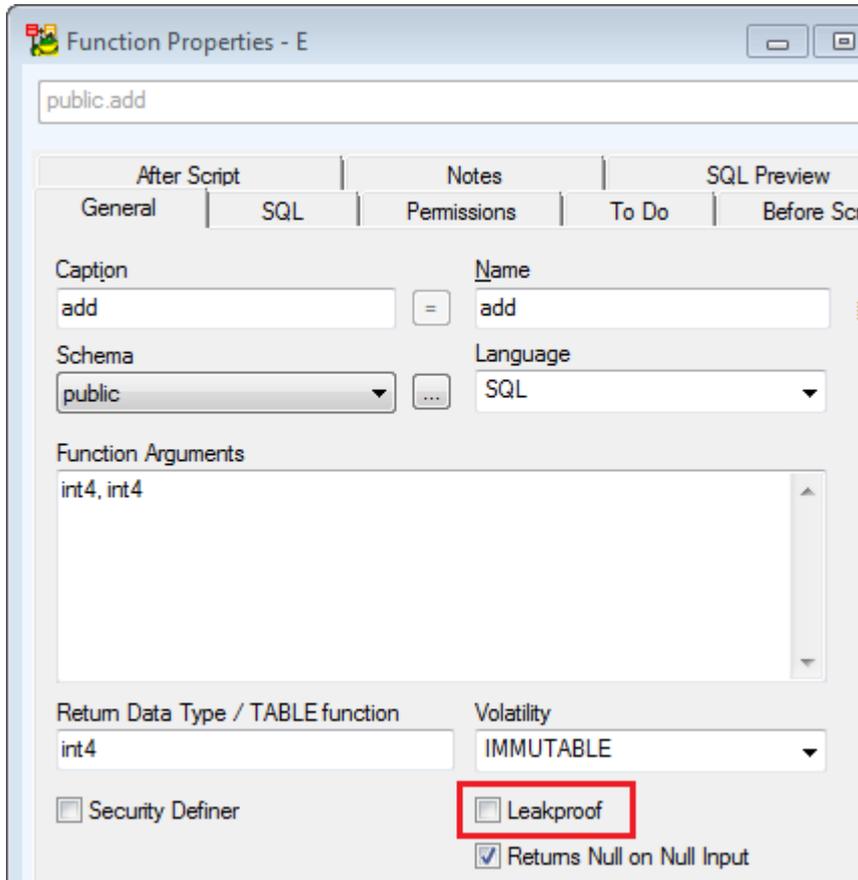
# Change Script Generation

- Support of the following PostgreSQL 9.5 features:

DDL	New Syntax
CREATE SEQUENCE	IF NOT EXISTS
CREATE MATERIALIZED VIEW	IF NOT EXISTS Support for SET and RESET of storage parameters <code>log_autovacuum_min_duration</code> , <code>toast.log_autovacuum_min_duration</code> (integer)
CREATE TABLE	Support for SET and RESET of storage parameters <code>log_autovacuum_min_duration</code> , <code>toast.log_autovacuum_min_duration</code> (integer)
CREATE INDEX	IF NOT EXISTS BRIN method Support for SET and RESET of <code>pages_per_range</code> parameter in WITH section Support for SET and RESET of new GIN method parameter <code>gin_pending_list_limit</code> in WITH section
CREATE FOREIGN TABLE	INHERITS

# Functions

**Leakproof** checkbox on tab **General**.



Generated SQL:

```
create function add4 (integer, integer) RETURNS integer
AS 'select $1 + $2;'
LANGUAGE SQL
LEAKPROOF
RETURNS NULL ON NULL INPUT;
```

## Reverse Engineering - PostgreSQL 9.5

See [Reverse Engineering - PostgreSQL 12](#) for more information.

## Script Generating - PostgreSQL 9.5

See [Script Generation - PostgreSQL 12](#) for more information.

# Specifics - PostgreSQL 10

## Reverse Engineering - PostgreSQL 10

See [Reverse Engineering - PostgreSQL 12](#) for more information.

## Script Generating - PostgreSQL 10

See [Script Generation - PostgreSQL 12](#) for more information.

# Specifics - PostgreSQL 11

## Reverse Engineering - PostgreSQL 11

See [Reverse Engineering - PostgreSQL 12](#) for more information.

## Script Generating - PostgreSQL 11

See [Script Generation - PostgreSQL 12](#) for more information.

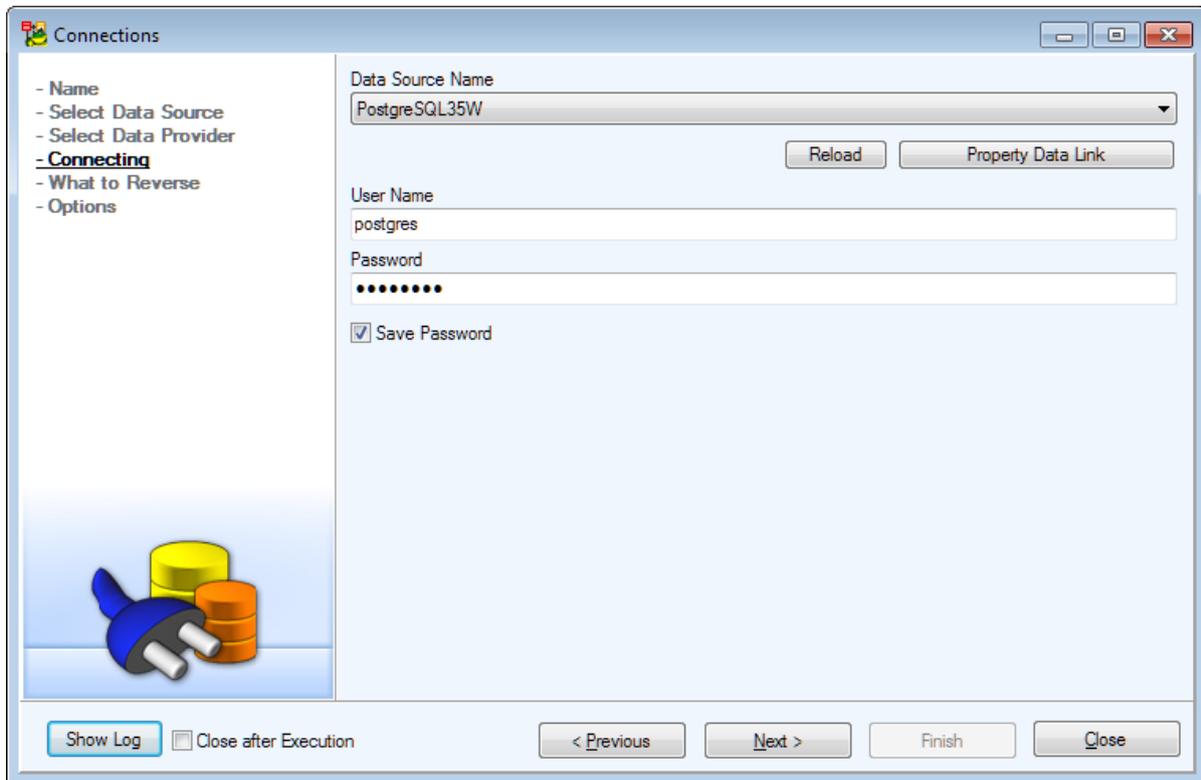
# Specifics - PostgreSQL 12

# Reverse Engineering - PostgreSQL 12

Available **Data Providers** are:

- **Connection via ODBC**
- **Native Connection**

**Connection via ODBC:**



**Native Connection:**

**Connections**

- Name
- Select Data Source
- Select Data Provider
- **Connecting**
- What to Reverse
- Options

**Host**

Host Name: 10.11.40.40      Port: 5444

**User**

User Name: qa      Password: ●●

Save Password

**Database**

Database Name: qa

**SSL**

SSL Mode: Disable

SSL CA File: [ ] ...

SSL Cert File: [ ] ...

SSL Key File: [ ] ...

**Connection String**

HostName=10.11.40.40; Port=5444; Database=qa; UID=qa;

Hide Log     Close after Execution    < Previous    Next >    Finish    Close

Id	Date	Time	Message
----	------	------	---------

# Script Generation - PostgreSQL 12

DDL Script Generation of PostgreSQL 10

What to Generate | Detail Settings | Referential Integrity | Select List

Use Quotation Marks  
 Generate Schema-Qualified Names  
Text Case Selection:  
Preserve Case

Generate Check Constraint Names Used in Domains  
 Generate Empty Comments as NULL  
 Generate "Create or Replace" in Generated Views and Functions

Drop Object Integrity

Drop Tables Integrity  
RESTRICT

Drop Schemas Integrity  
RESTRICT

Generate IF EXISTS in DROP statements  
 Create Comments  
 Grant Roles to User/Role  
 Generate Permissions Only to Object Types Selected on Tab What to Generate

Terminator :  
Dollar Quoting Type in Function Definition String \$\$

Order of Generated Objects

You can change order of generated objects. Click Edit and define how objects will be generated.

Edit...

Show Preview

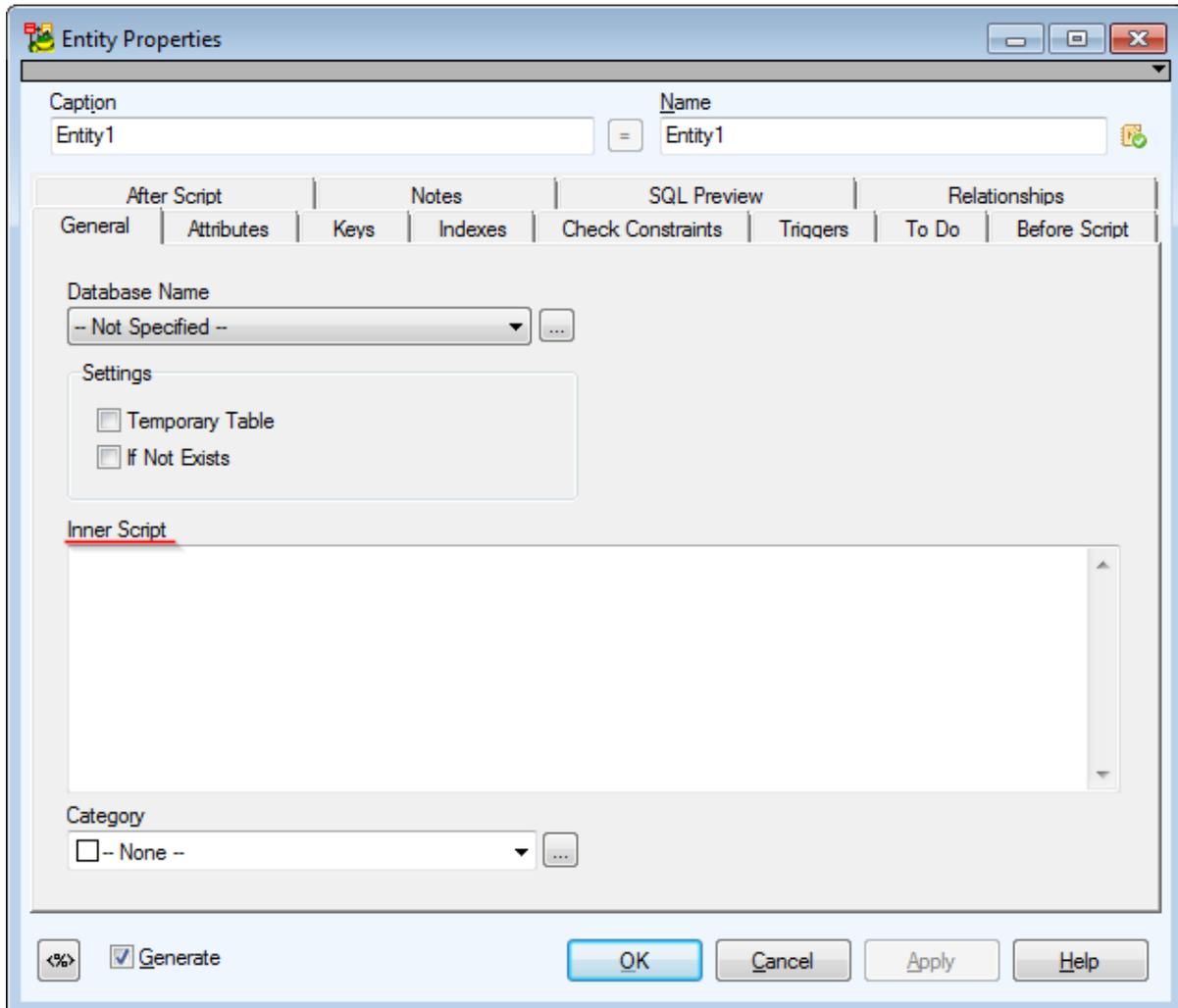
OK Cancel

# Specifics - SQLite 3.7

Though SQLite database is case insensitive, Reverse Engineering is case sensitive. This leads to a problem that e.g. when a column name is "ATR" and an index is defined with column name "Atr", the names do not get paired and an error message is thrown.

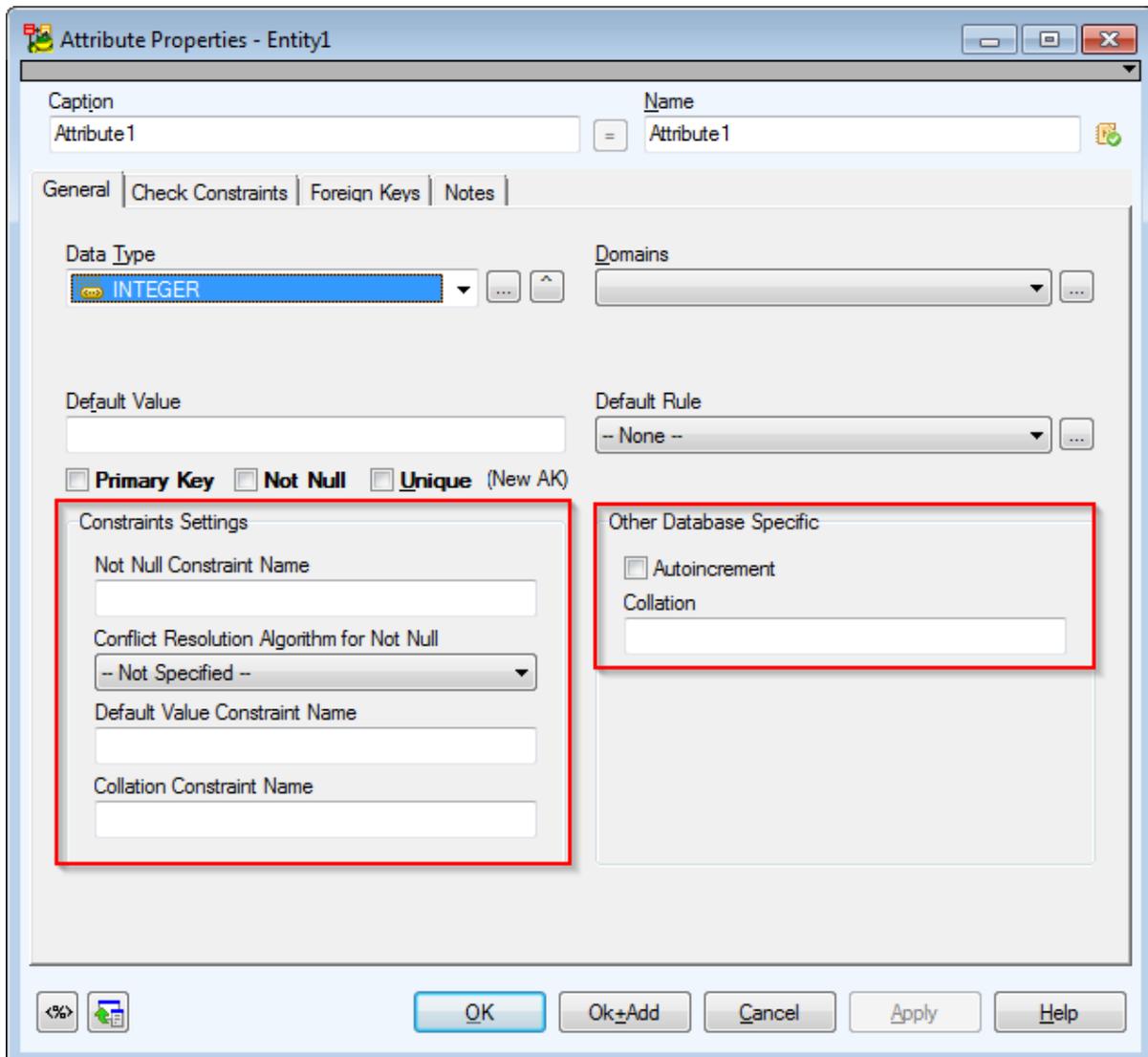
If some object name includes UTF characters, the object does not get loaded during Reverse Engineering.

## Entity



**Inner Script**box — e.g. for defining constraint in text. Code in Inner Script is generated in the body of CREATE TABLE command, at the end after all attributes and constraints. During reverse engineering, foreign key constraint definitions, which are not possible to visualize by relationship, are loaded here.

# Attribute

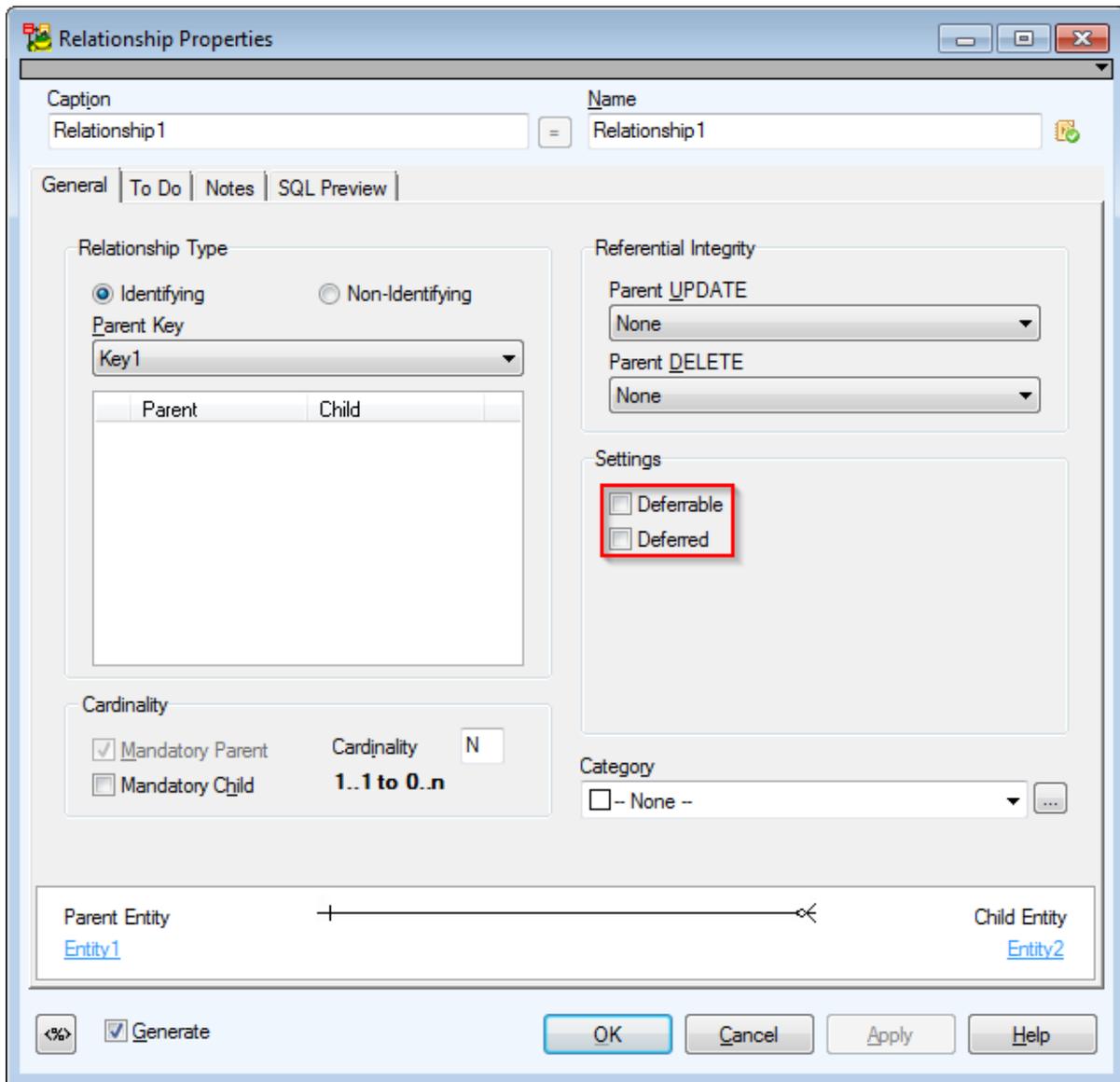


**Autoincrement**— Data type must be set to *INTEGER* and the particular attribute must be in a single-attribute primary key. Note that in SQLite syntax, Autoincrement is defined as a part of a column constraint of a primary key statement but in Toad Data Modeler it is set directly in the Attribute.

## Collation

3 built-in collating functions: BINARY, NOCASE, and RTRIM, defined in text in particular box. For keys and indexes you can define Collation for the selected item in Key/Index Properties dialog, Attributes/Items tab, in the **Collation** column (see the screenshots below.)

# Relationship



Property MATCH is available in syntax but it is not used and therefore not supported in Toad Data Modeler.

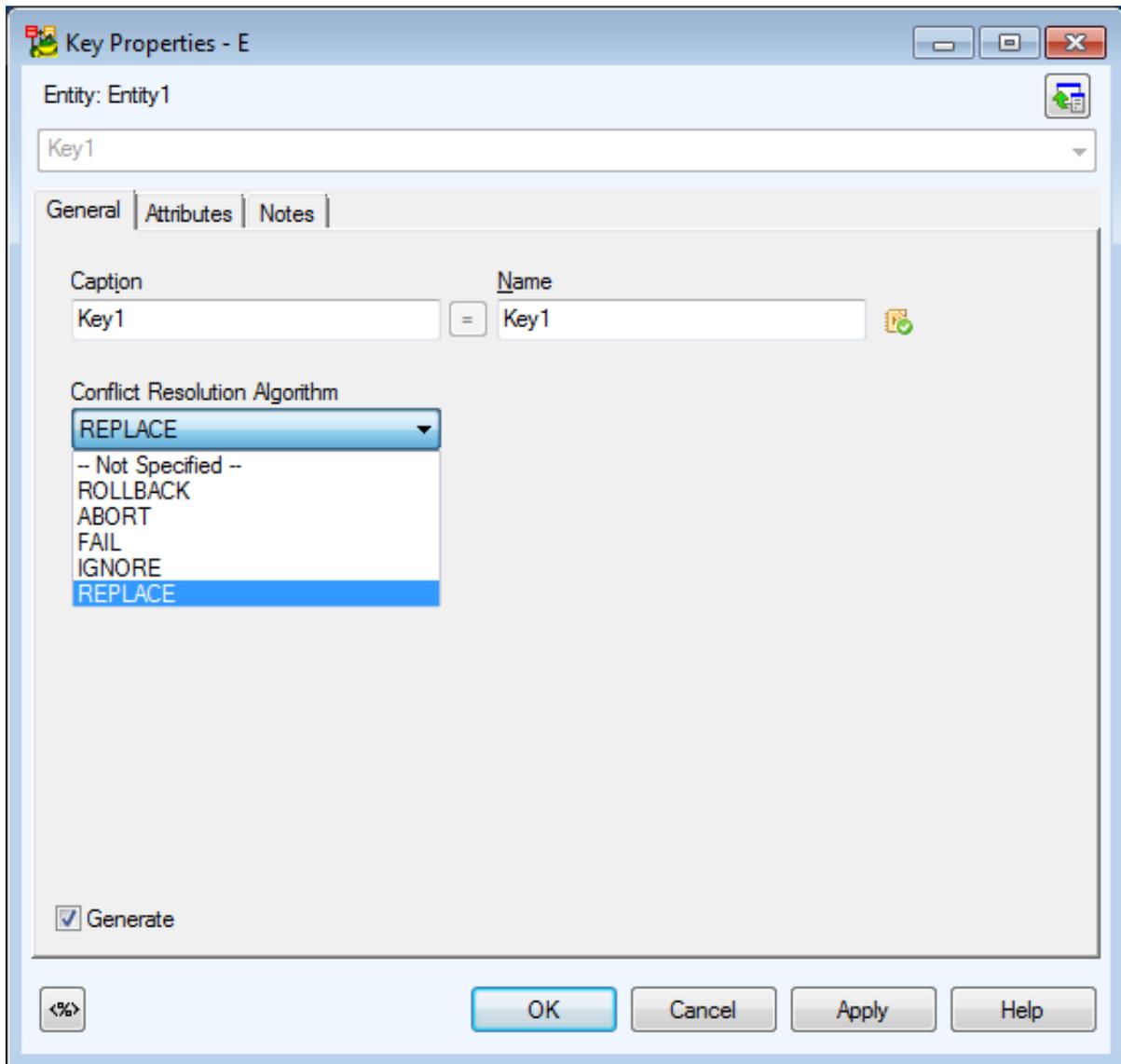
Relationships can be defined only by CREATE TABLE command.

It is not possible to have relationships between entities from different databases.

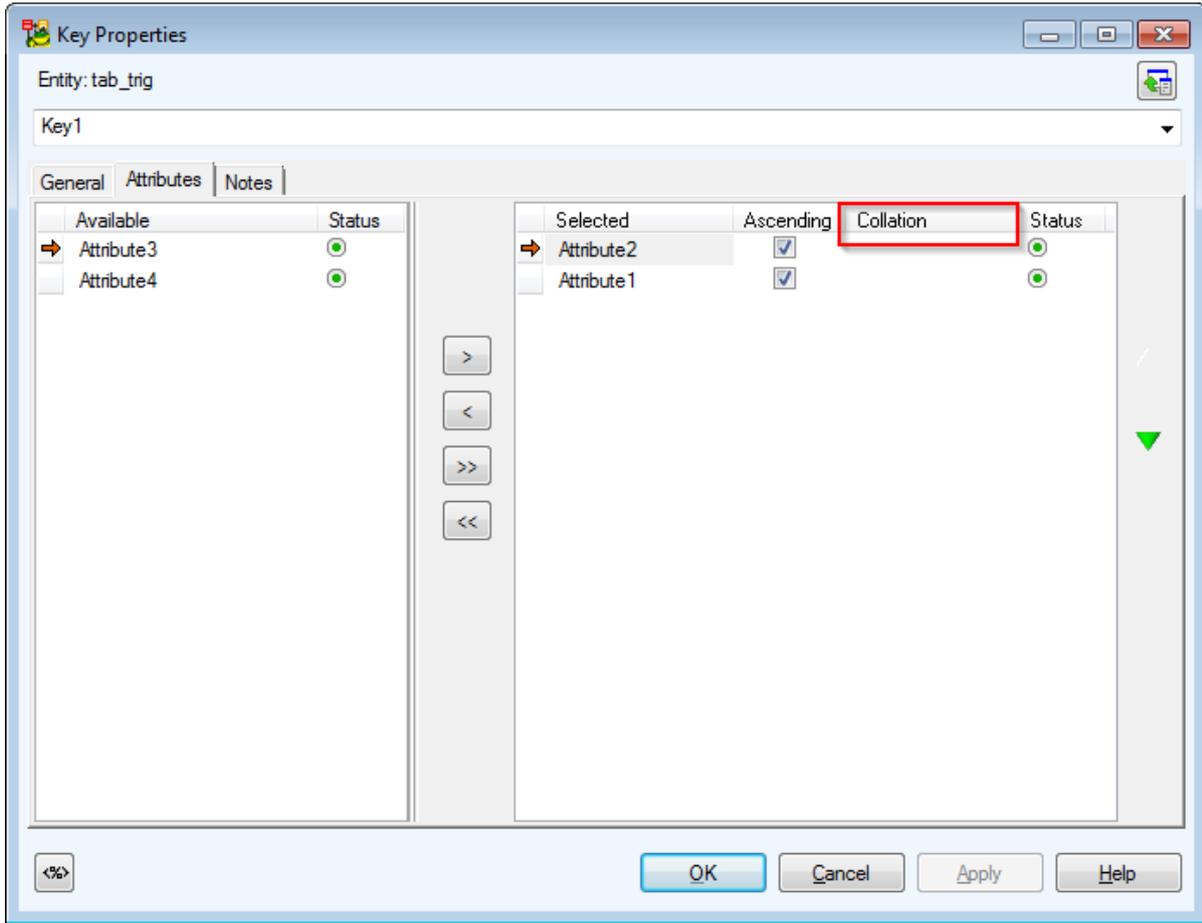
If a foreign key is created, it can contain a reference to a non-existing table/name. During reverse engineering, thus foreign key is loaded in the Inner Script box of the particular entity.

**i** Note: By default, foreign keys are not supported in SQLite; they can be created but are disabled. They can be enabled by command `PRAGMA foreign_keys=ON;`, but default value after SQLite start is OFF!

# Key

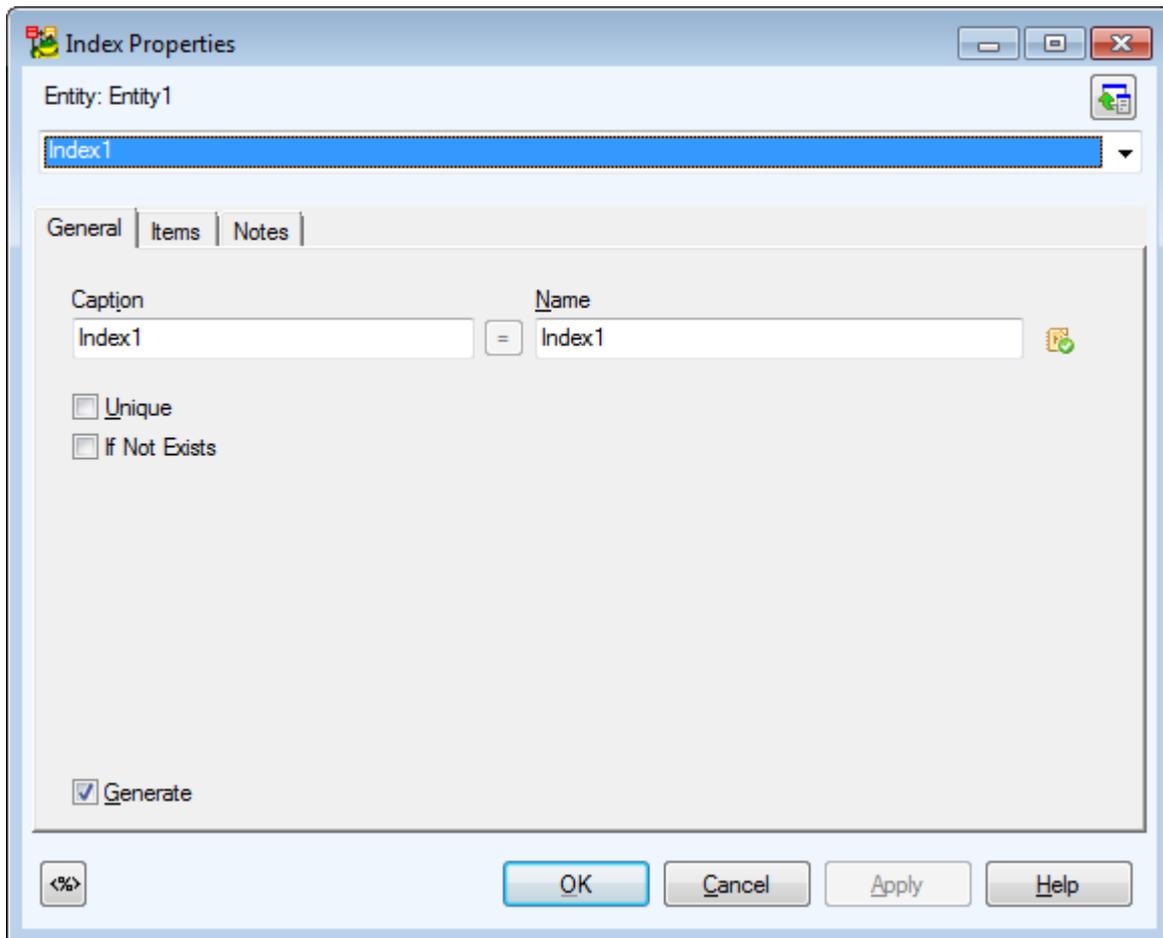


New Conflict Resolution Algorithm box.



On tab **Attributes**, you can define Collation for the selected item in the **Collation** column.

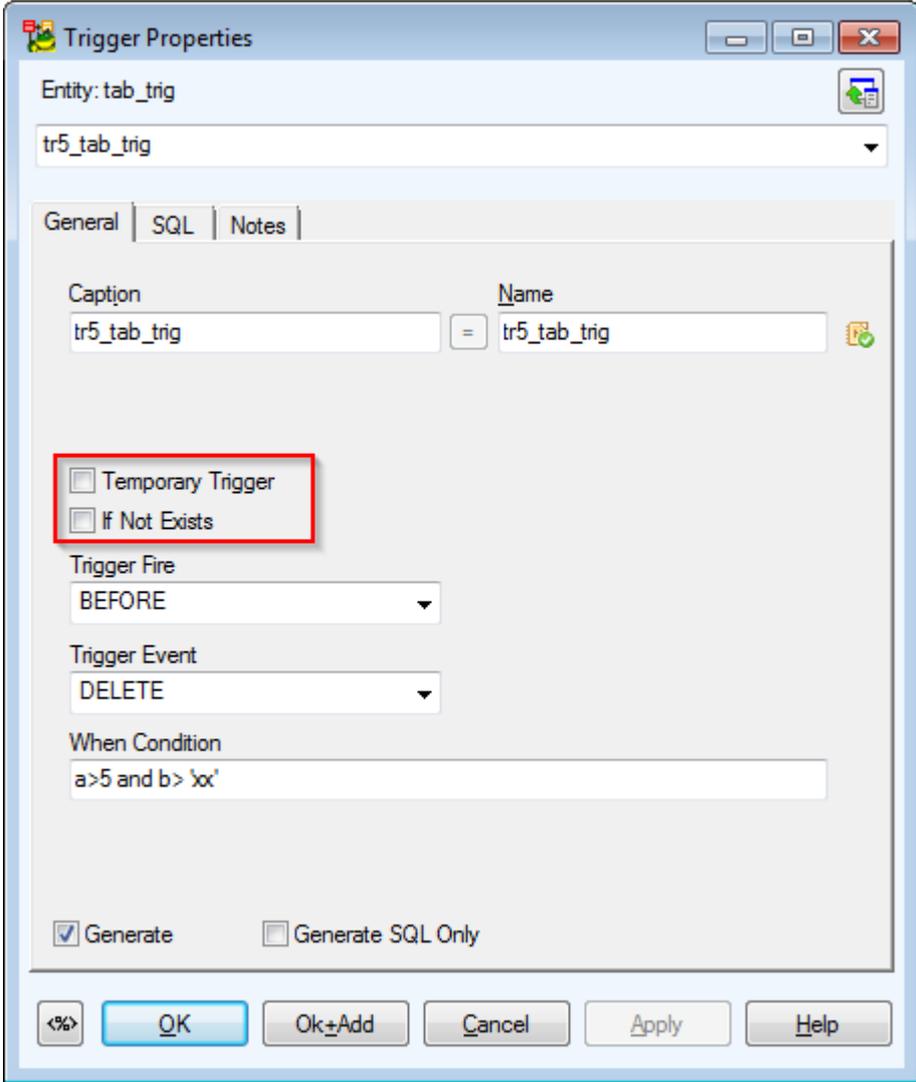
# Index



New **If Not Exists** checkbox.

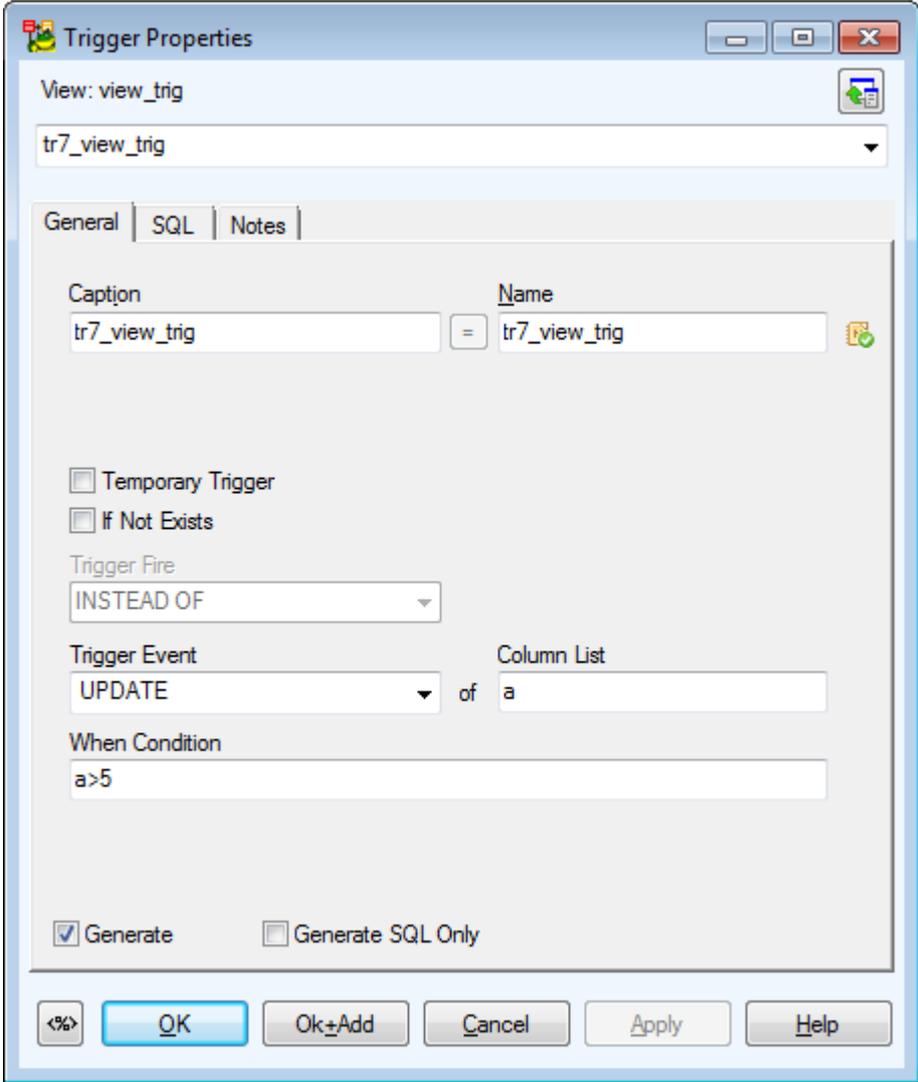
On tab **Items**, you can define Collation for the selected item in the **Collation** column.

# Trigger - Entity



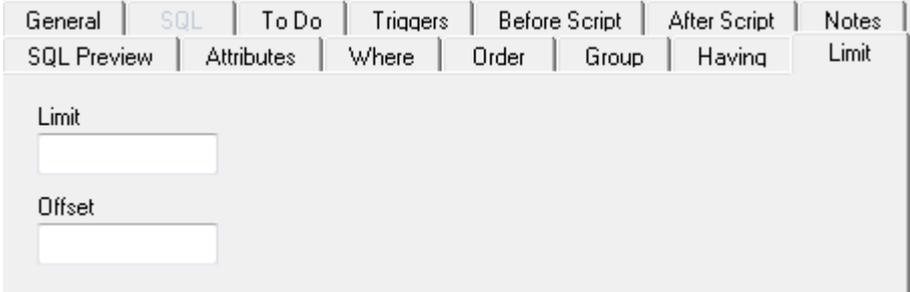
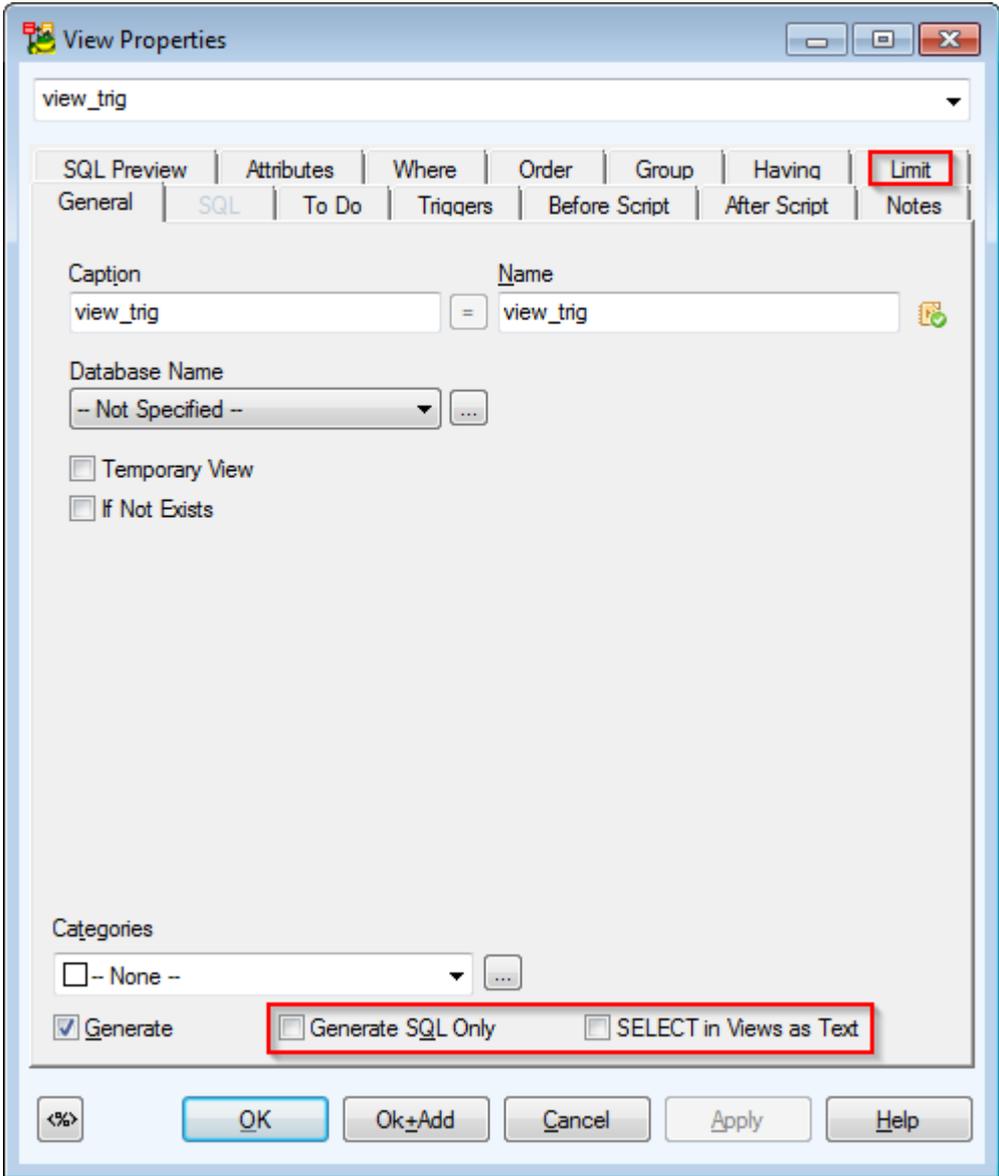
New checkboxes **Temporary Trigger** and **If Not Exists**.

# Trigger - View



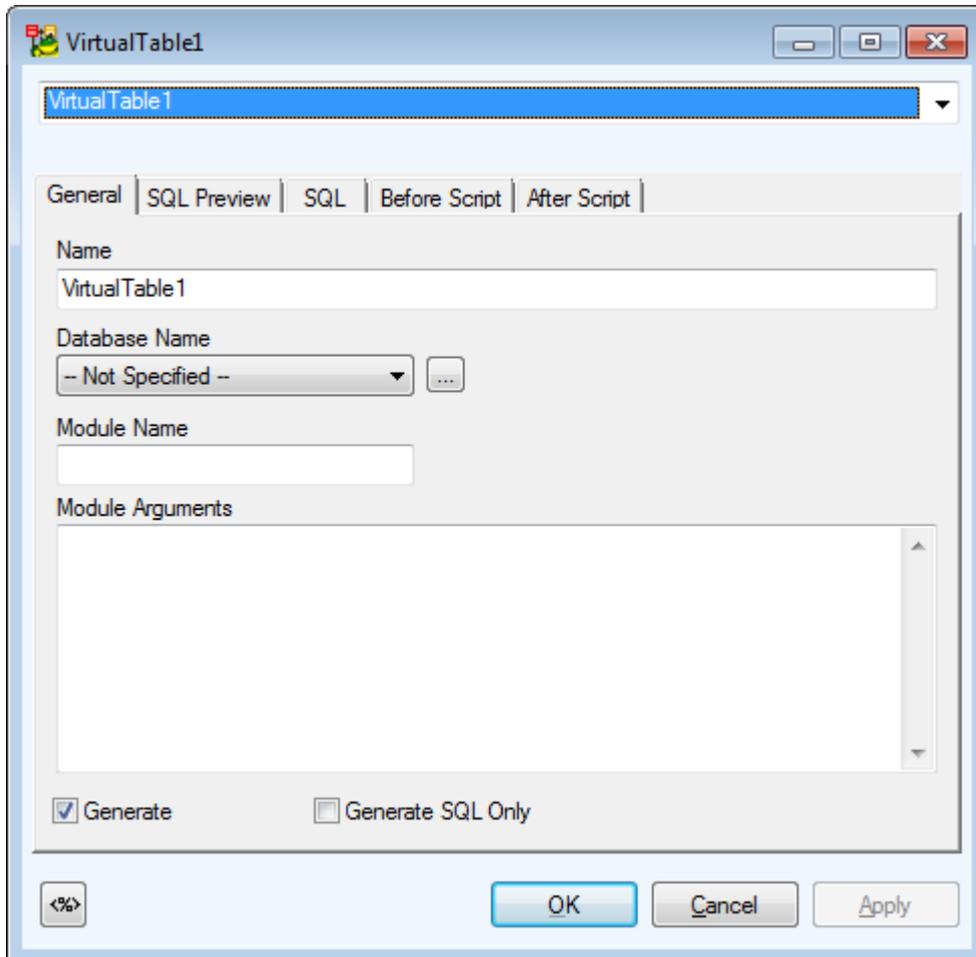
View Trigger has **Trigger Fire** box disabled with *INSTEAD OF* set.

# View



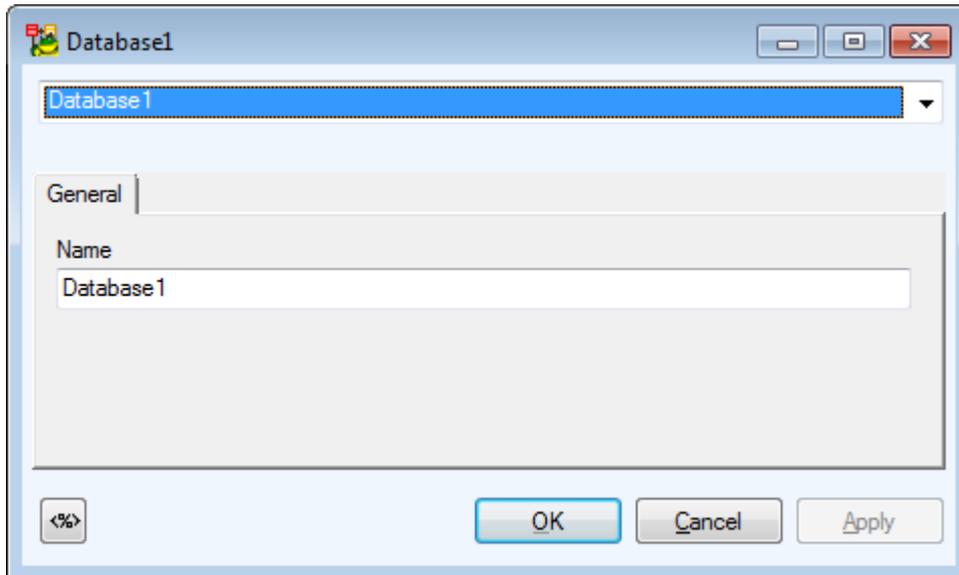
The content of a new **Limit** tab is generated in script only if **Generate SQL Only** and **SELECT in Views as Text** are not selected.

## Virtual Table



During Reverse Engineering, virtual table is loaded as text - CREATE VIRTUAL TABLE.

# Database



Database serves only for listing function. Available for Entity, View, Virtual Table. It denotes to which database the object belongs.

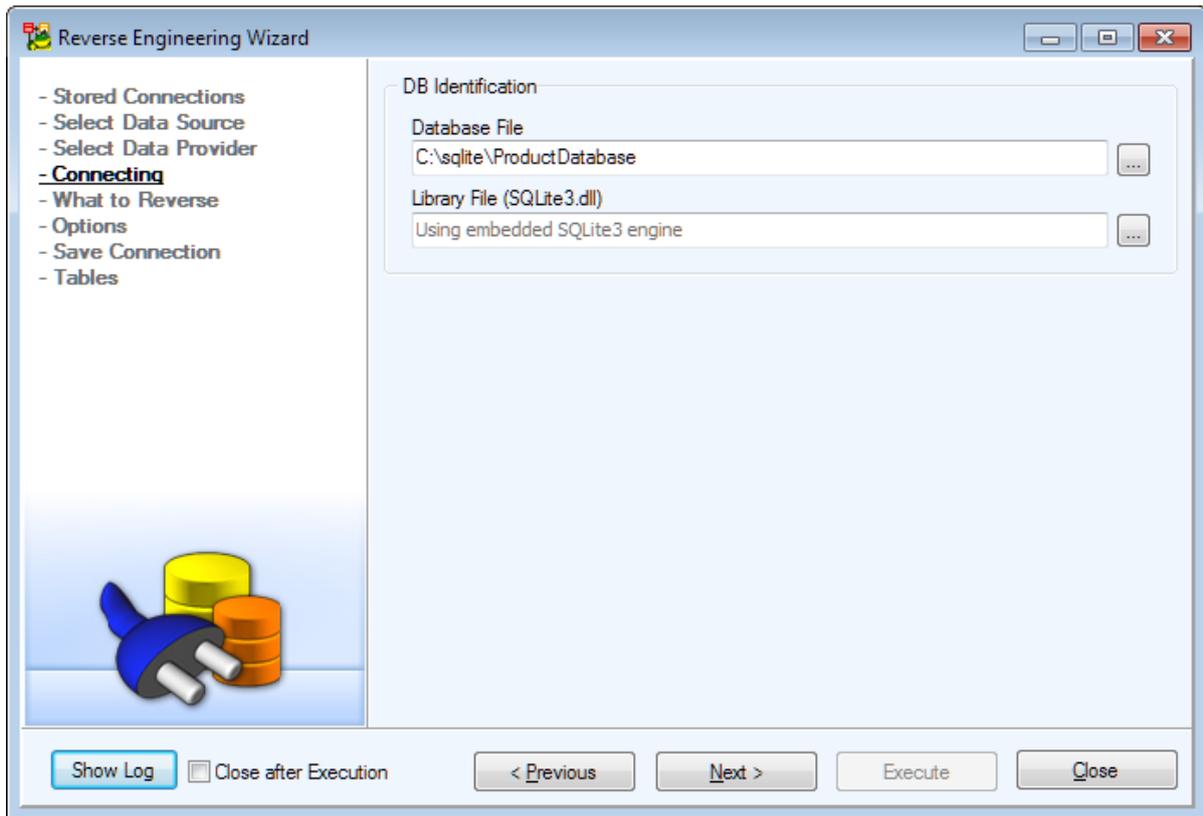
During reverse engineering, objects from only one database are loaded. Therefore, database name is not loaded during reverse engineering.

## Reverse Engineering - SQLite 3.7

Available **Data Providers** are:

- **Native Connection**

**Native Connection:**



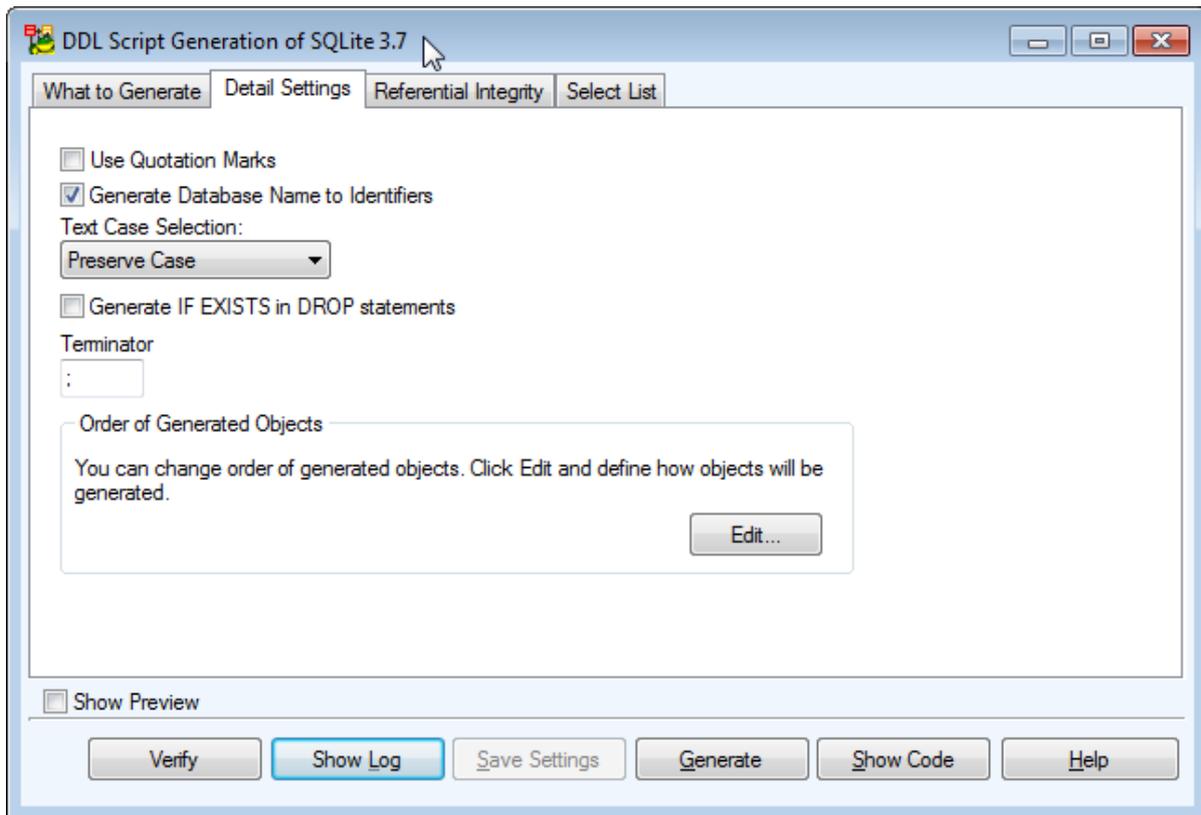
Though SQLite database is case insensitive, Reverse Engineering in Toad Data Modeler is case sensitive. This leads to a problem that e.g. when a column name is "ATR" and an index is defined with column name "Atr", the names do not get paired and an error message is thrown.

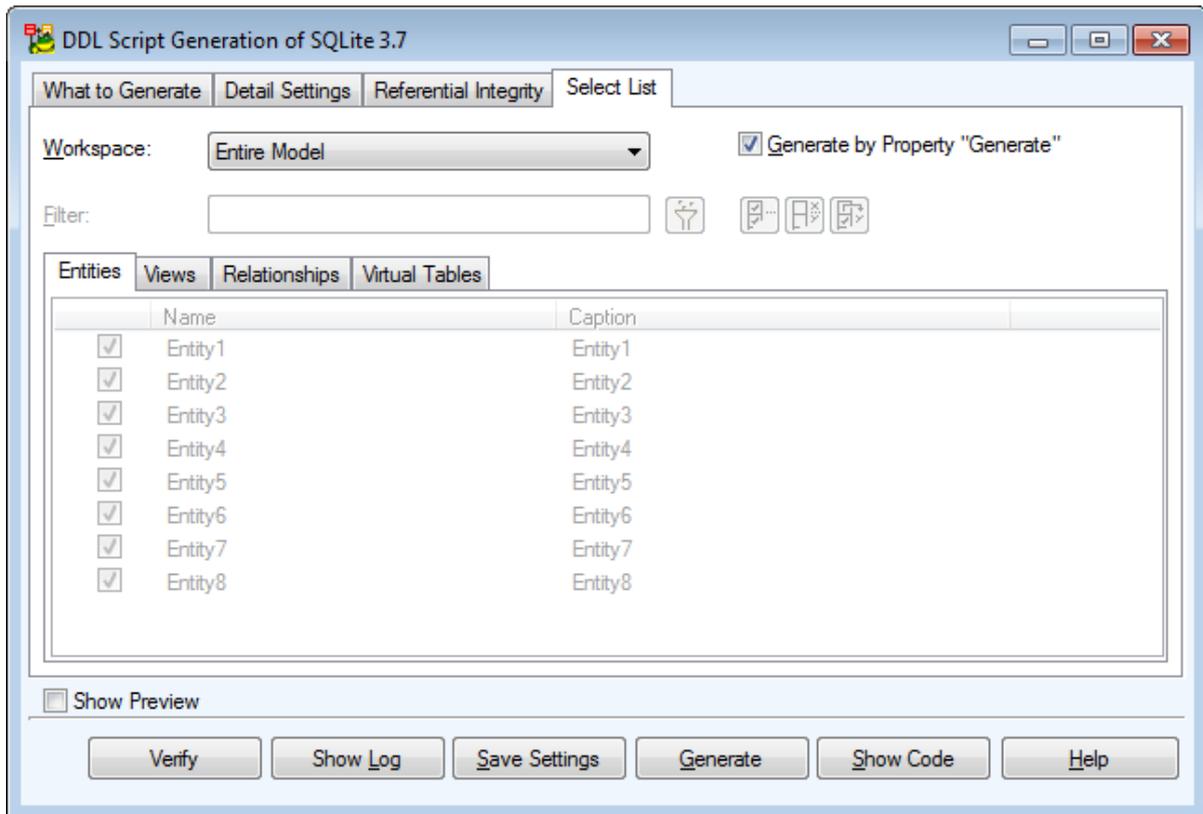
If some object name includes UTF characters, then the object does not get loaded during Reverse Engineering. Appropriate SQLite3.dll library is necessary for successful connection.

**i** Note:

1. On page Options, the **Load Some Objects as SQL Sentence Only** checkbox is applicable for Views, Triggers and Virtual Tables. If selected, triggers are loaded as they are saved in system table.
2. During Reverse Engineering, objects from only one database are loaded. Therefore, database name is not loaded during Reverse Engineering.

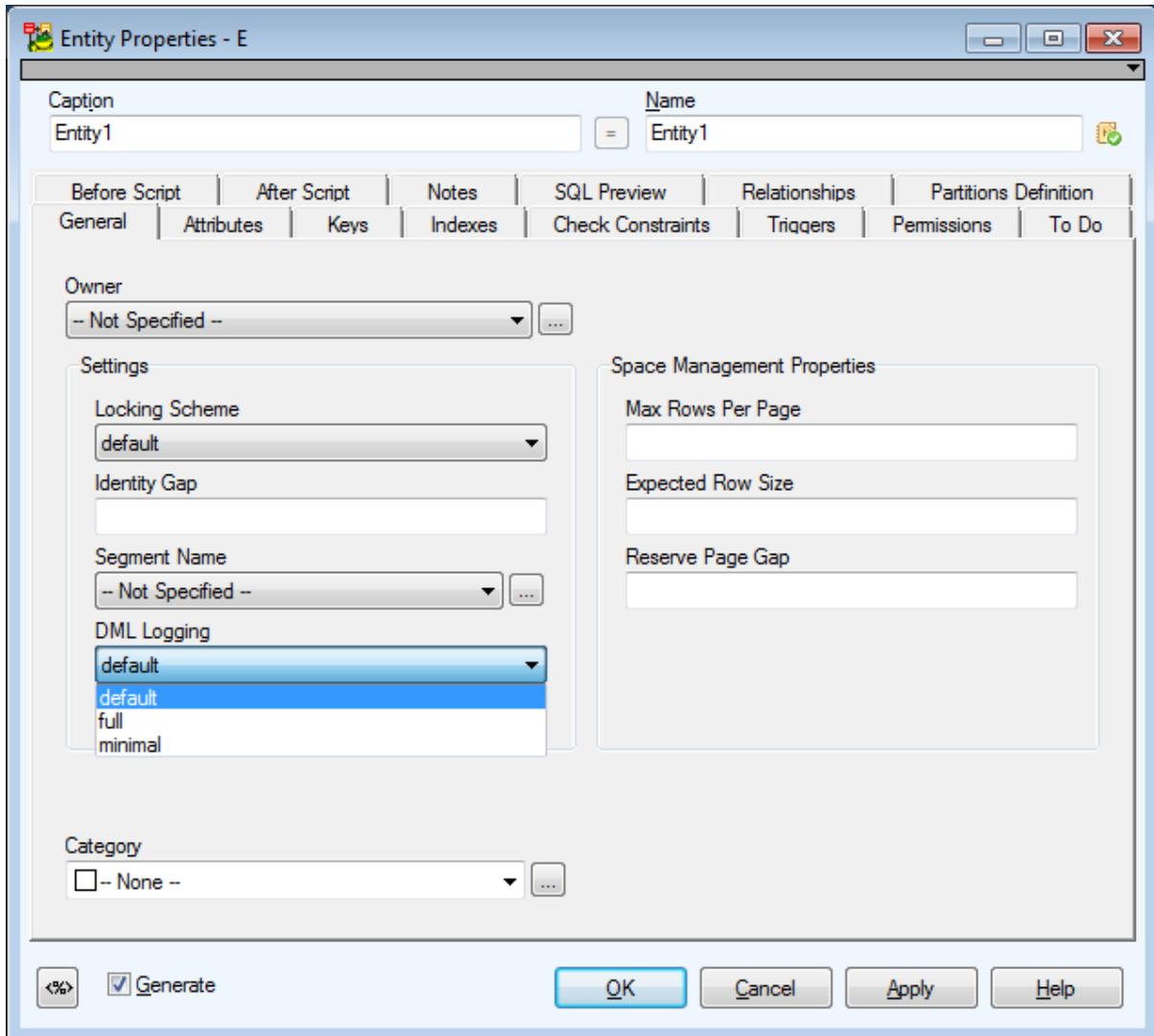
# Script Generation - SQLite 3.7





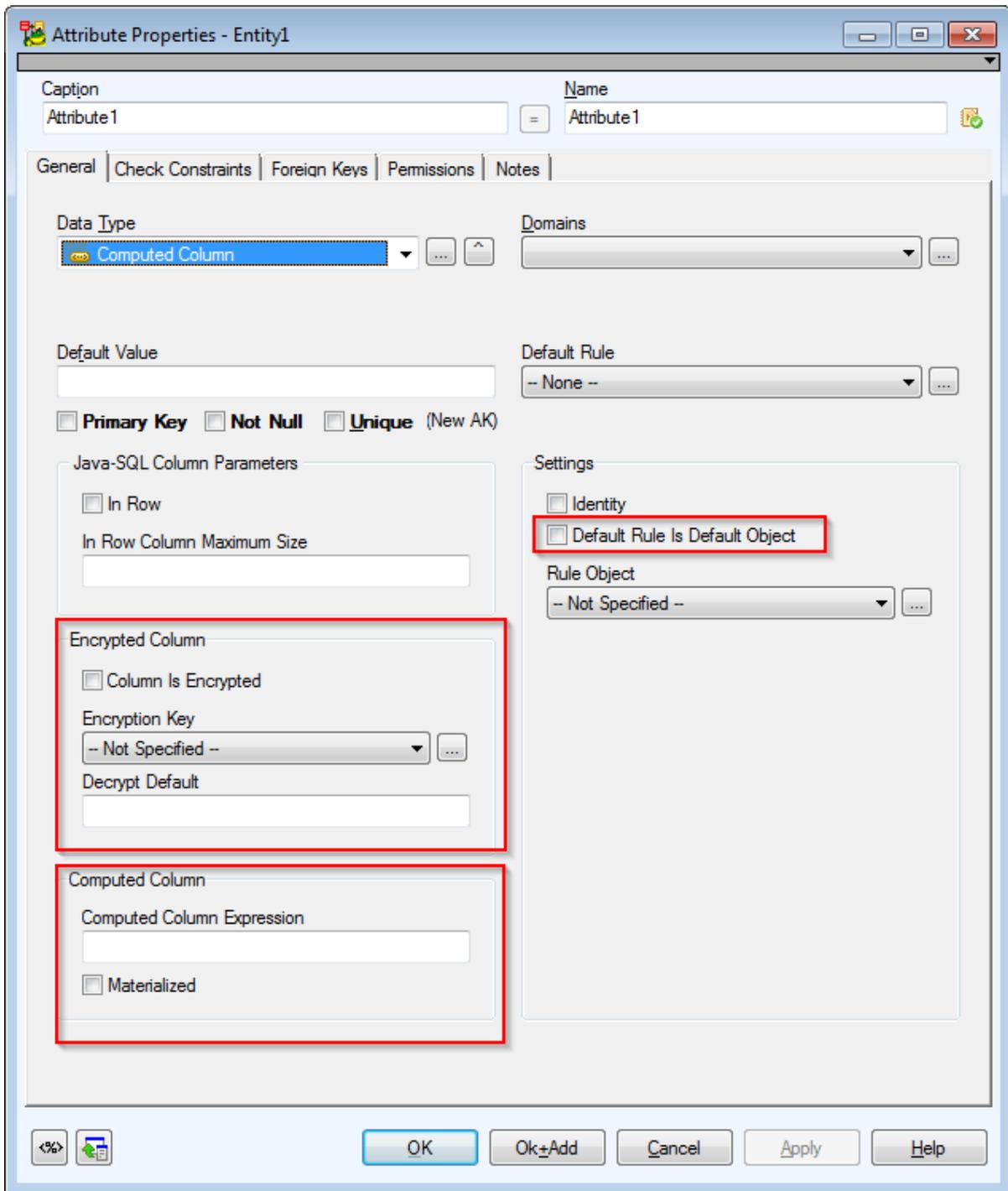
# Specifics - Sybase ASE 15.5

## Entity



DML Logging box with *default* / *full* / *minimal* values.

# Attribute



New **Encrypted Column** area with **Encryption Key** and **Decrypt Default**.

*Computed Column* - It is not a data type. Select this item from **Data Type** box if you want to set column as computed.

**Default rule Is Default Object** - Select this checkbox if you want to bind default rule (known as default object in Sybase ASE) to attribute.

**Rule Object** - Select a rule object that will be bound to attribute.

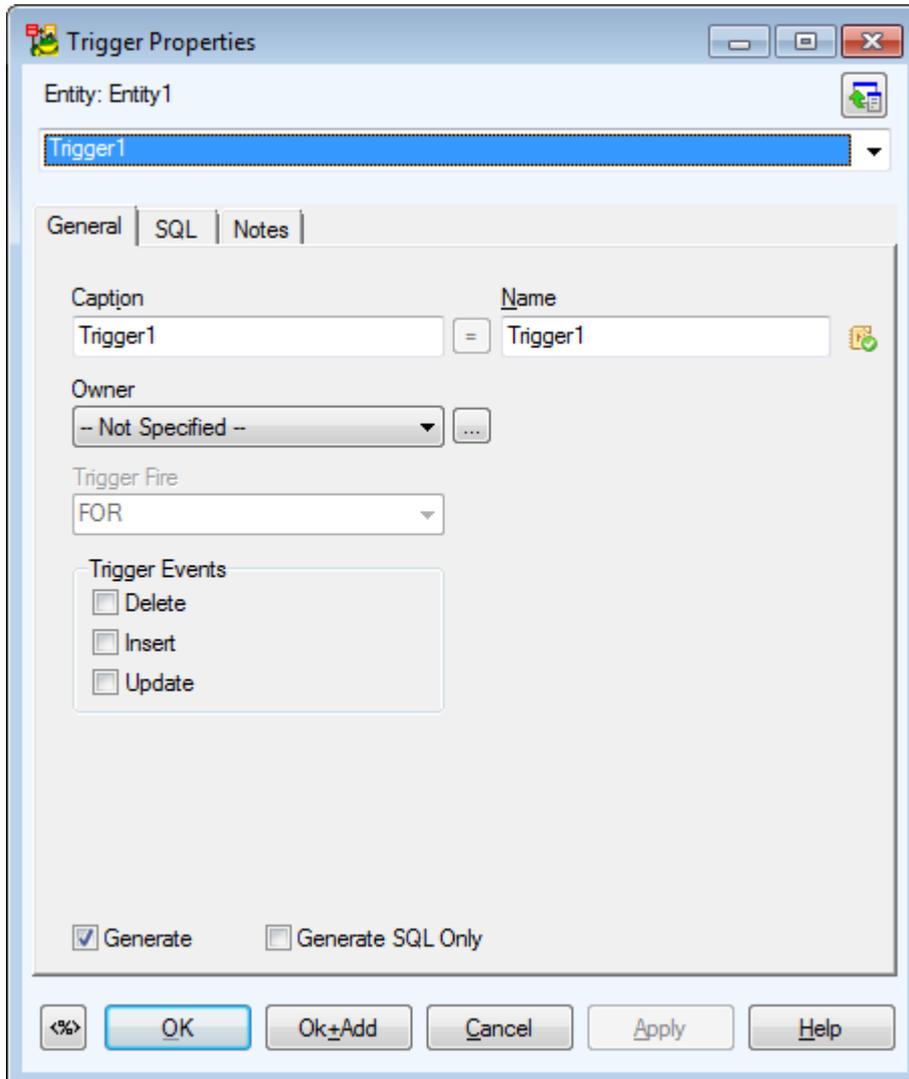
## Encryption Keys

The screenshot shows a dialog box titled "EncryptionKey1". At the top, there is a dropdown menu with "EncryptionKey1" selected. Below this are two tabs: "General" and "SQL Preview". The "General" tab is active and contains the following fields and options:

- Name:** EncryptionKey1
- Owner:** -- Not Specified --
- Database Name:** (empty text box)
- Algorithm:** AES
- Key Length:** 128
- Password:** (empty text box)
- Initialization Vector (random)
- Padding (random)
- Generate
- Default

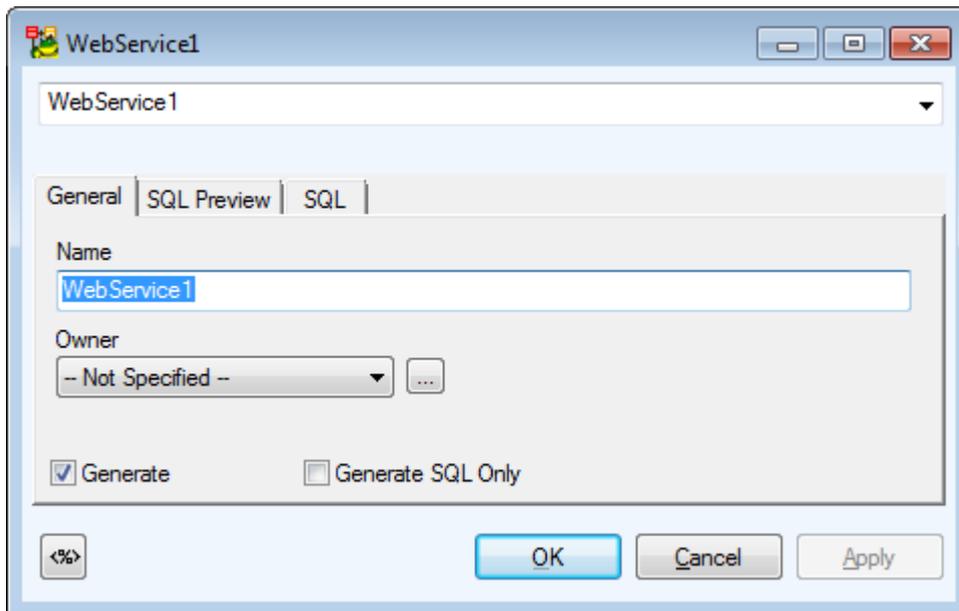
At the bottom of the dialog, there are four buttons: "<%", "OK", "Cancel", and "Apply".

# Trigger



New inactive (informational) box **Trigger Fire** .  
Trigger is now available also for Views.

# Web Services



See other objects in Model Explorer:

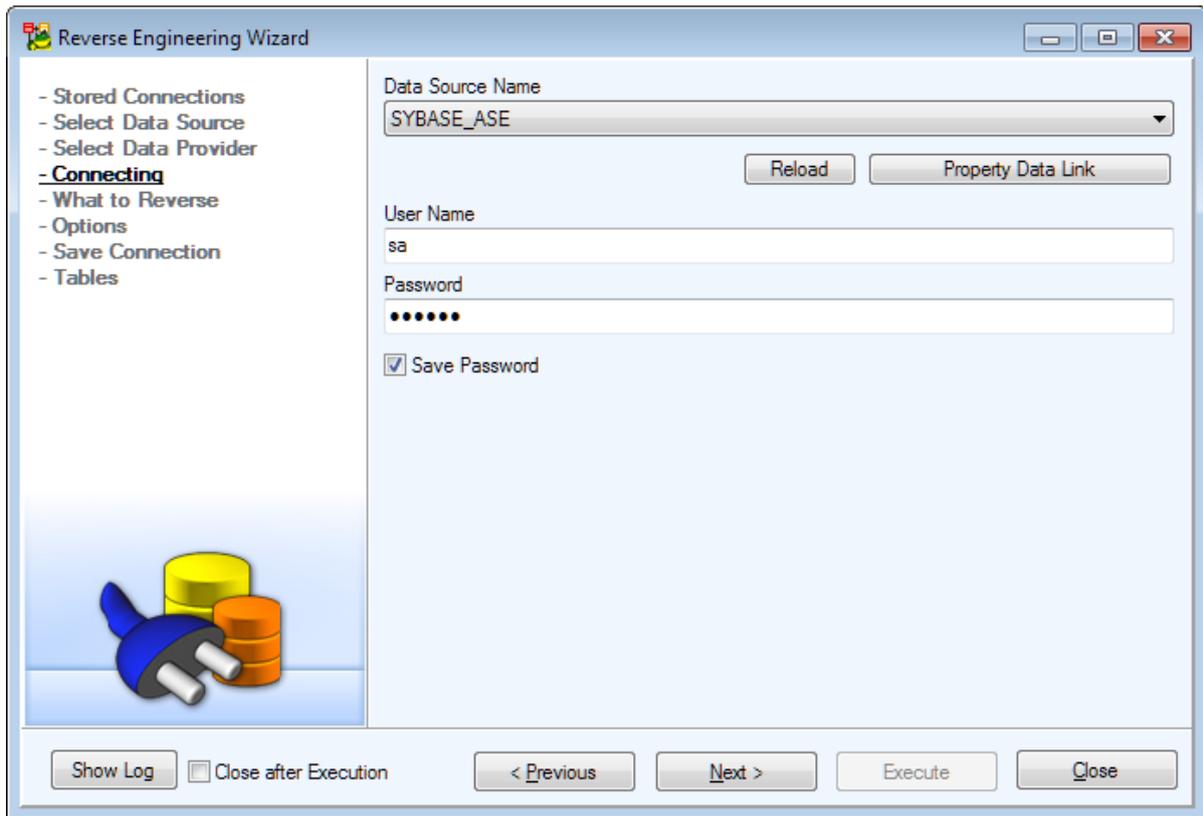
- Segments
- Encryption Key
- Web Service

## Reverse Engineering - Sybase ASE 15.5

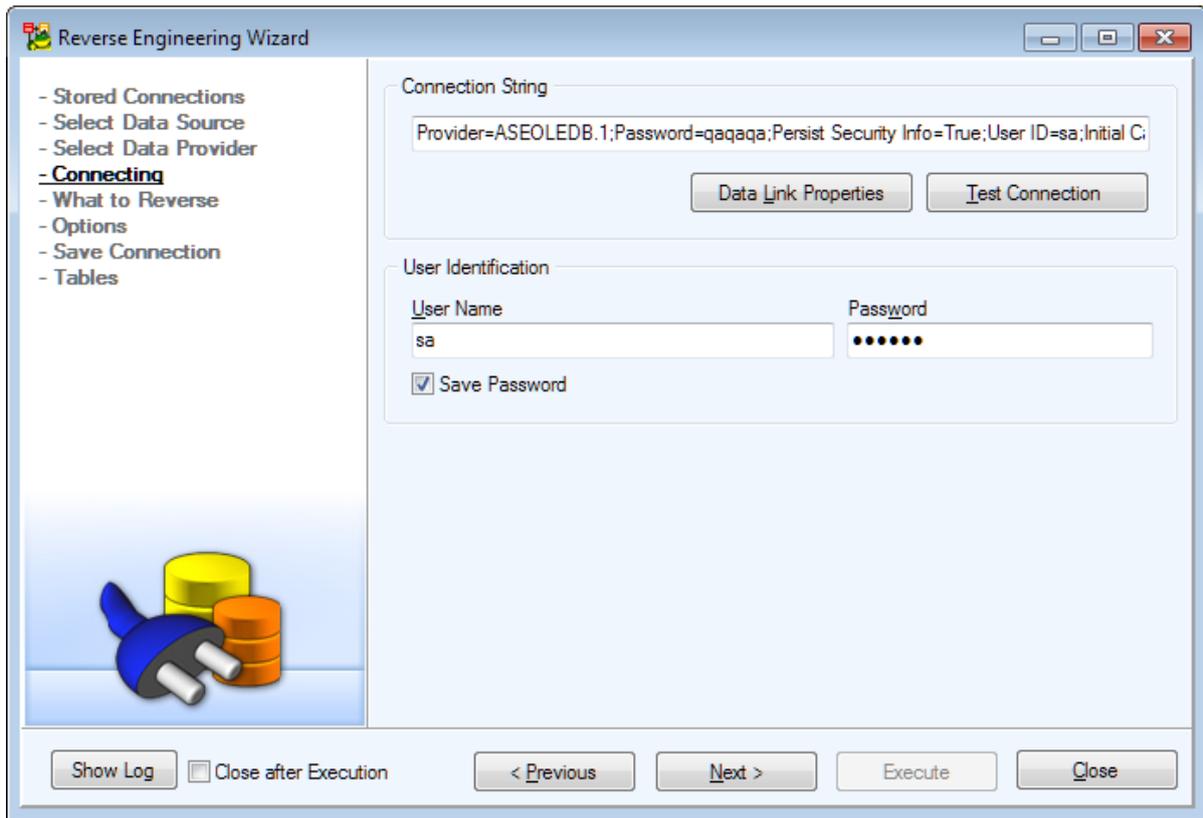
Available **Data Providers** are:

- **Connection via ODBC**
- **Connection via ADO**
- **Native Connection**

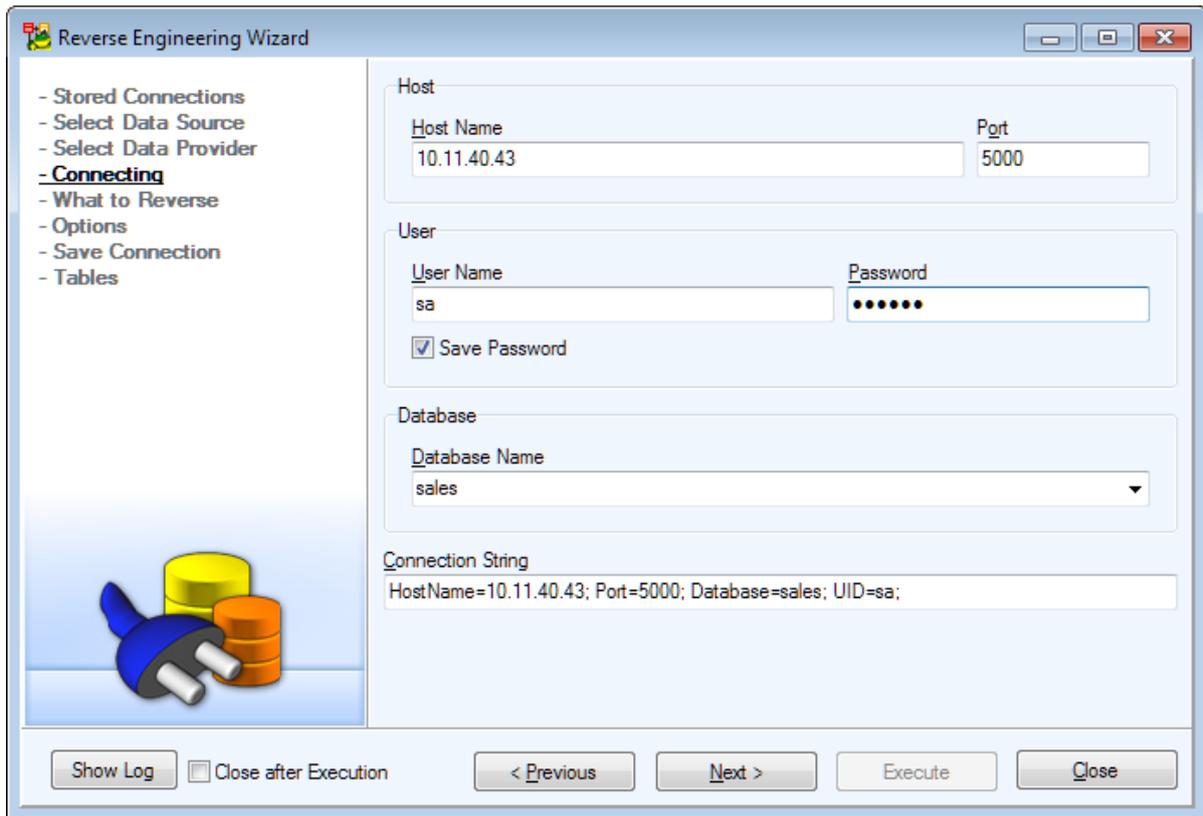
**Connection via ODBC**



Connection via ADO

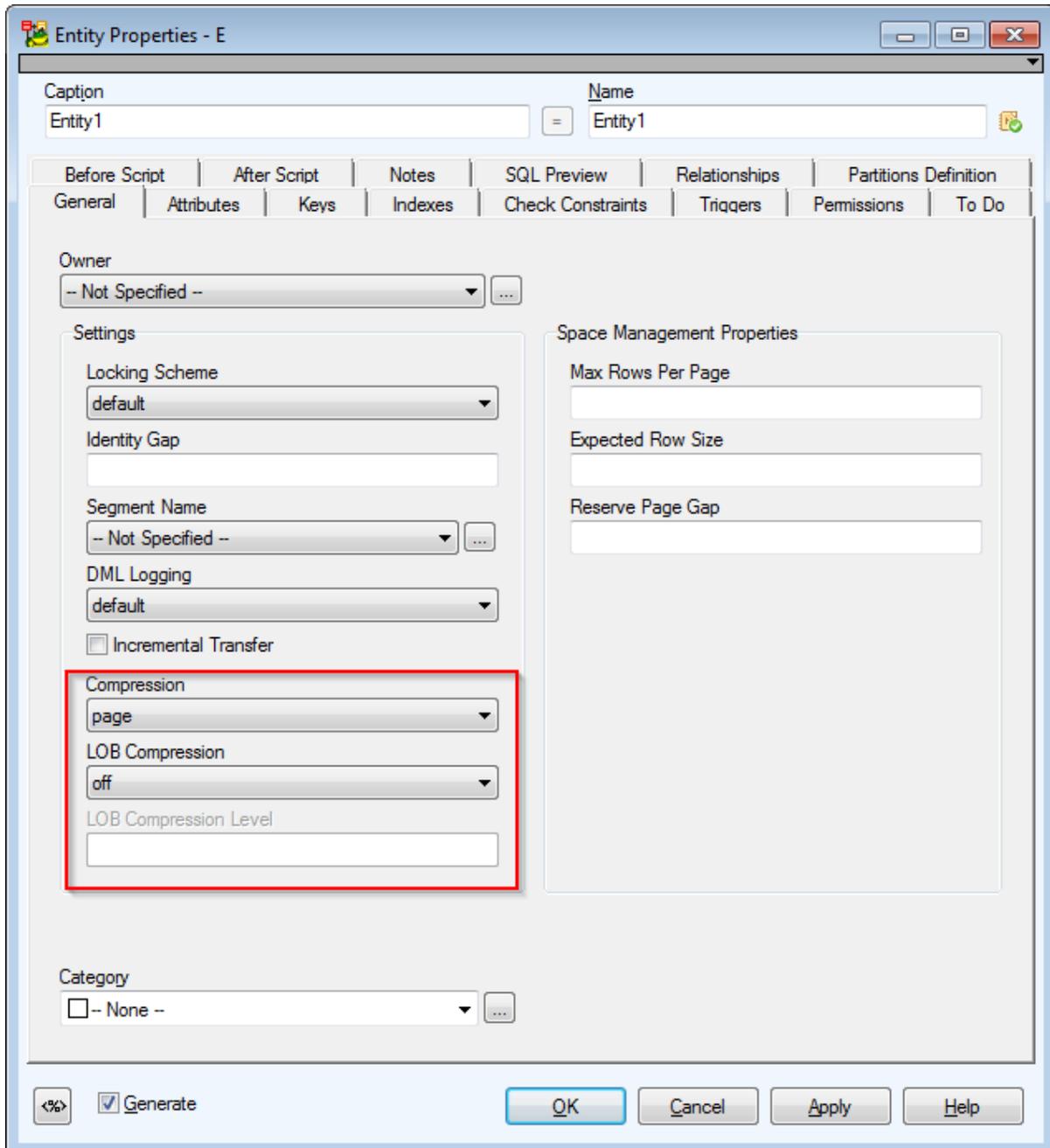


**Native Connection:**



# Specifics - Sybase ASE 15.7

## Entity



Compression box: none, page, row.

Examples:

#### Table Compression

```
create table t01 (a varchar(50) not null, b varchar(50) not null) with compression = none
```

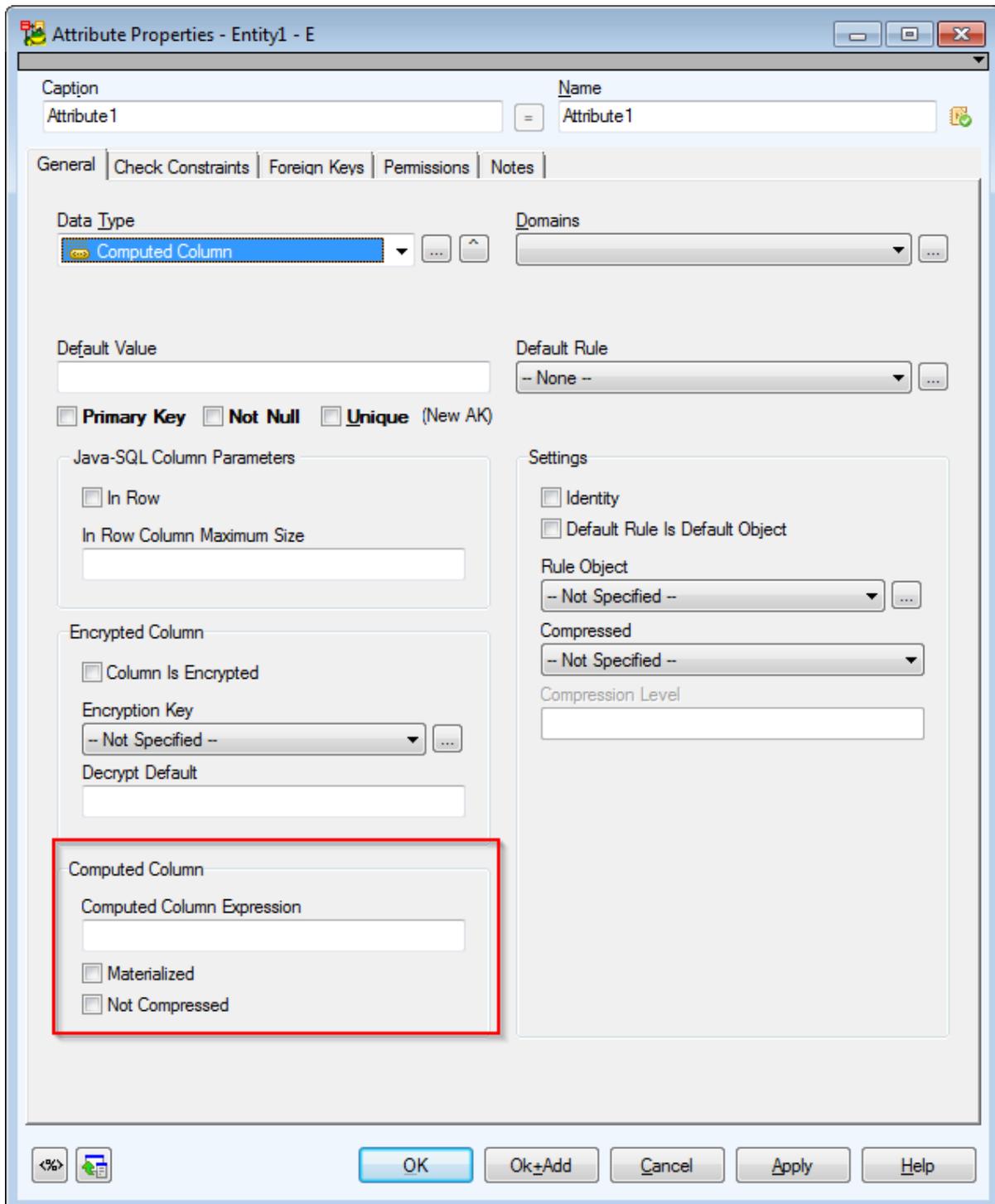
#### Table Lob Compression + level

```
create table t02 (a varchar integer) with lob_compression = 0
```

#### Column Compressed

```
create table tab03 (a text not compressed)
```

# Attribute



From the **Data Type** box, select *Computed Column* and define other properties in the **Computed Column** area.

**Example:**

create table tab04 (acompa as 'orderno' materialized not compressed) with lob\_compression = 5

See other objects in Model Explorer:

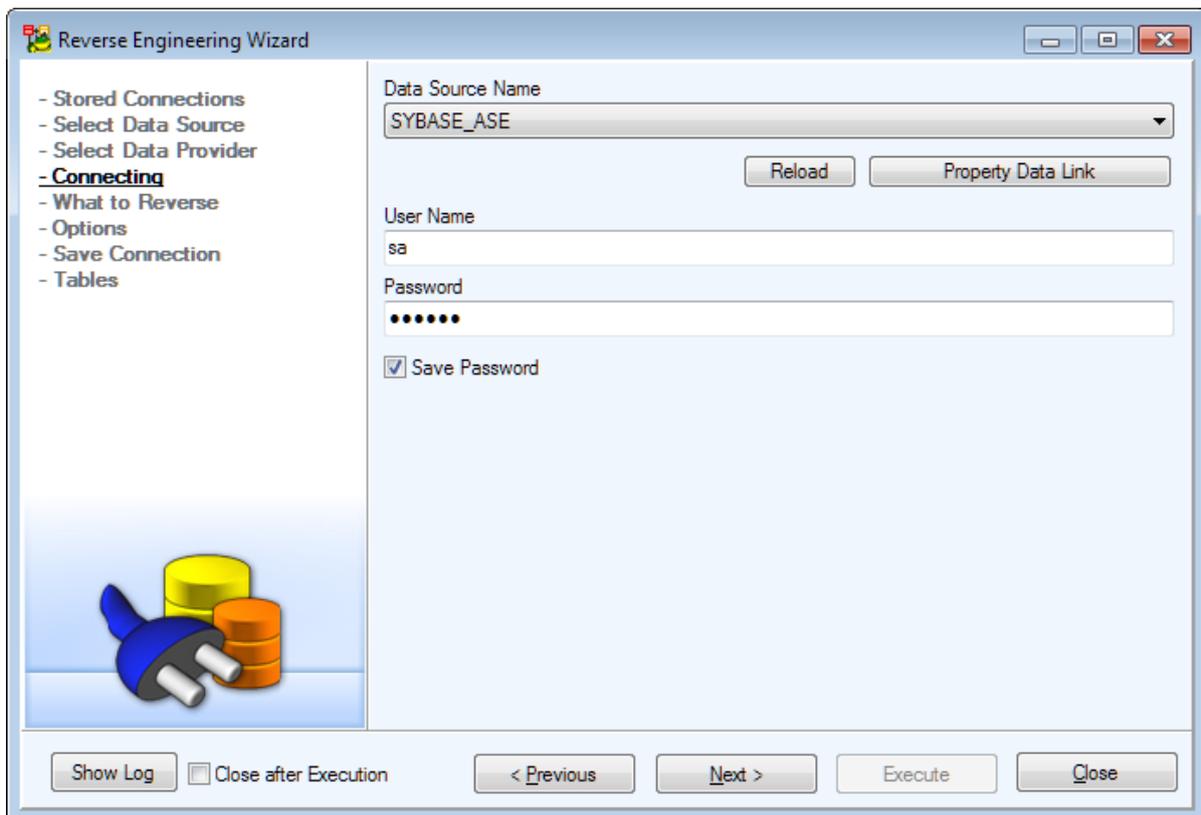
- Encryption Keys
- Segments
- Web Services

## Reverse Engineering - Sybase ASE 15.7

Available **Data Providers** are:

- **Connection via ODBC**
- **Connection via ADO**
- **Native Connection**

### Connection via ODBC

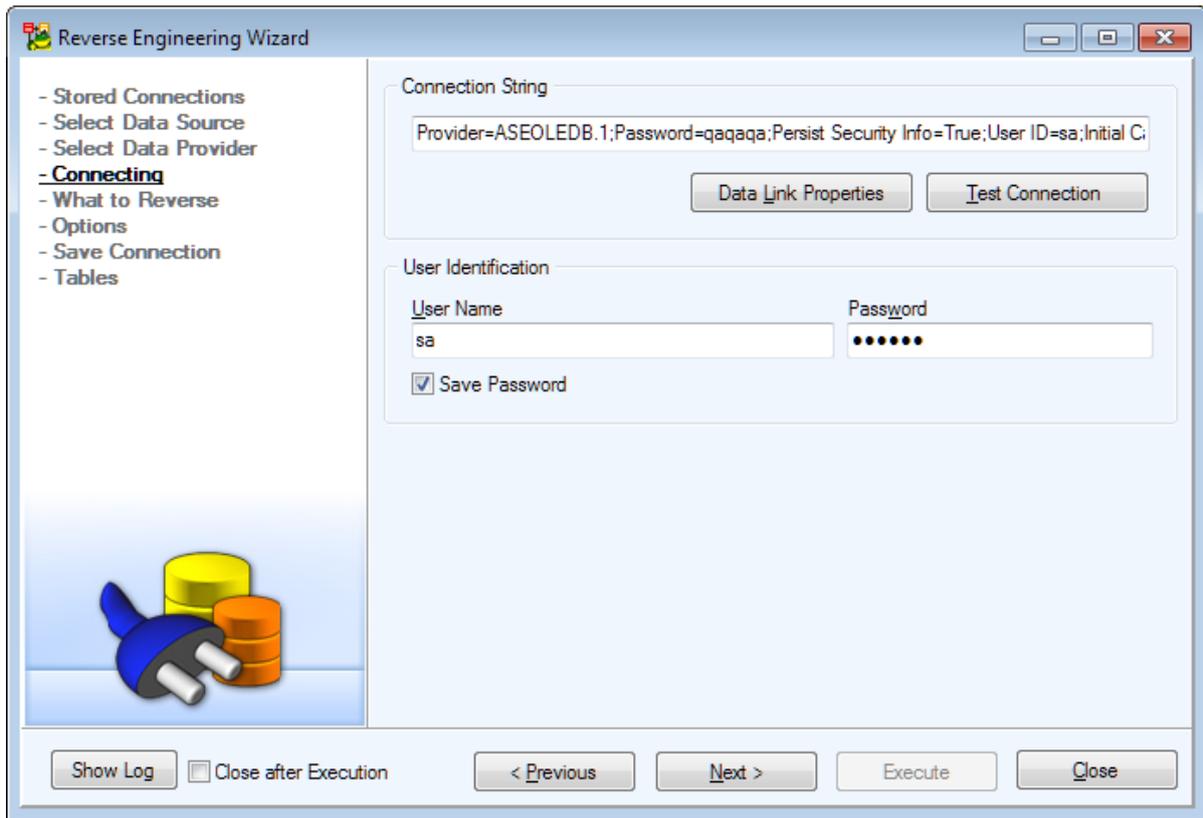


The screenshot shows the 'Reverse Engineering Wizard' dialog box. On the left, a tree view lists the steps: 'Stored Connections', 'Select Data Source', 'Select Data Provider', 'Connecting' (highlighted), 'What to Reverse', 'Options', 'Save Connection', and 'Tables'. Below the tree is an icon of a blue plug and three database cylinders (yellow, orange, and red). The main area contains the following fields and controls:

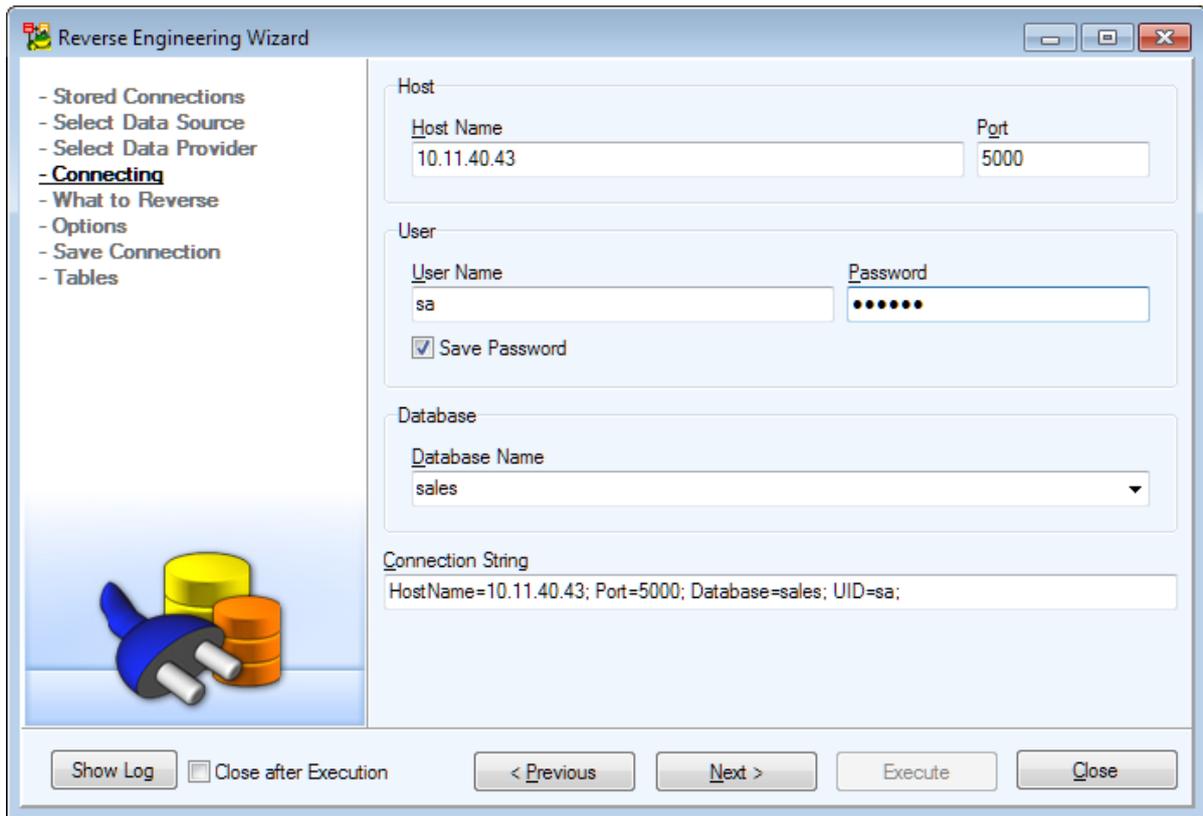
- Data Source Name:** A dropdown menu showing 'SYBASE\_ASE'. Below it are 'Reload' and 'Property Data Link' buttons.
- User Name:** A text box containing 'sa'.
- Password:** A text box with seven black dots.
- Save Password:** A checked checkbox.

At the bottom, there are buttons for 'Show Log', 'Close after Execution' (unchecked), '< Previous', 'Next >', 'Execute', and 'Close'.

### Connection via ADO



**Native Connection:**



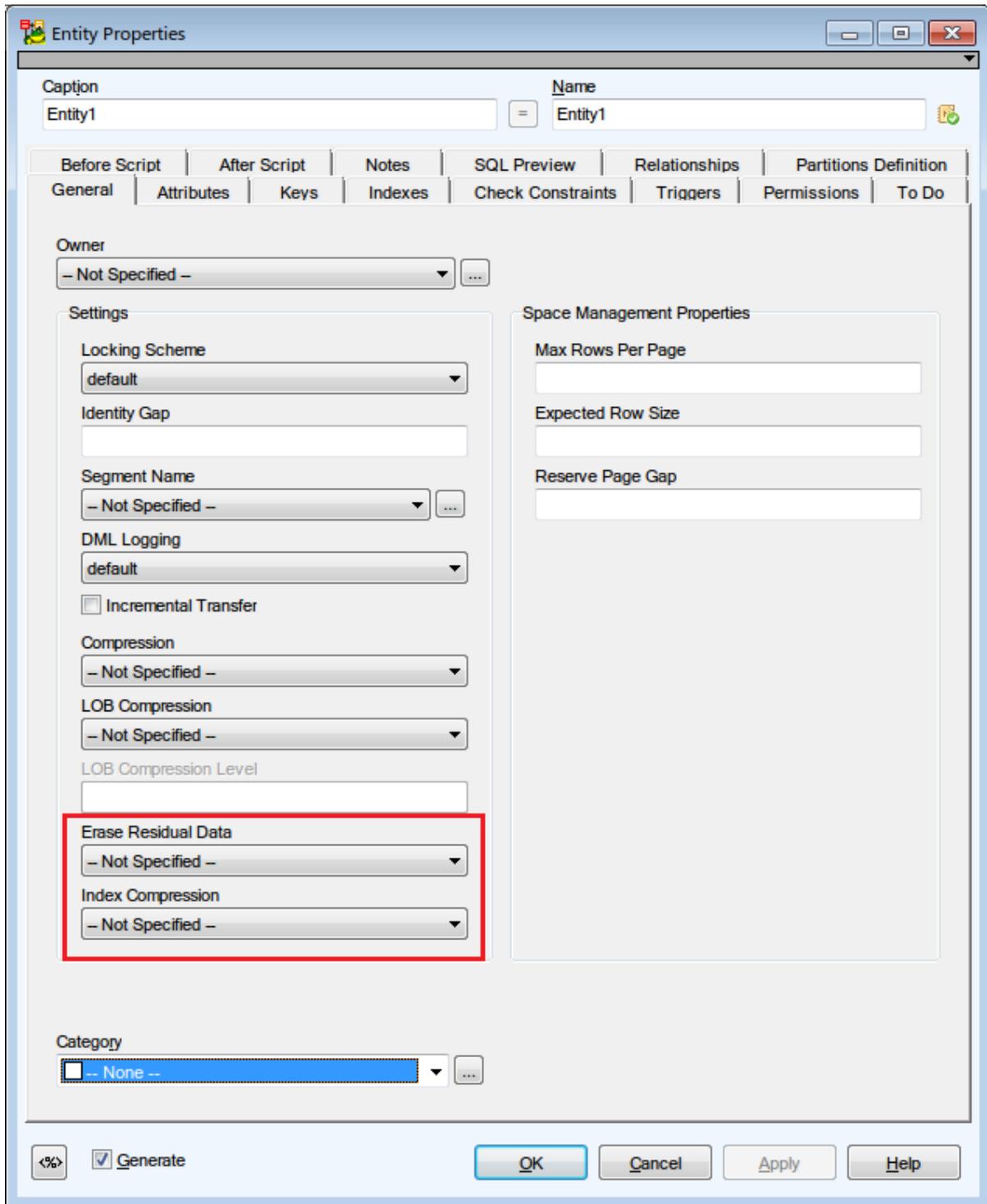
# Specifics - SAP ASE 16.0

New *OR REPLACE* statement for **Functions, Procedures, Triggers, Views, Defaults, Rules**

- *CREATE [OR REPLACE] objectType*

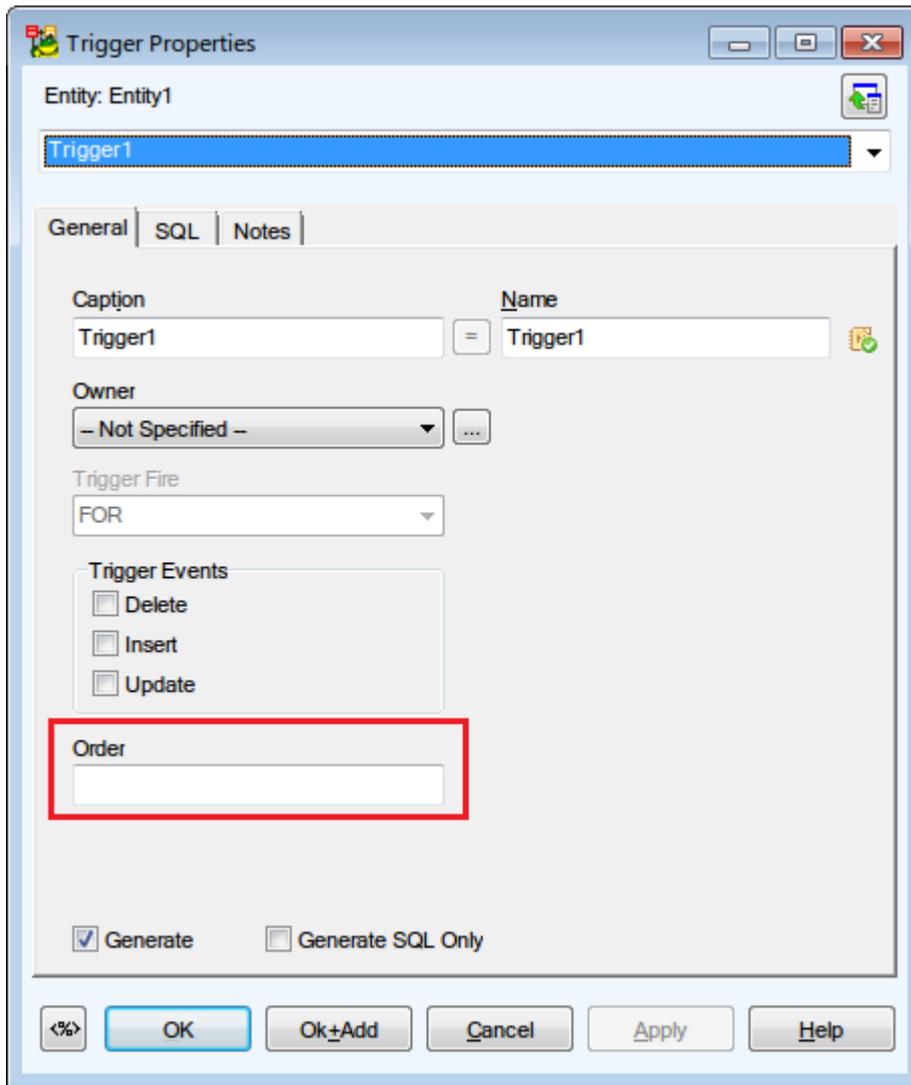
## Entities

*ERASE RESIDUAL DATA, INDEX COMPRESSION* parameters



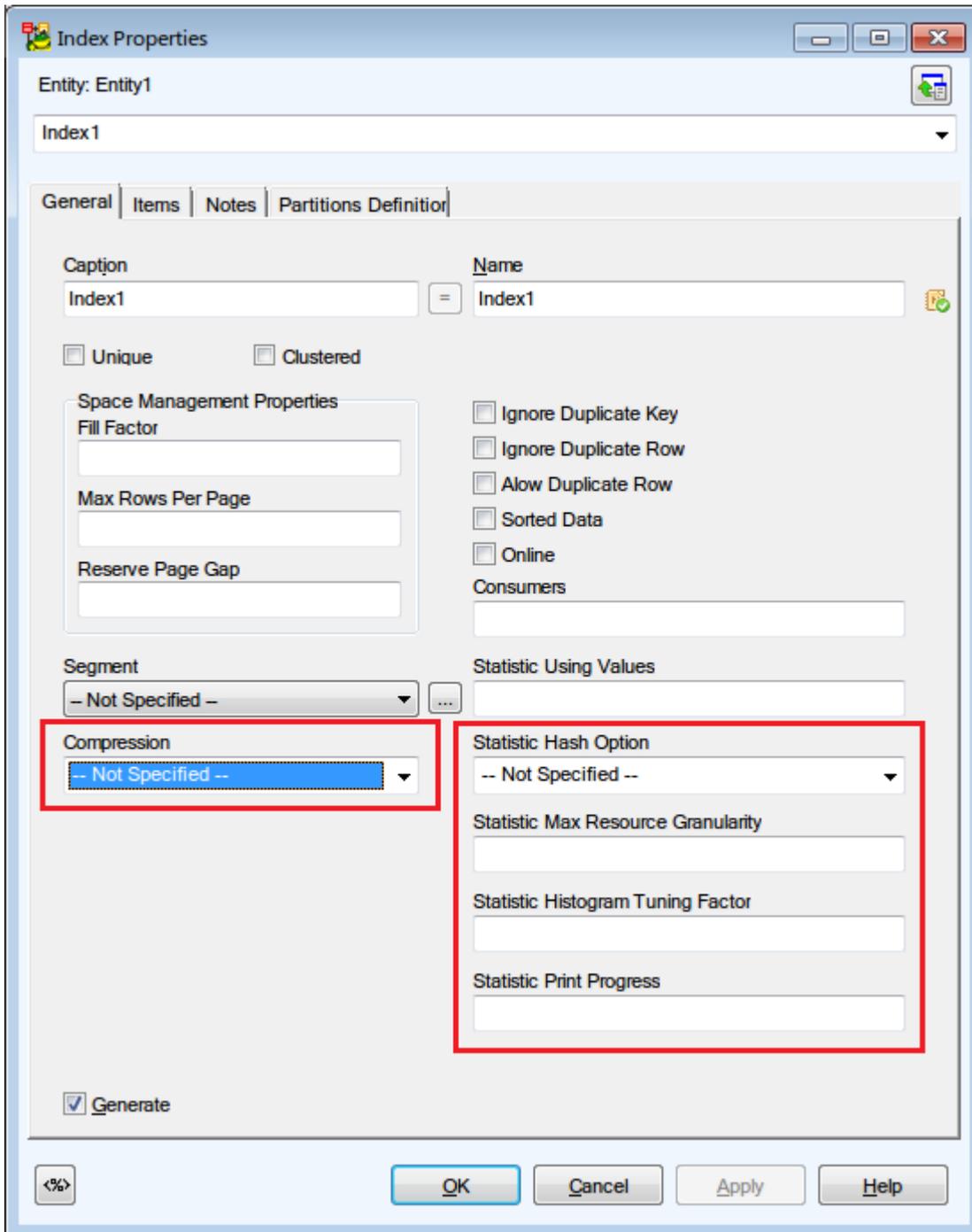
## Triggers

*ORDER* parameter for **Entity Triggers**



## Indexes

*STATISTICS HASH OPTIONS, STATISTICS MAX RESOURCE GRANULARITY, STATISTICS HISTOGRAM TUNING FACTOR, STATISTICS PRINT PROGRESS, WITH INDEX COMPRESSION* parameters



## Procedures

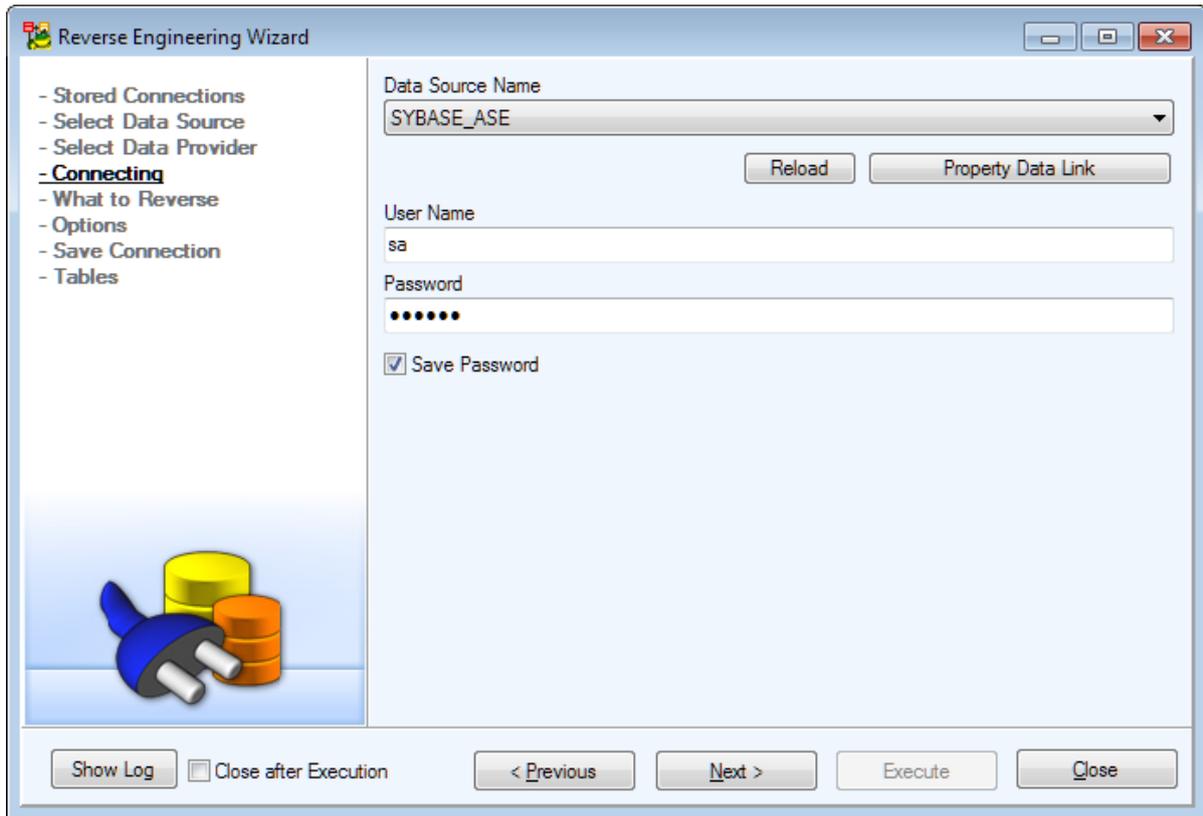
*WITH RECOMPILE, EXECUTE AS* parameters

# Reverse Engineering - SAP ASE 16.0

Available **Data Providers** are:

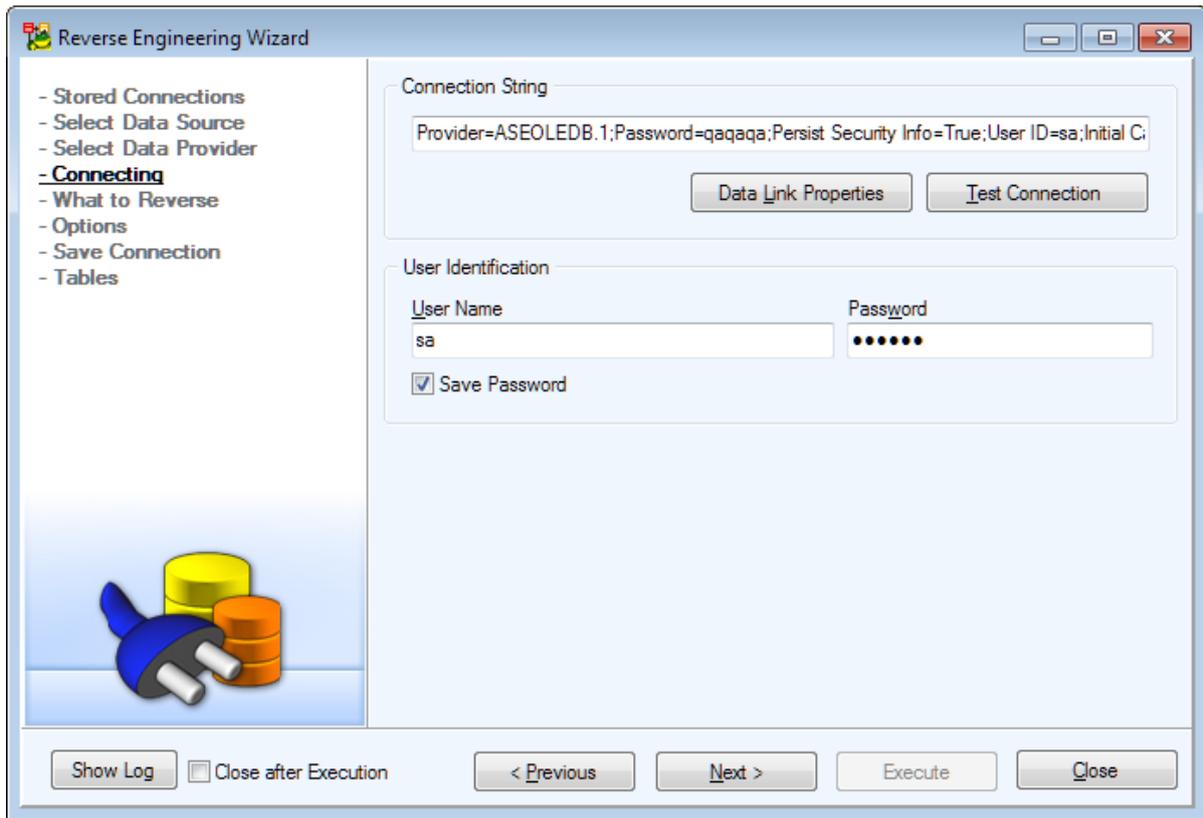
- **Connection via ODBC**
- **Connection via ADO**
- **Native Connection**

## Connection via ODBC

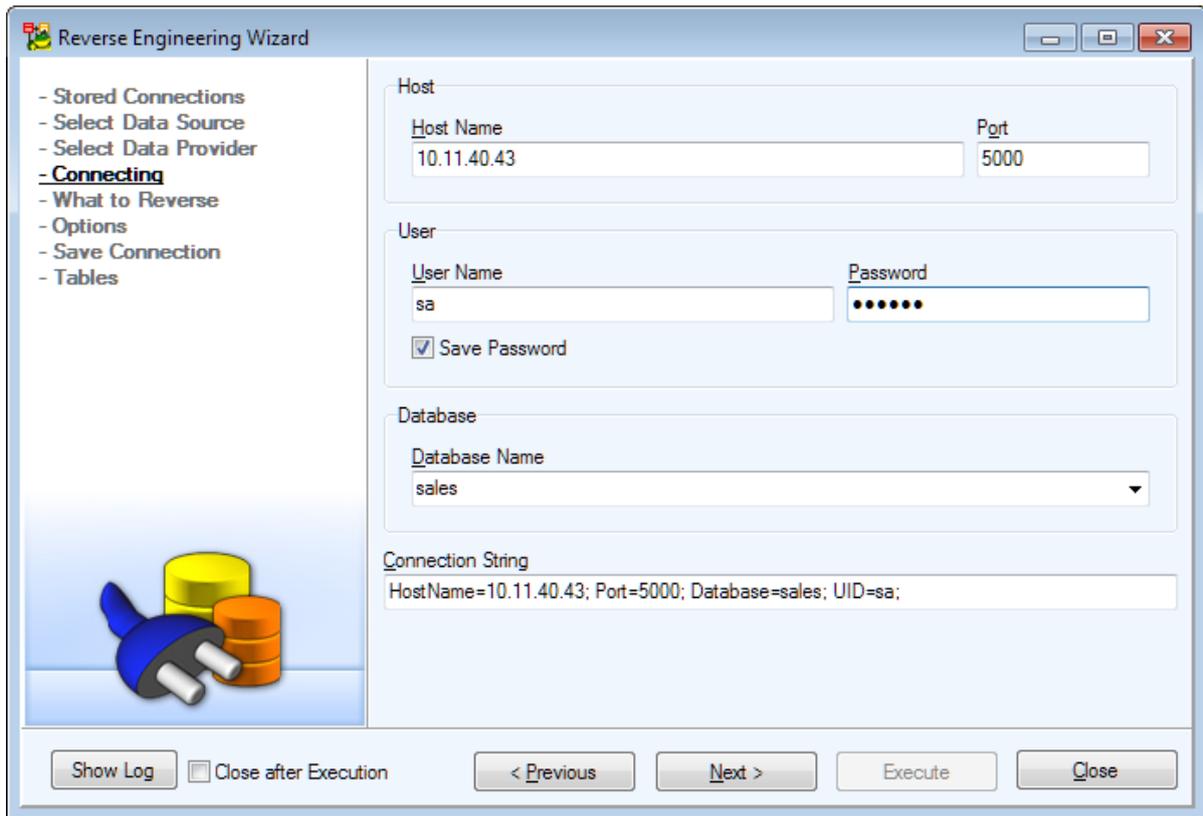


The screenshot shows the 'Reverse Engineering Wizard' dialog box. The left sidebar contains a list of steps: '- Stored Connections', '- Select Data Source', '- Select Data Provider', '- **Connecting**', '- What to Reverse', '- Options', '- Save Connection', and '- Tables'. The main area is titled 'Data Source Name' and has a dropdown menu set to 'SYBASE\_ASE'. Below this are two buttons: 'Reload' and 'Property Data Link'. The 'User Name' field contains 'sa'. The 'Password' field is masked with dots. There is a checked checkbox for 'Save Password'. At the bottom, there are buttons for 'Show Log', 'Close after Execution' (unchecked), '< Previous', 'Next >', 'Execute', and 'Close'. An icon of a blue plug and orange cylinders is visible in the bottom left corner of the dialog.

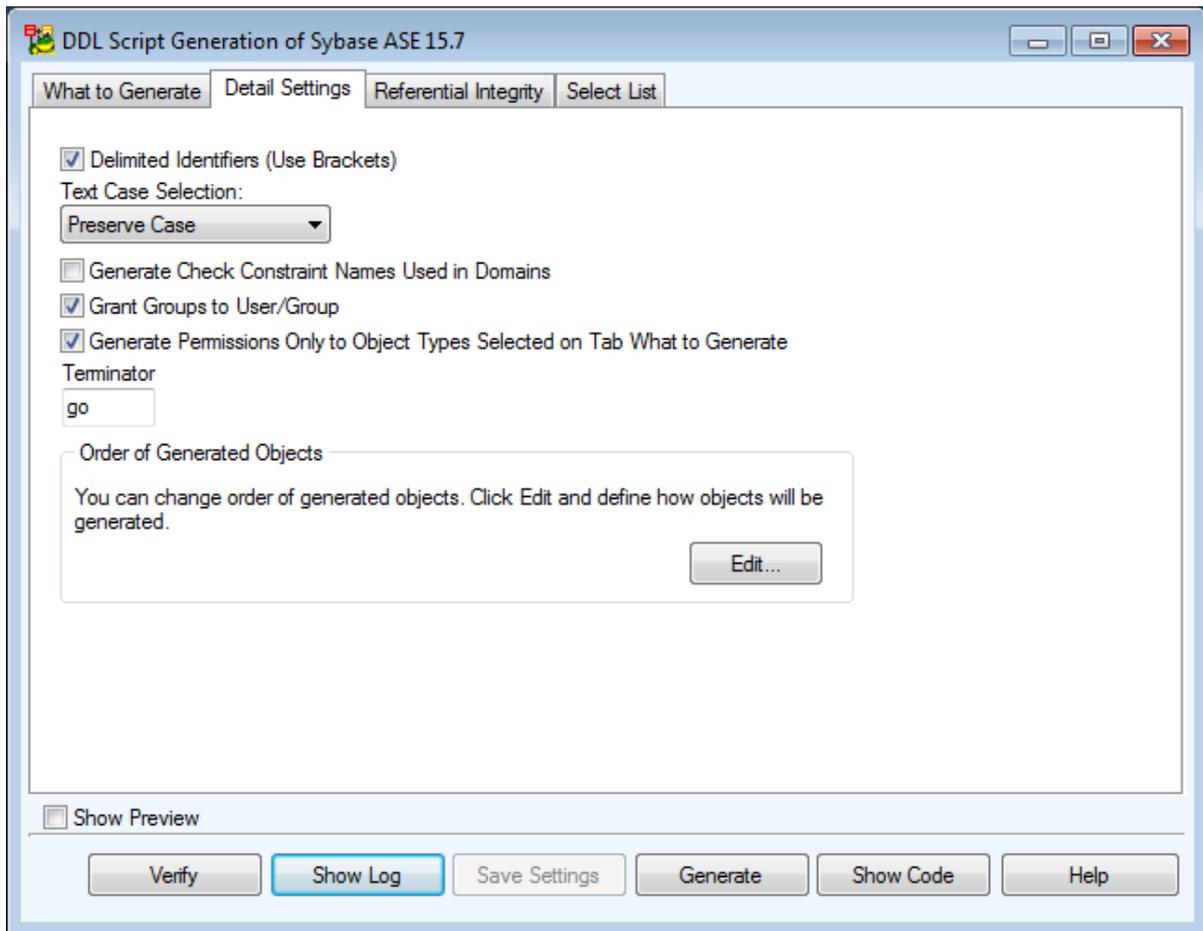
## Connection via ADO



**Native Connection:**

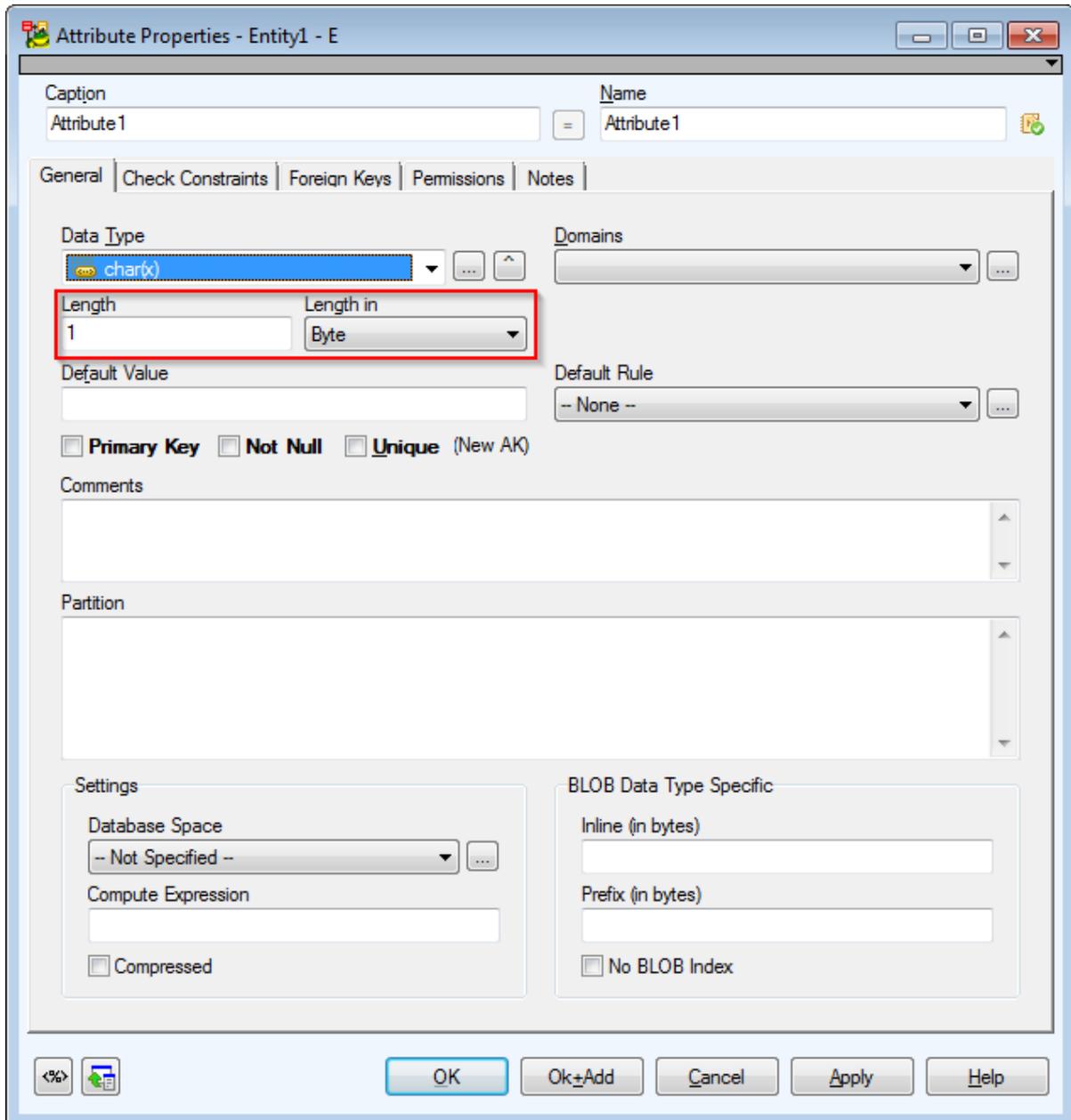


# Script Generation - SAP ASE 16.0



# Specifics - Sybase IQ 15.2

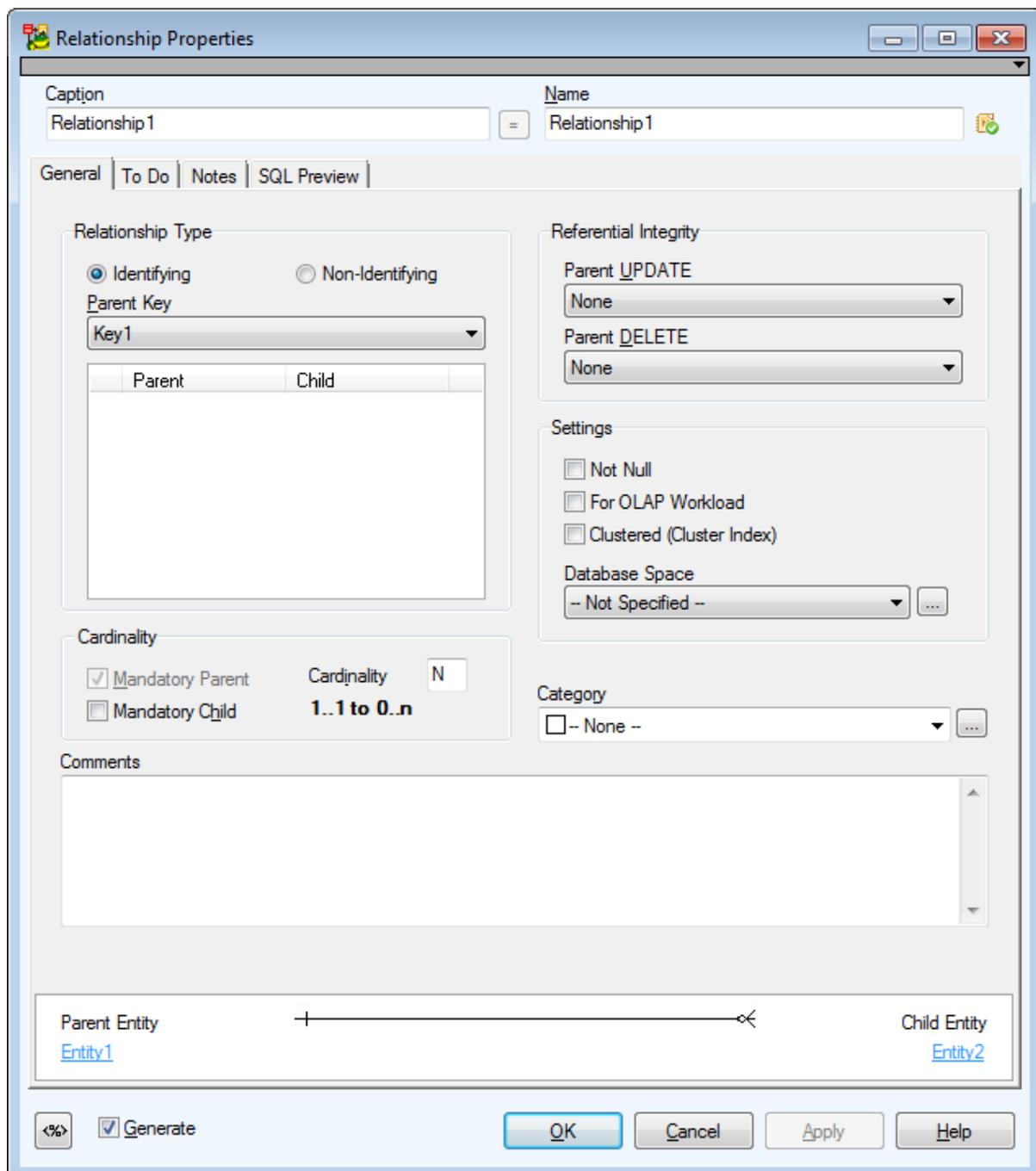
## Attribute



Data types marked with '\*' are system domains.

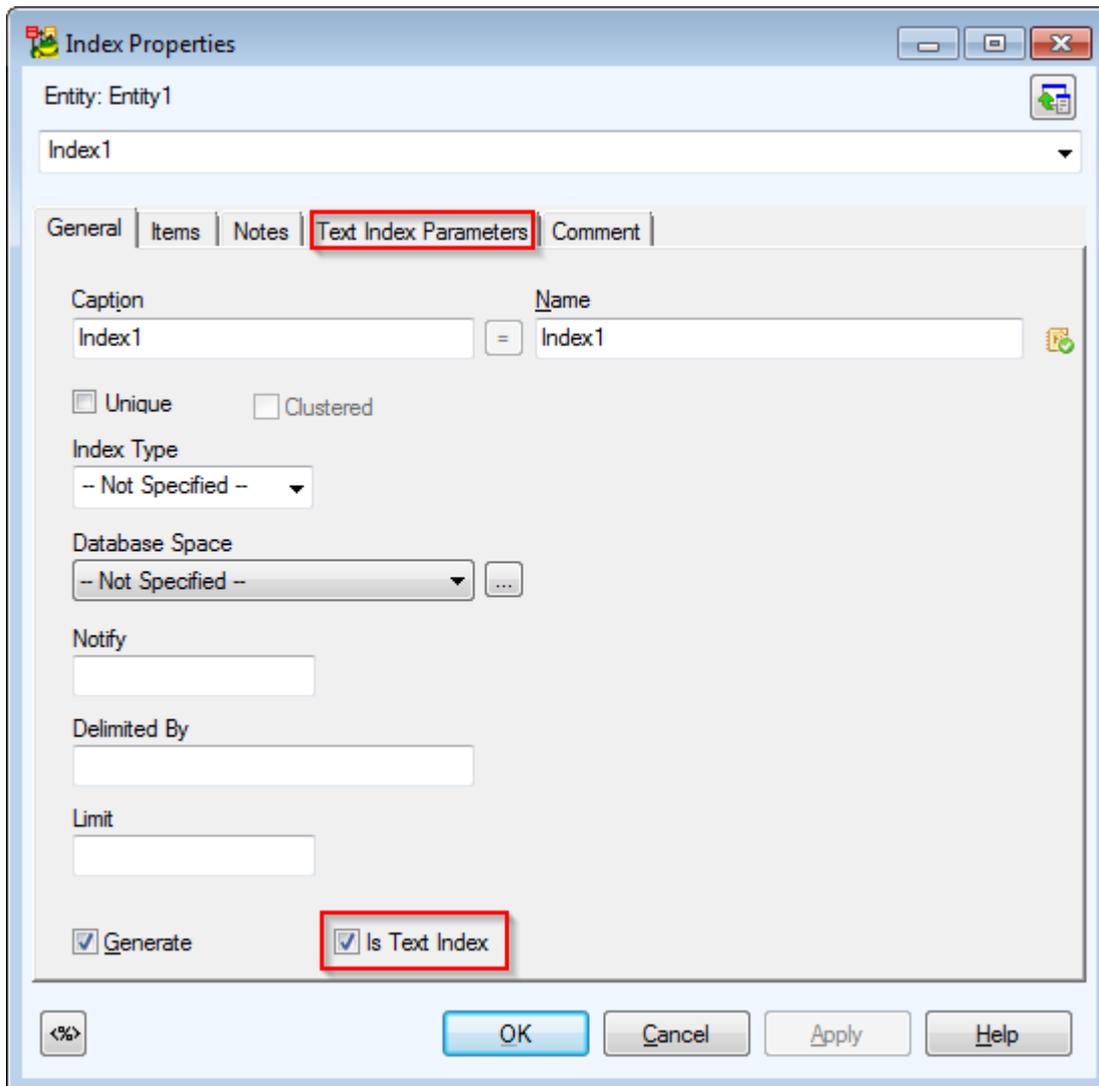
Second parameter BYTE (byte-length semantics [default]) or CHAR (character-length semantics) are available for data types CHAR and VARCHAR.

# Relationship

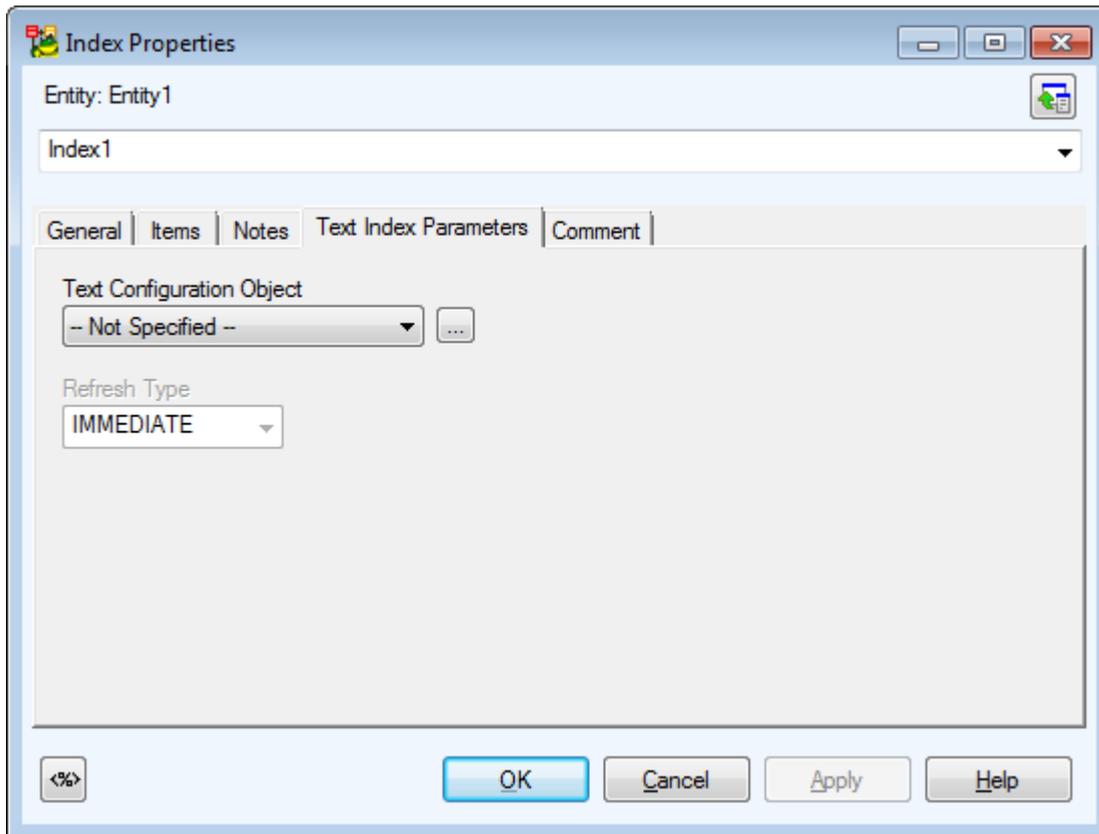


It is not possible to load FOR OLAP WORKLOAD statements during reverse engineering.

# Index



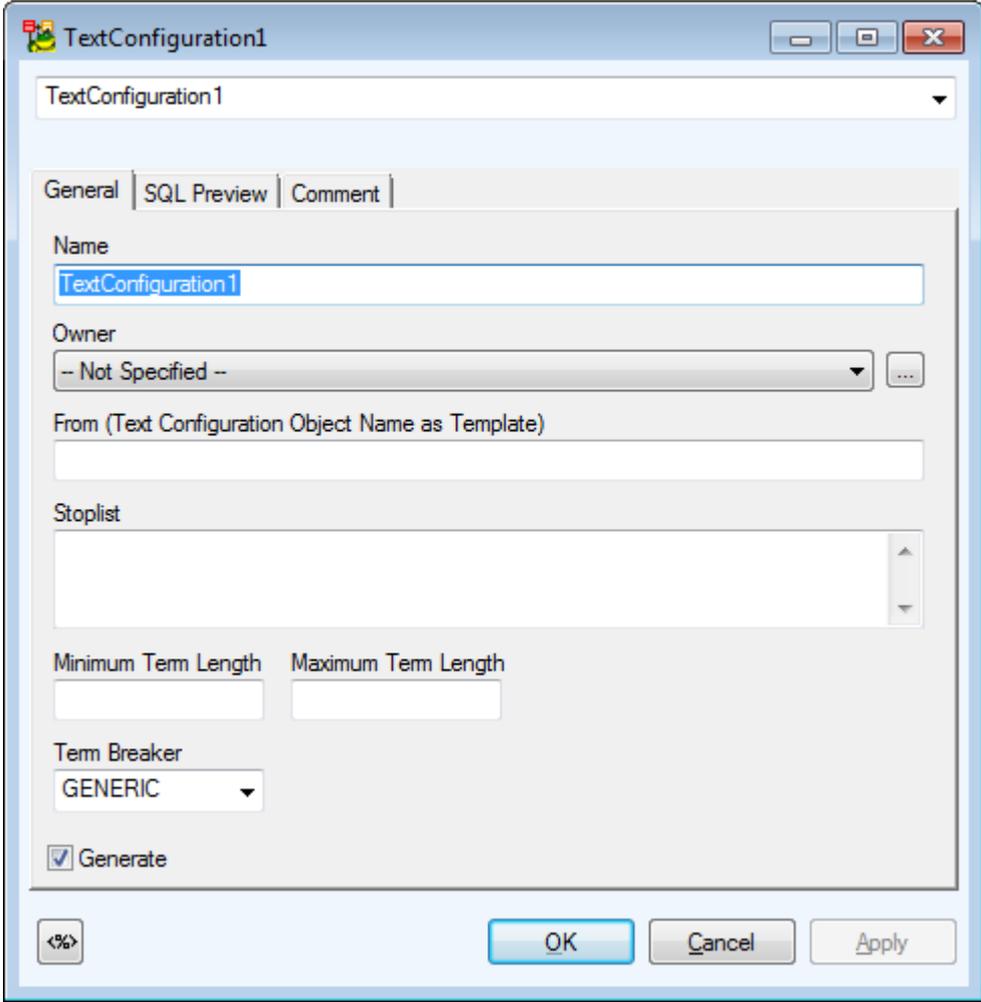
TEXT INDEX supported – select the **Is Text Index** checkbox and see the options on tab **Text Index Parameters**.



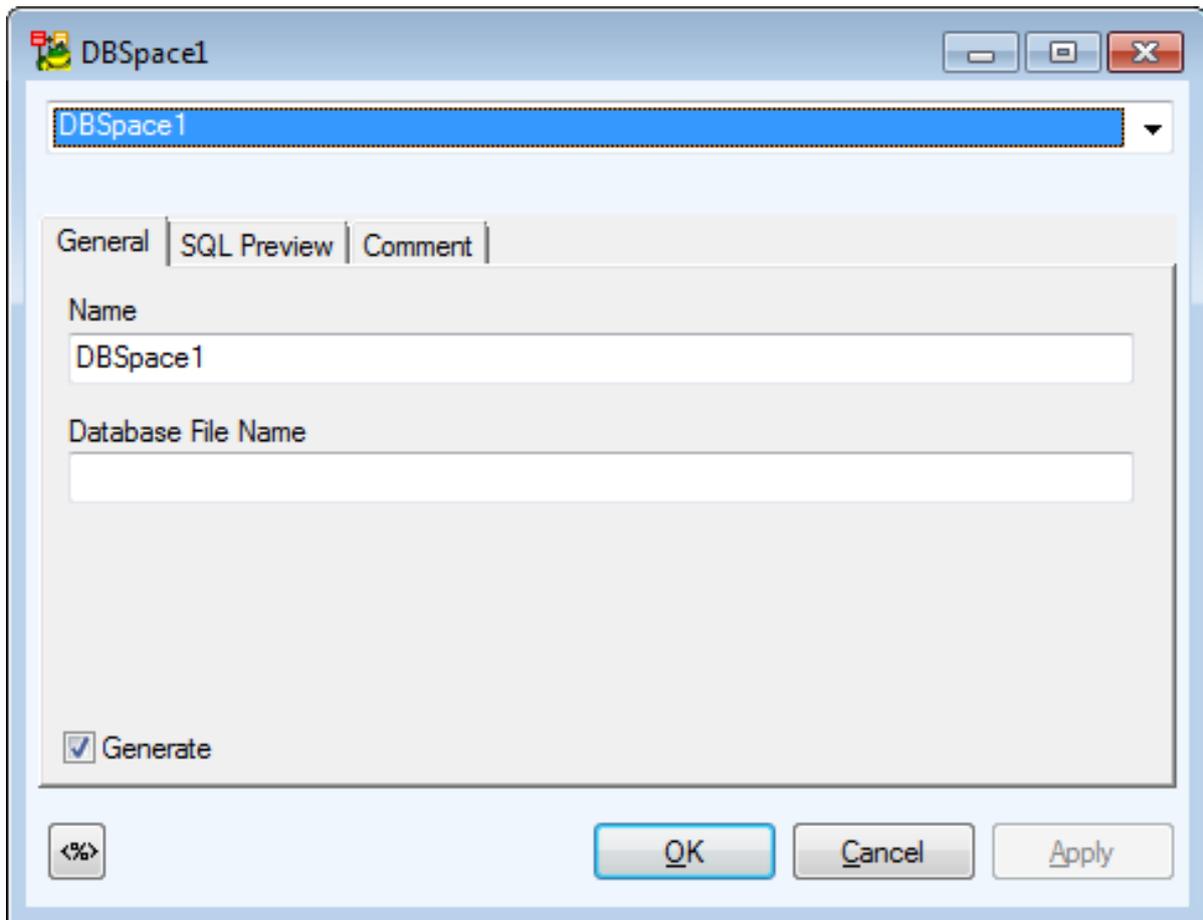
Other objects in Model Explorer:

- Database Spaces
- Text Configuration Objects
- User-Defined Messages

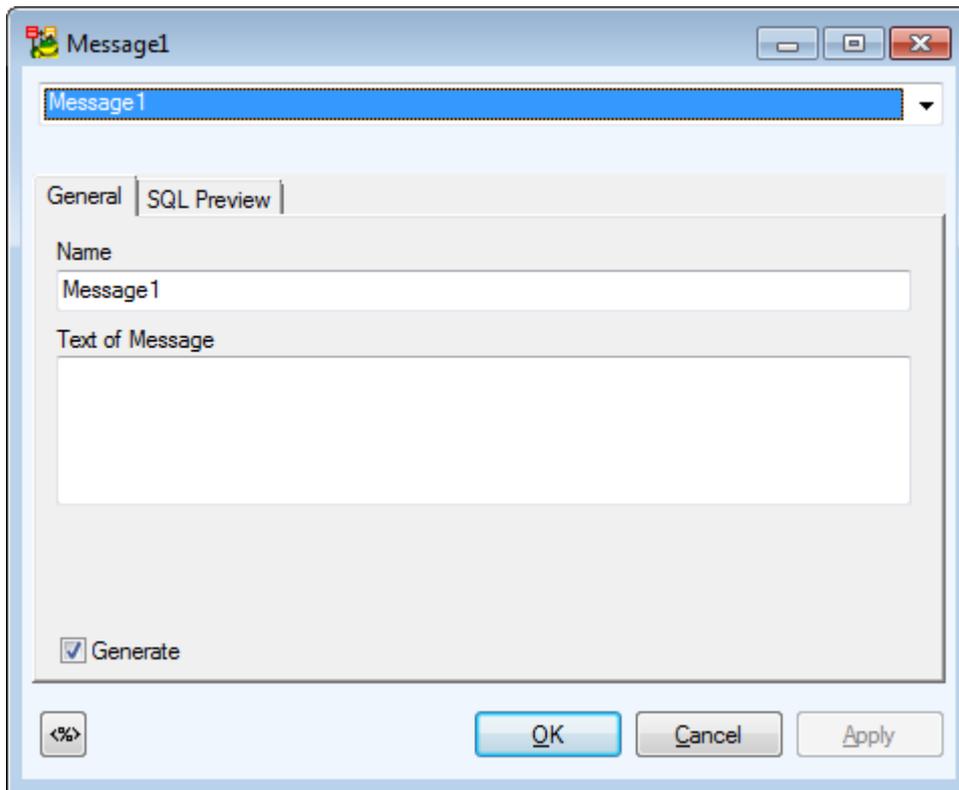
# Text Configuration Object



## Database Spaces (DBSPACE (51001))



## User-Defined Messages (MESSAGE (53201))

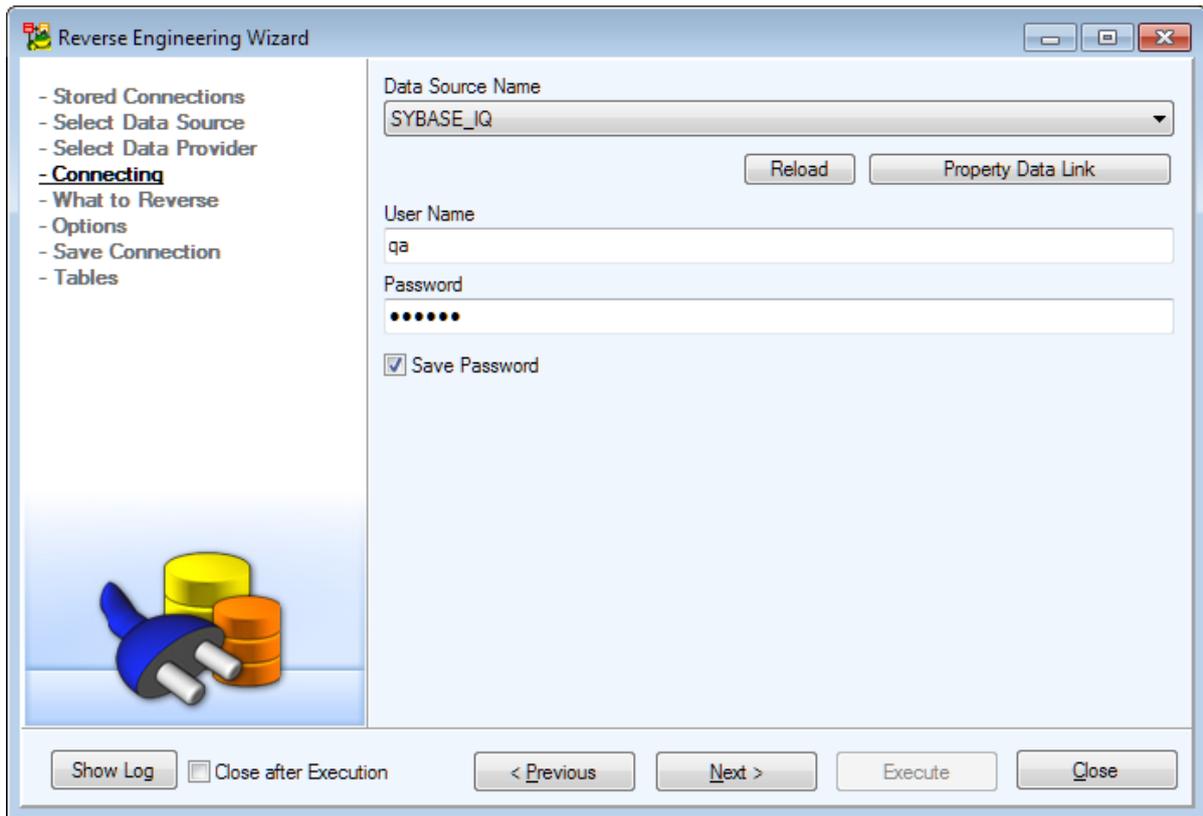


## Reverse Engineering - Sybase IQ 15.2

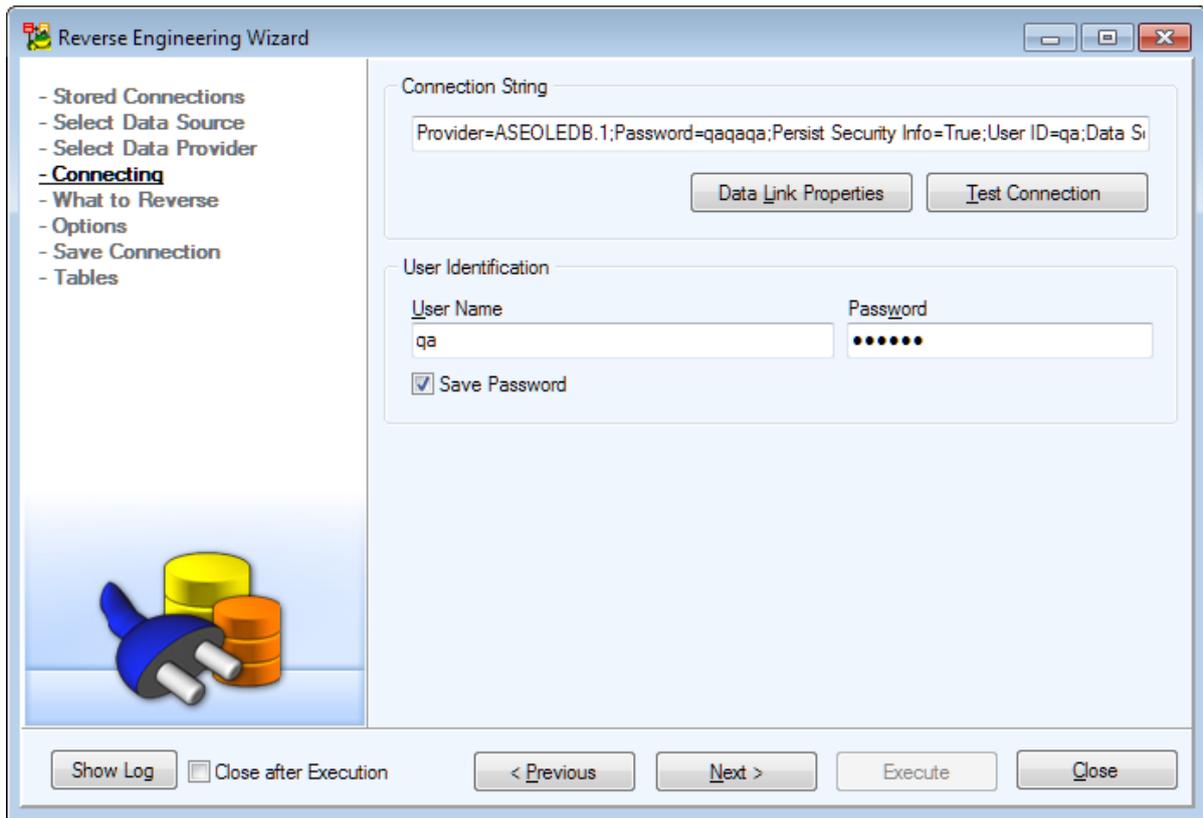
Available **Data Providers** are:

- Connection via ODBC
- Connection via ADO

**Connection via ODBC**

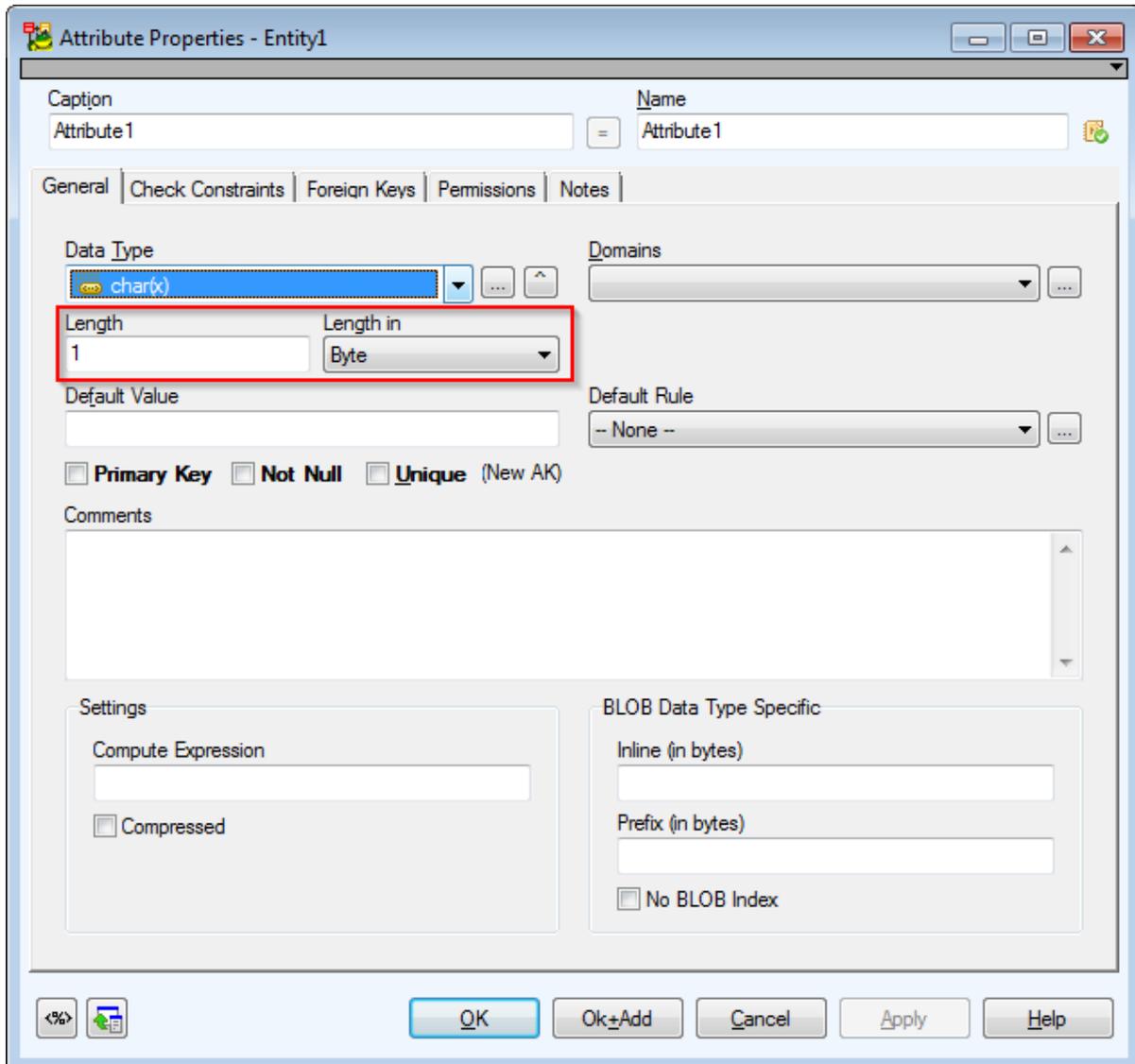


Connection via ADO



# Specifics - Sybase SQL Anywhere 11

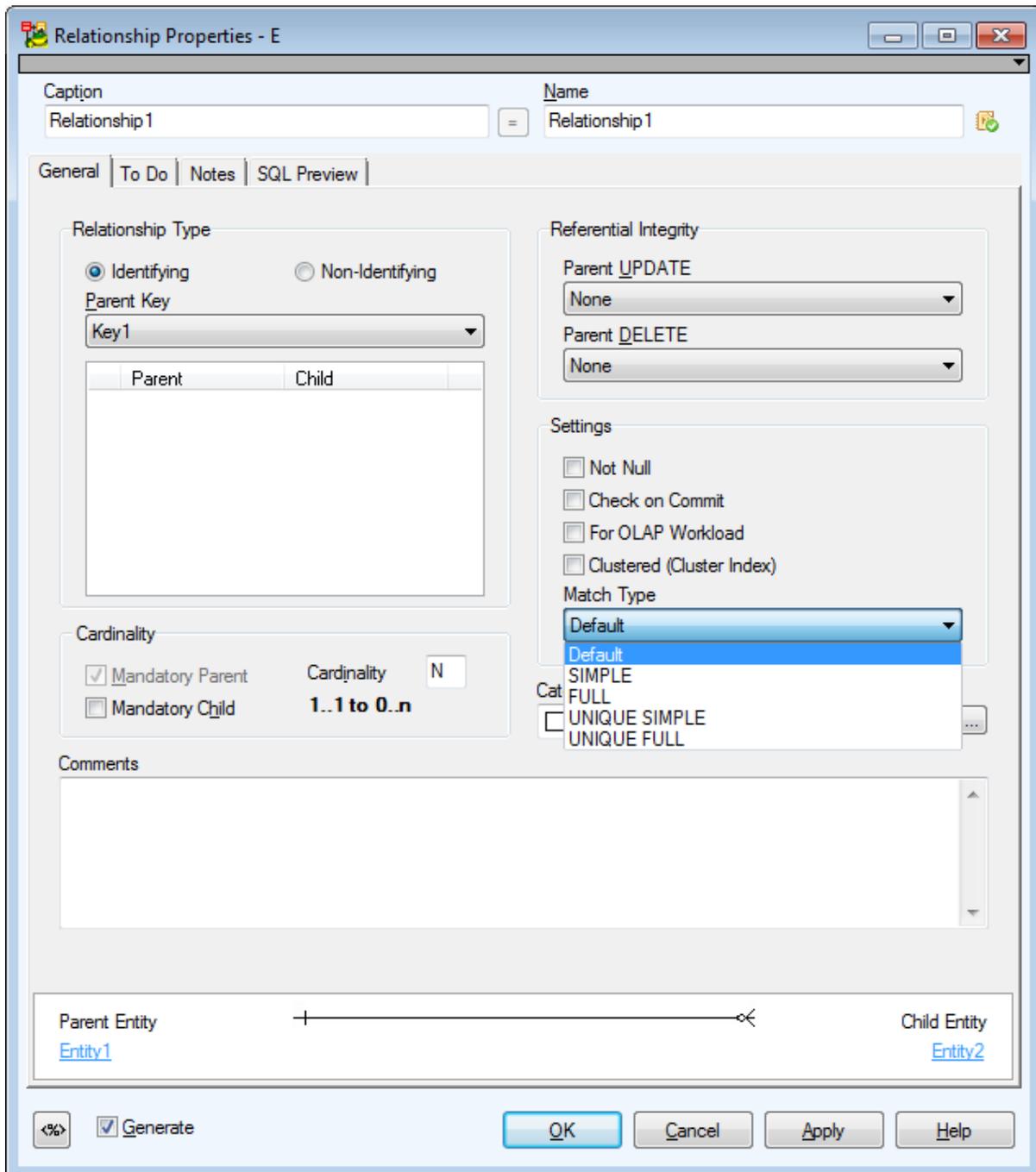
## Attribute



Data types marked with '\*' are system domains.

Second parameter BYTE (byte-length semantics [default]) or CHAR (character-length semantics) are available for data types CHAR and VARCHAR.

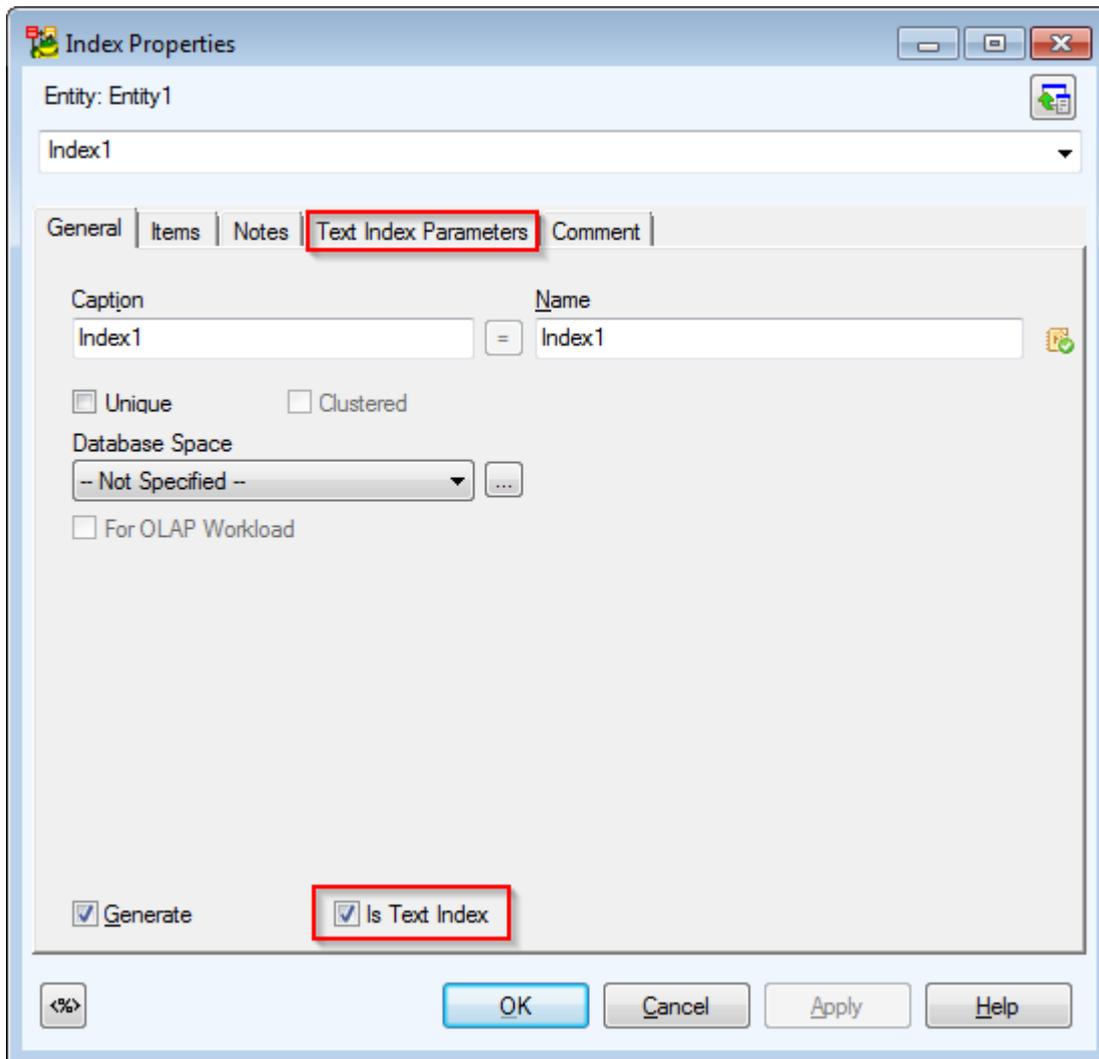
# Relationship



Box **Match Type** with options *Default*, *SIMPLE*, *FULL*, *UNIQUE SIMPLE*, *UNIQUE FULL*.

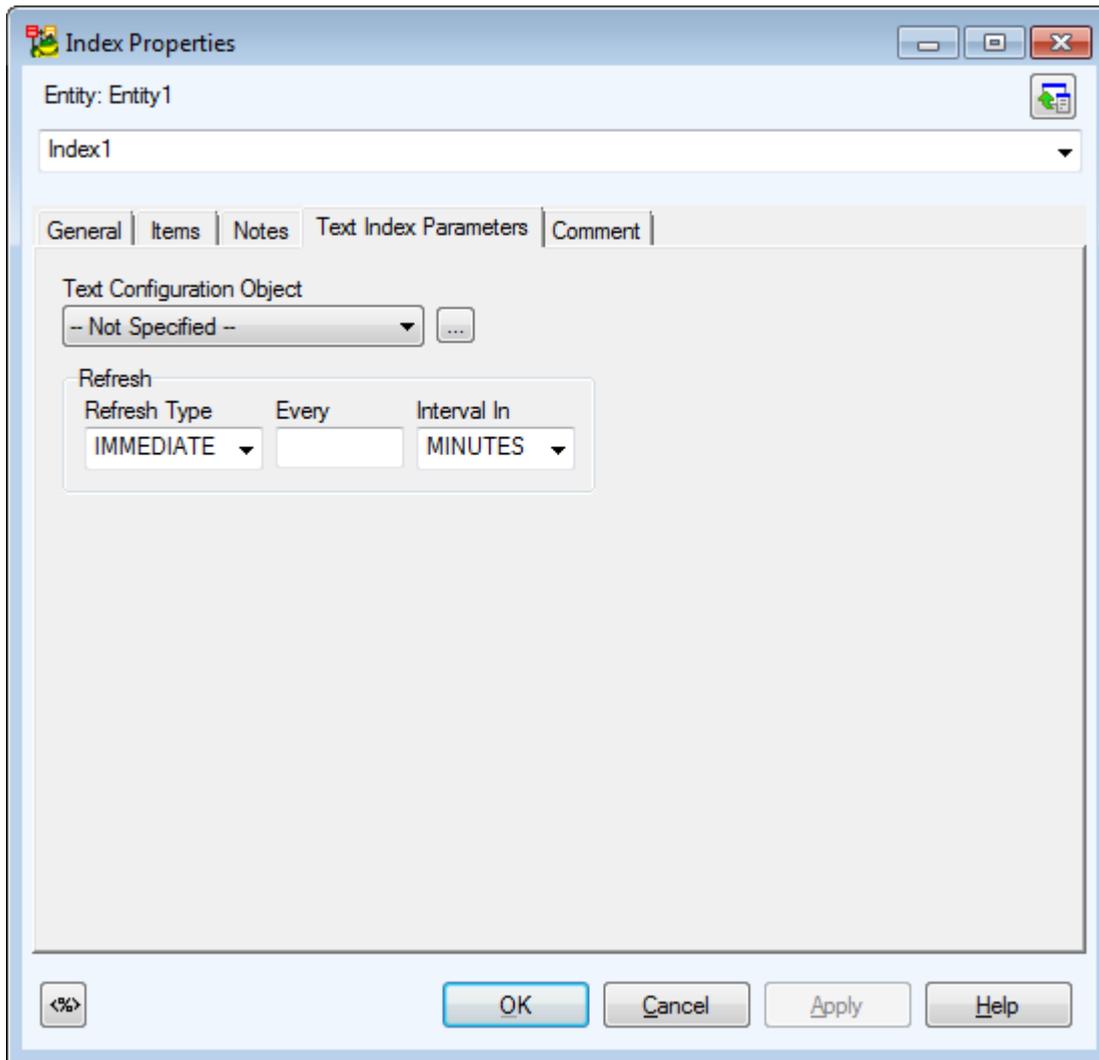
It is not possible to reload FOR OLAP WORKLOAD during reverse engineering.

# Index



It is not possible to reload FOR OLAP WORKLOAD during reverse engineering.

TEXT INDEX supported – select the **Is Text Index** checkbox and define other properties on tab **Text Index Parameters**.



Other objects in Model Explorer:

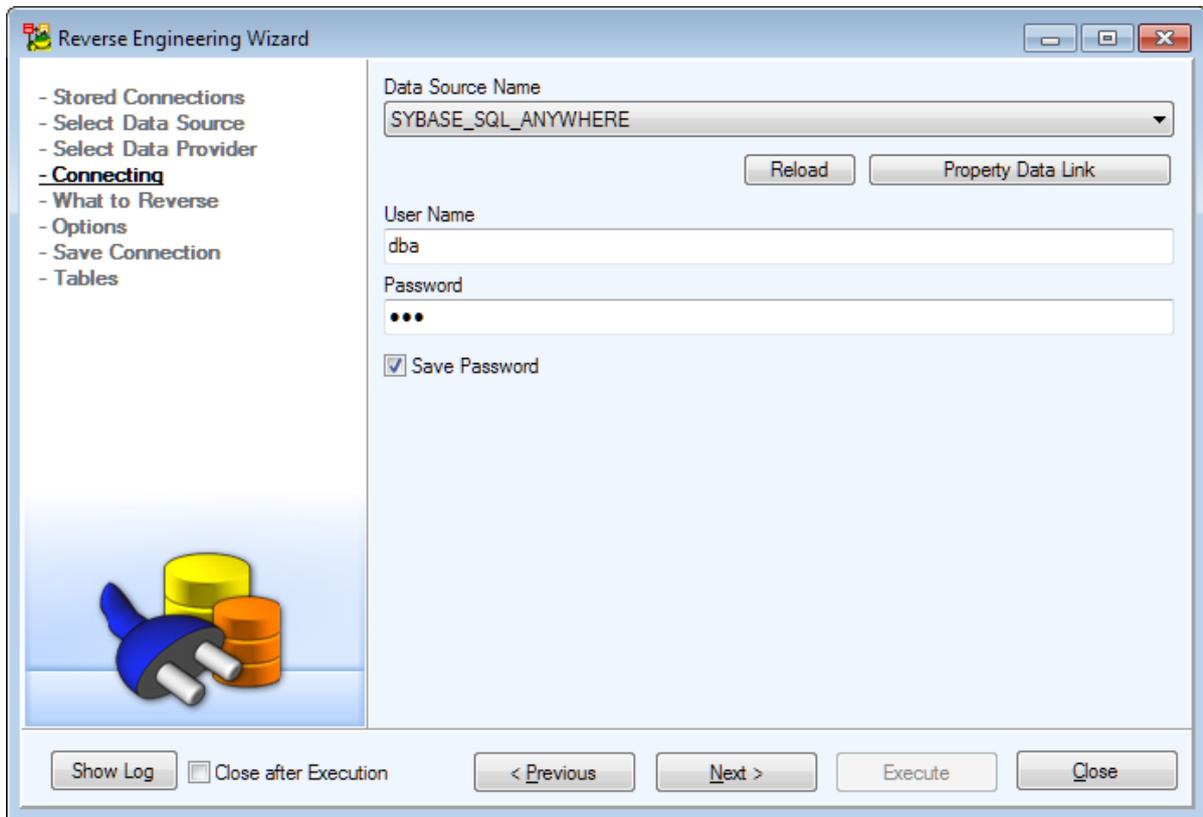
- Database Spaces
- Text Configuration Objects
- User-Defined Messages

## Reverse Engineering - Sybase SQL Anywhere 11

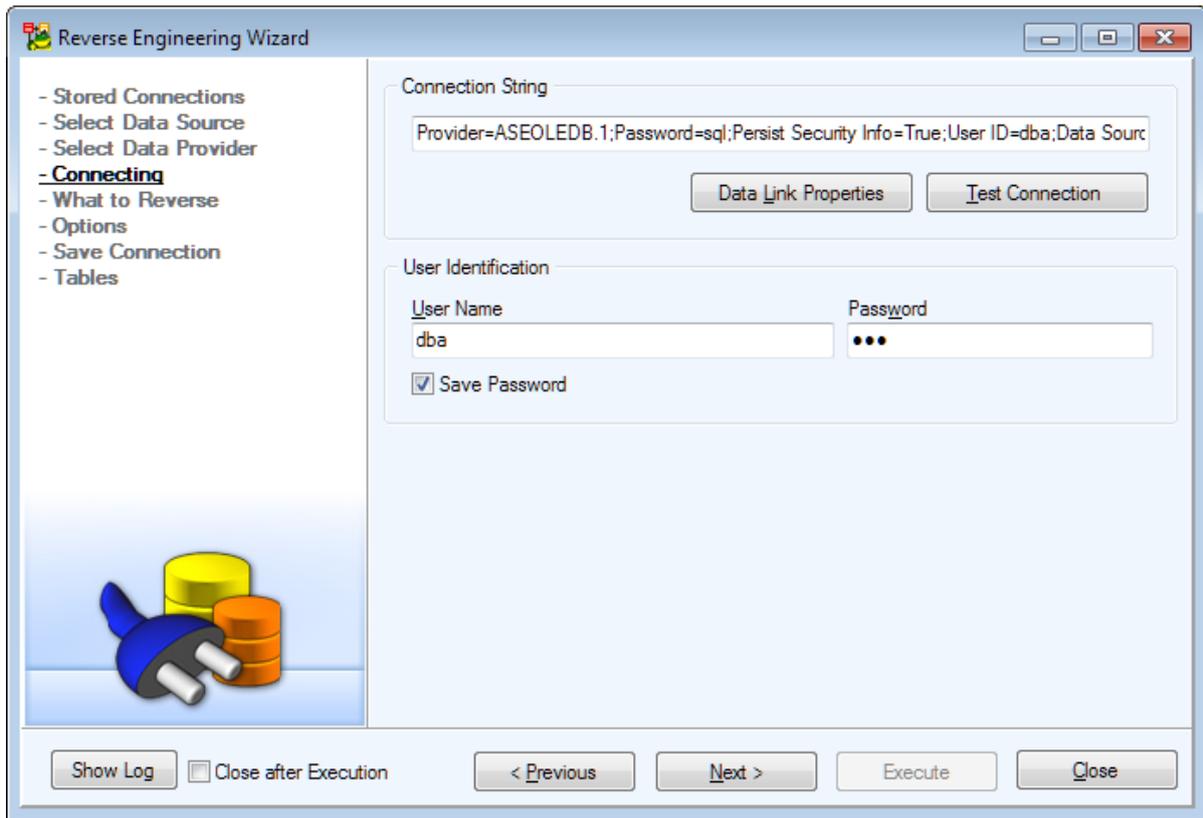
Available **Data Providers** are:

- Connection via ODBC
- Connection via ADO

### Connection via ODBC



### Connection via ADO



# Specifics - SAP SQL Anywhere 17

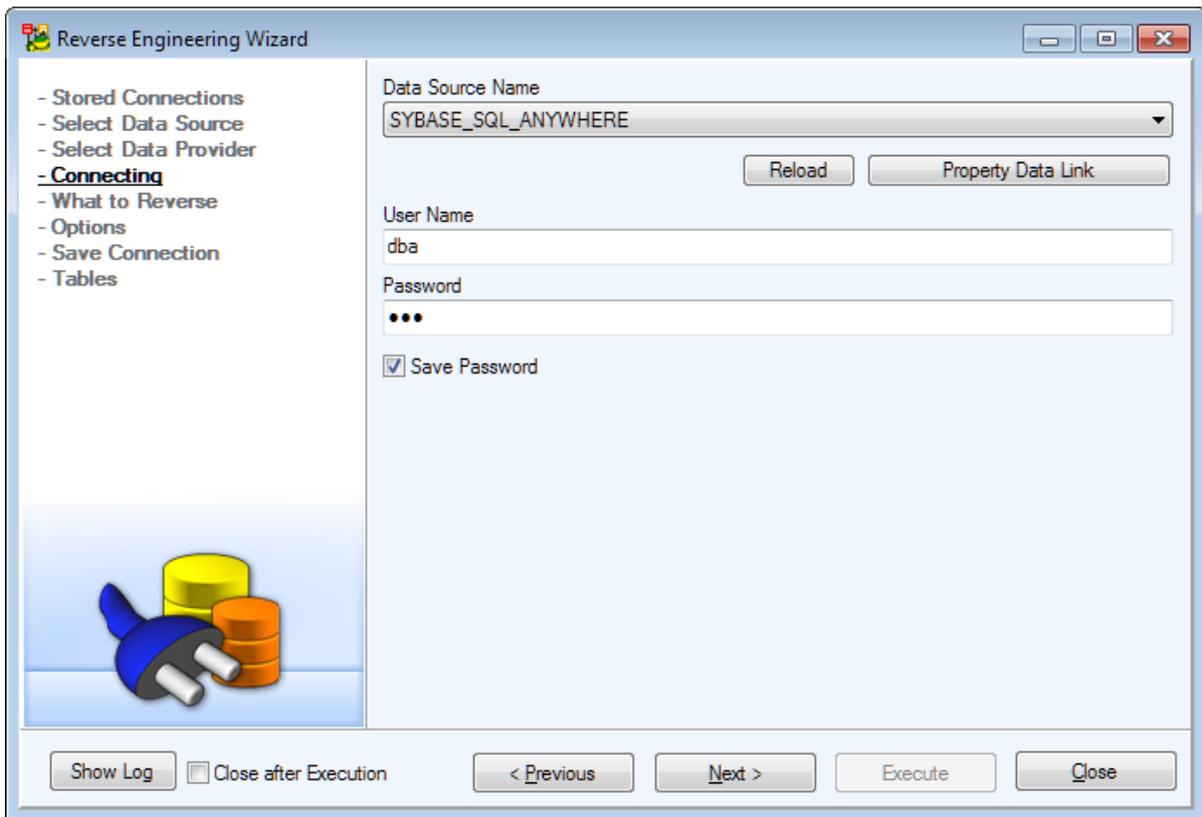
- Global **Options** are available in **Options | Model | Physical Model | SAP/Sybase SQL Anywhere | SAP SQL Anywhere 17**
- New checkbox **Generate IF EXISTS in DROP statements** in **DDL Script Generation | Detail Settings**
- Changes to DDL - CREATE DOMAIN, CREATE PROCEDURE, CREATE INDEX, CREATE TABLE, CREATE FUNCTION
- New object - SEQUENCE; available syntax CREATE SEQUENCE and DROP SEQUENCE
- New datatypes - TIMESTAMP WITH TIME ZONE, DATETIMEOFFSET and spatial datatypes

## Reverse Engineering - SAP SQL Anywhere 17

Available **Data Providers** are:

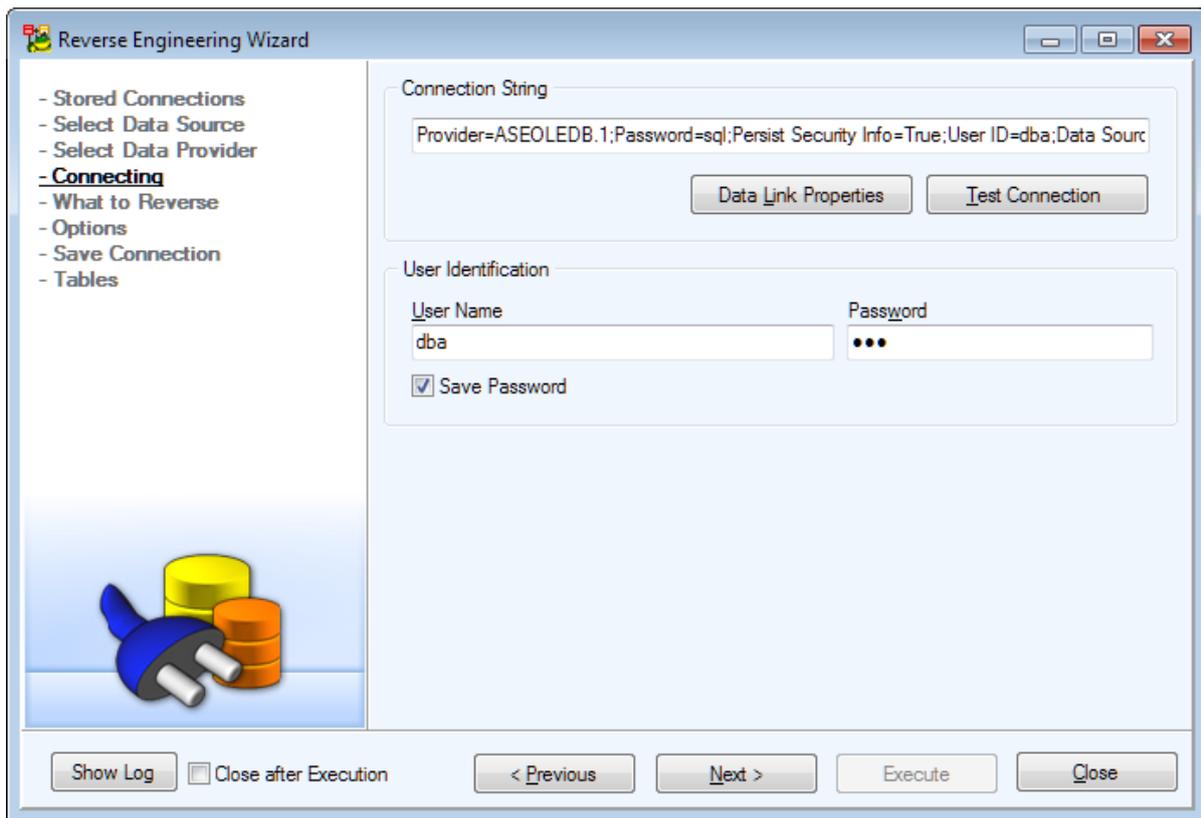
- **Connection via ODBC**
- **Connection via ADO**

### Connection via ODBC

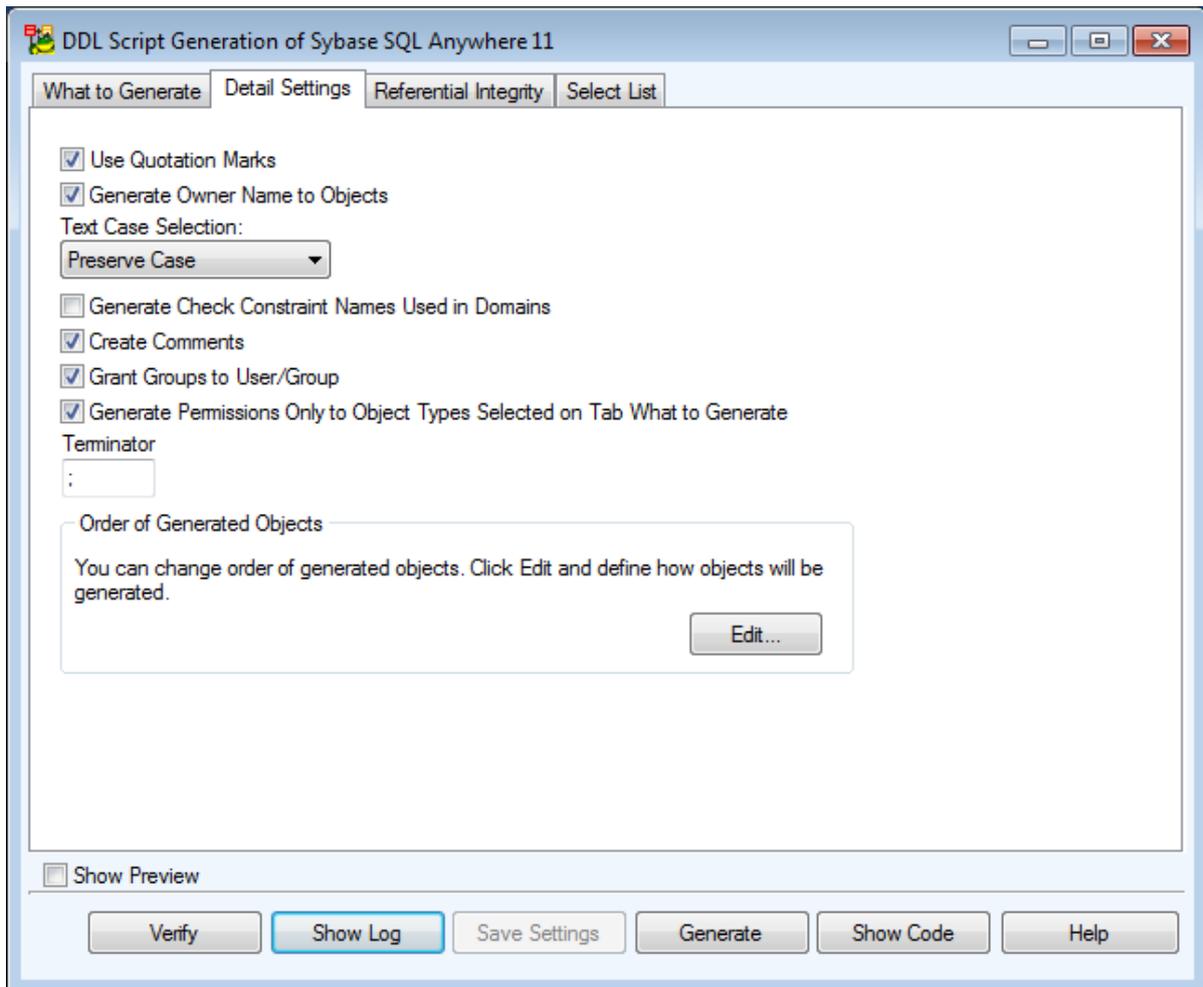


The screenshot shows the 'Reverse Engineering Wizard' dialog box. On the left is a tree view with the following items: - Stored Connections, - Select Data Source, - Select Data Provider, - **Connecting** (highlighted), - What to Reverse, - Options, - Save Connection, - Tables. Below the tree is an icon of a blue network plug and two orange database cylinders. The main area contains the following fields and controls: 'Data Source Name' dropdown menu with 'SYBASE\_SQL\_ANYWHERE' selected; 'Reload' and 'Property Data Link' buttons; 'User Name' text box with 'dba'; 'Password' text box with masked characters; and a checked 'Save Password' checkbox. At the bottom are buttons for 'Show Log', 'Close after Execution' (checkbox), '< Previous', 'Next >', 'Execute', and 'Close'.

### Connection via ADO

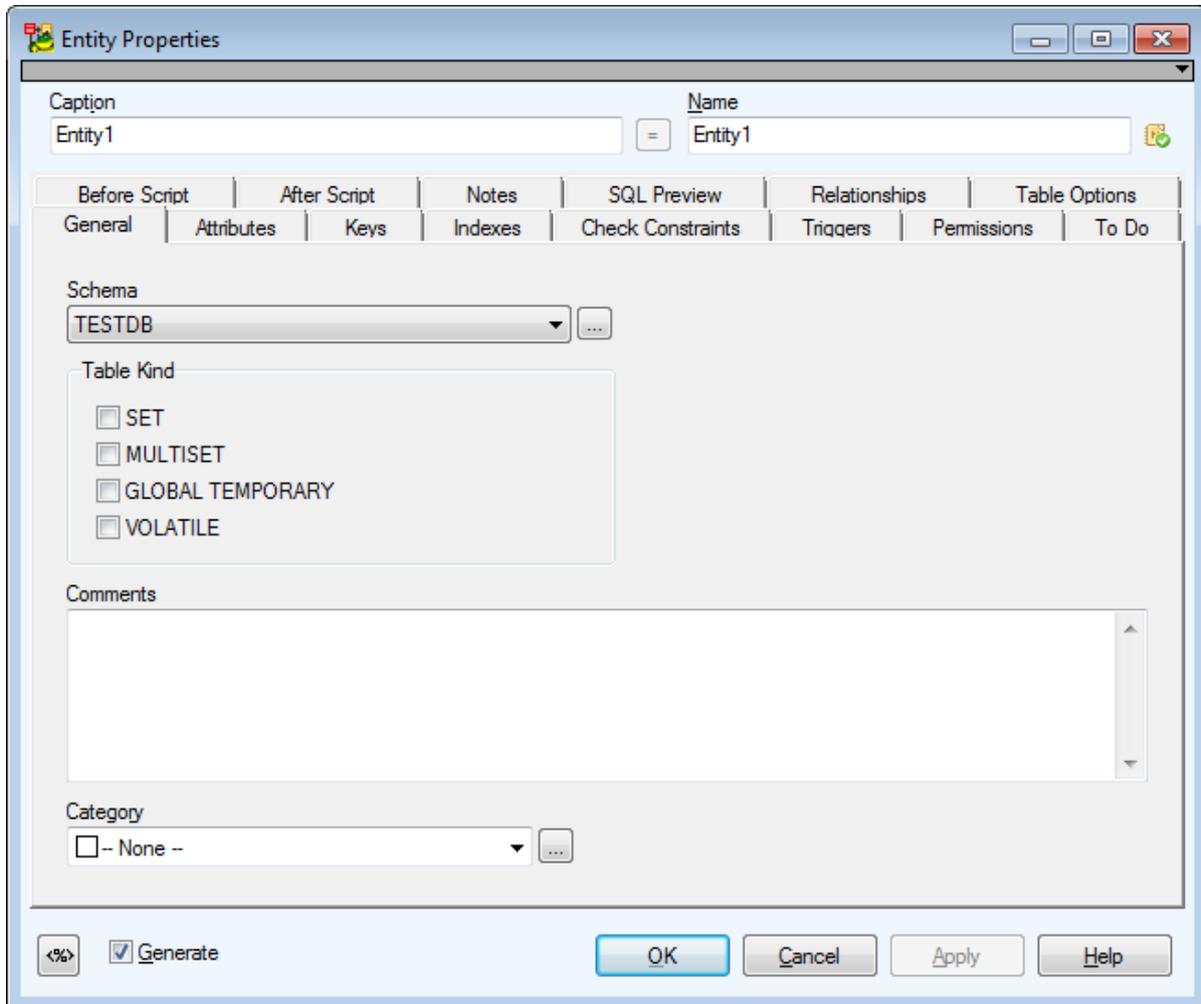


# Script Generation - Sybase SQL Anywhere 11

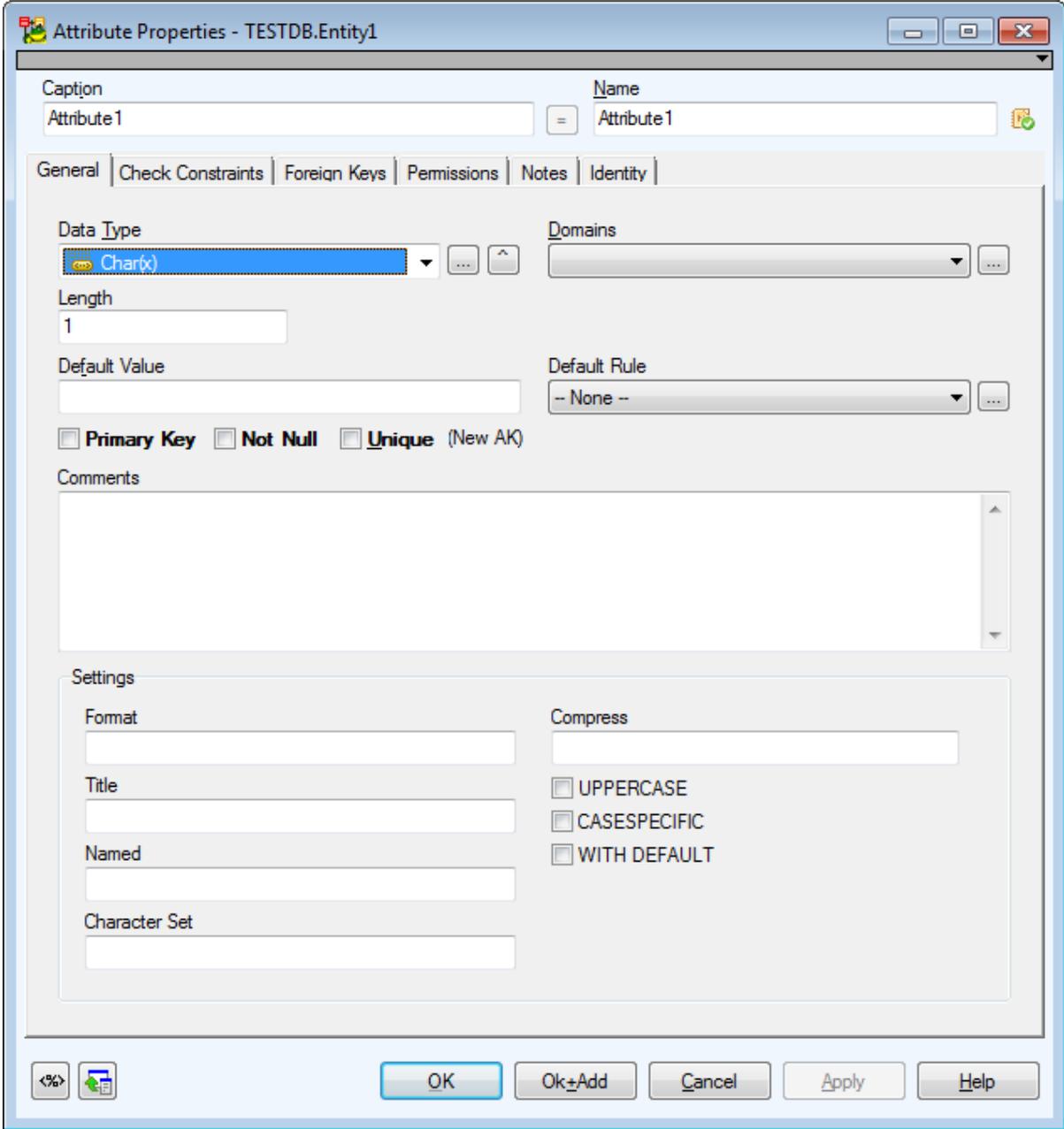


# Specifics - Teradata 13

## Entity

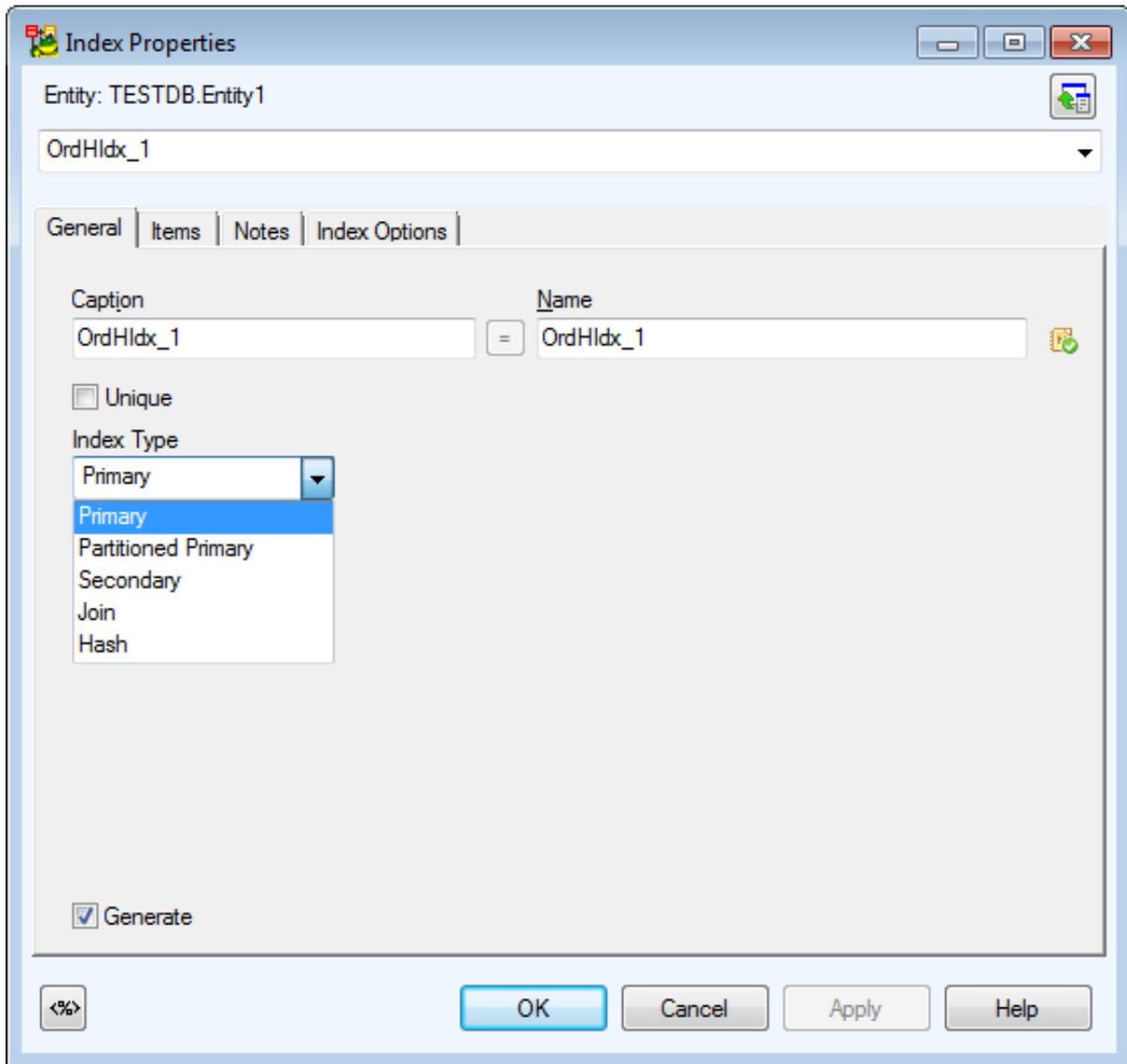


# Attribute

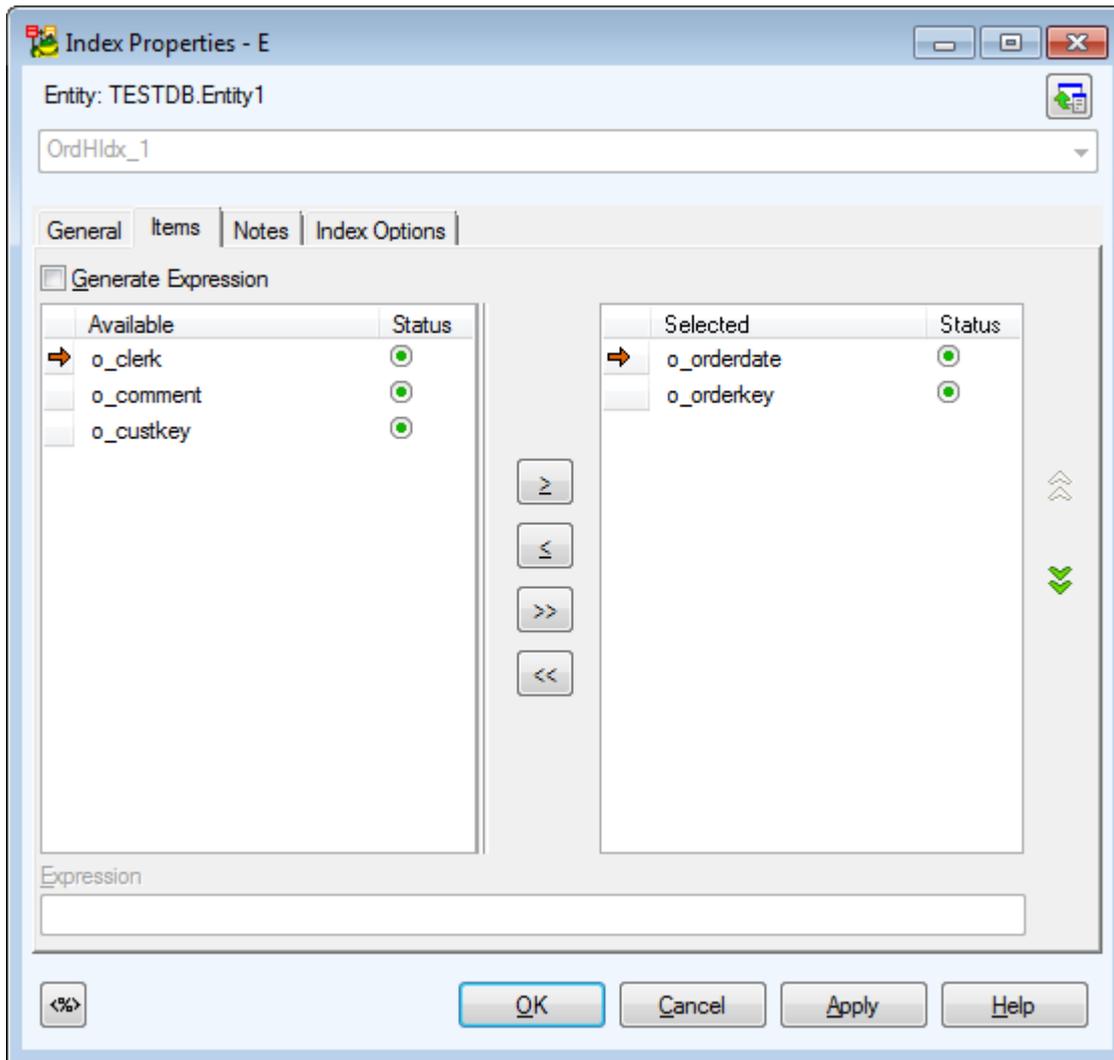


# Index

Primary index is generated only as an inside create table statement.



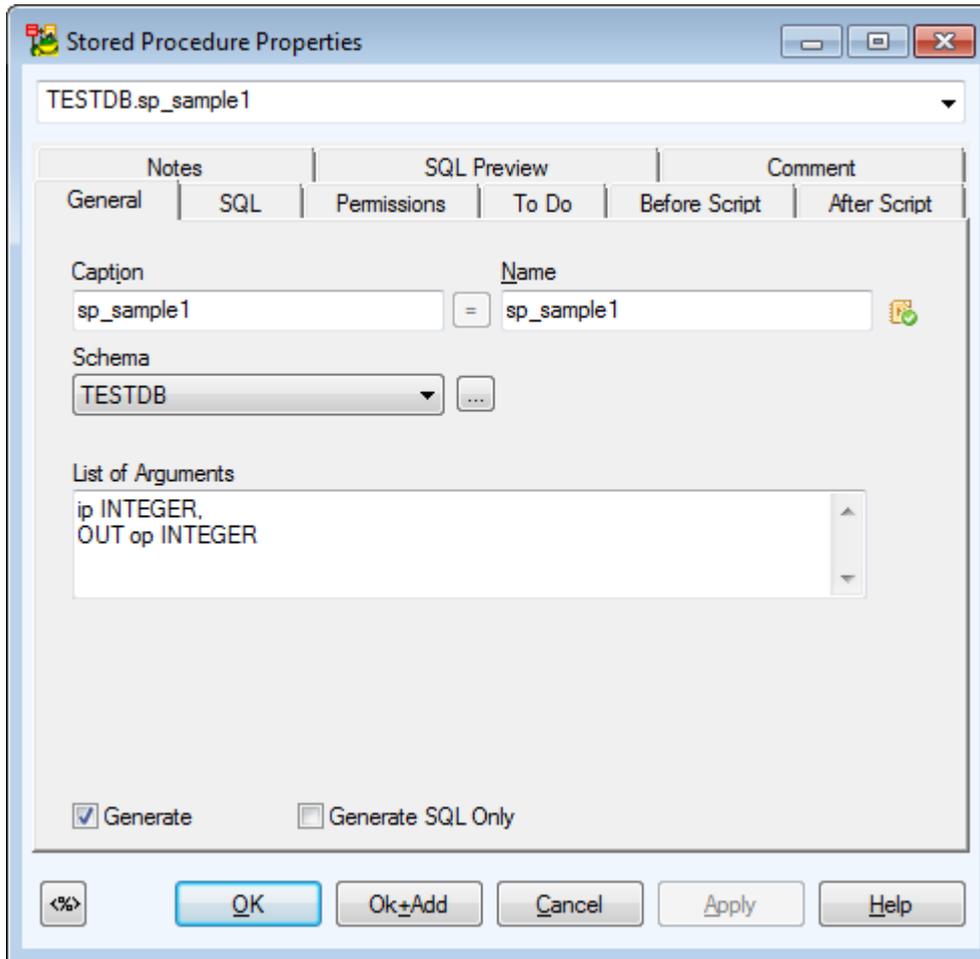
For index type *Join* and *Hash*, Schema is available.



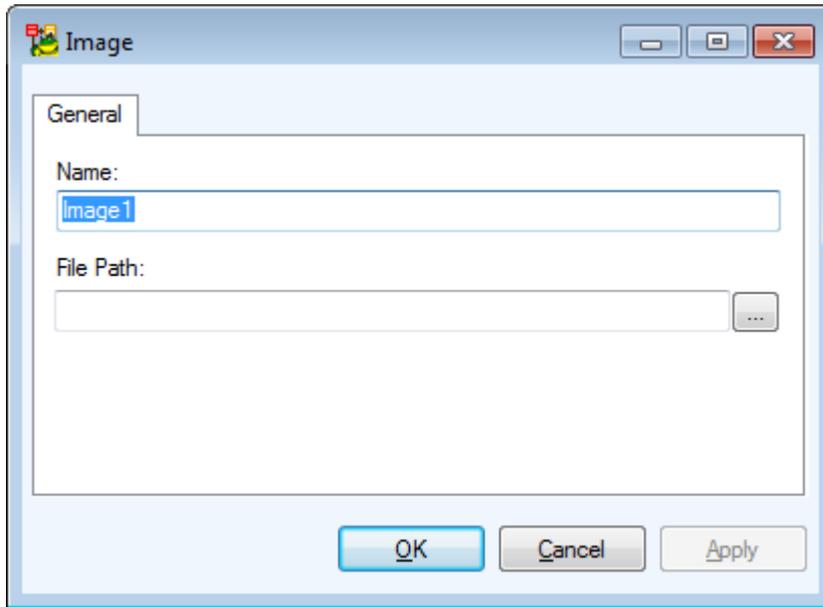
General | Items | Notes | Index Options

BY (o\_orderdate)  
ORDER BY (o\_orderdate);

# Procedure



# Images

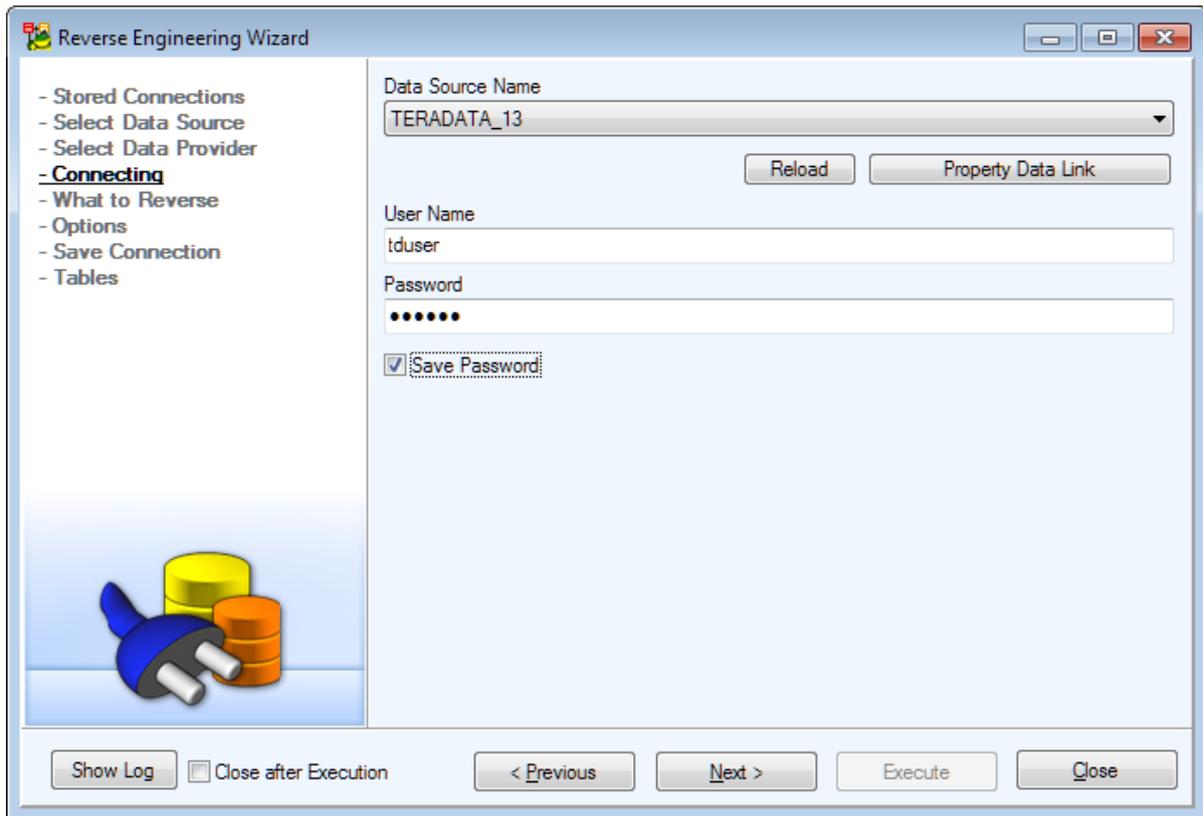


## Reverse Engineering - Teradata 13

Available **Data Providers** are:

- **Connection via ODBC**

**Connection via ODBC**



# Specifics - Vertica Database 8.0

Options are available in **Options | Model | Physical Model | Vertica | Vertica 8.0**

## ***Objects supported***

- Tables
- Columns
- Keys
- CheckConstraints
- Functions
- Libraries
- Procedures
- Projections
- Sequences
- Schemas
- Users
- Views

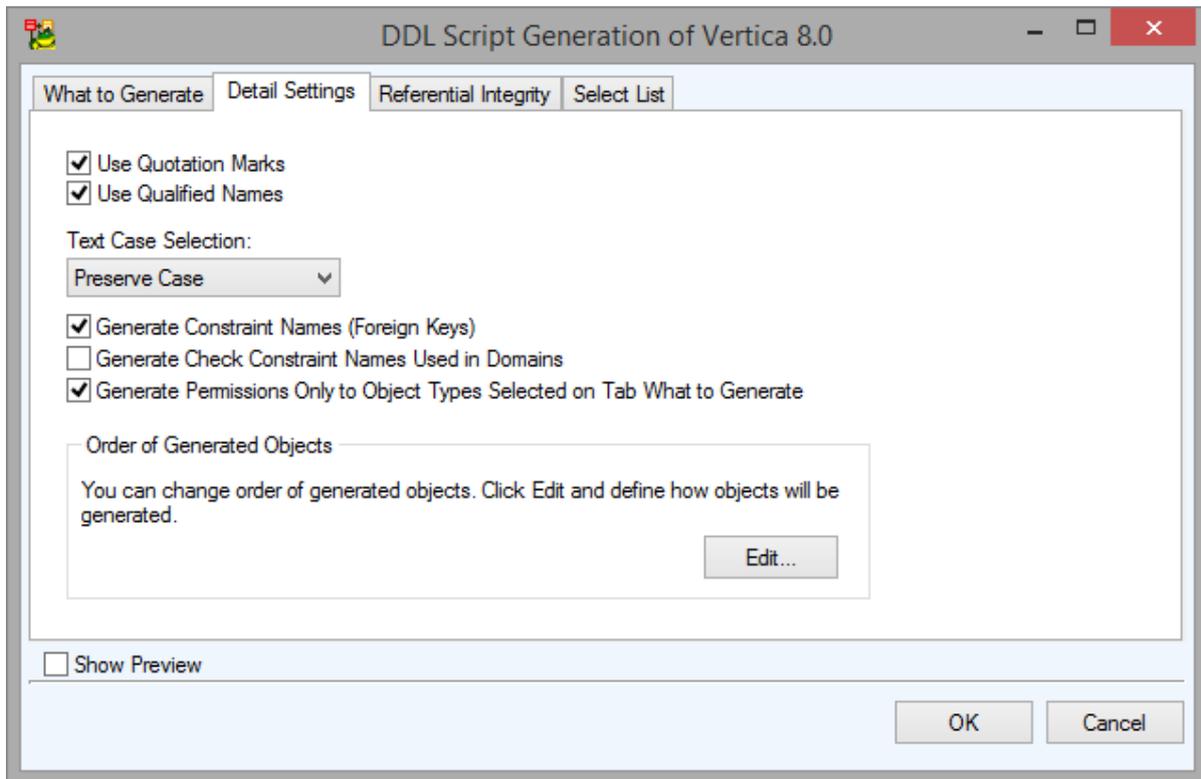
**i** | **NOTE:** In Vertica Database there are no indexes and triggers.

## Reverse Engineering - Vertica 8.0

Available **Data Providers** are:

- **Connection via ODBC**
- Database Connection via ODBC (Open Database Connectivity) Driver. ODBC Driver is not part of Toad Data Modeler. In most cases, it is distributed directly with database.

# Script Generation - Vertica Database 8.0



# Legal Notices

## Third-party components

This product contains the third-party components listed below. For third-party license information, go to <https://www.quest.com/legal/license-agreements.aspx>.

Component	License
TPC Benchmark Samples 5.21	TPC, TPC Benchmark and TPC-C are trademarks of the Transaction Processing Performance Council. All other materials are ©2015 TPC. All rights reserved. Copying is by permission of the Transaction Processing Performance Council
Spring4D 1.2.1	Copyright (c) 2009 - 2018 Spring4D Team Apache 2.0 License
Virtual Treeview 6.6	Mozilla Public License version 1.1

## Acknowledgments

### Gallery Samples

Permissions for database specific and other gallery items were given by Oracle, Microsoft and TPC.org representatives.



## A

- Action Definition 331
- Add Object to Gallery 102
- Advanced Options 204
- Aggregations 566
- Alignment Toolbar 87
- All Items 261
- Allow Null Attributes in Keys 669
- Allow to Modify System Selected OTPs 666
- Alter Reports 372
- Alter Scripts 333
- Alternate Key 116, 186, 204
- Amazon Redshift 1.0
  - Reverse Engineering 689
  - Script Generation 690
  - Specifics 689
- Anchor Points 311
- Application Options 659
- Application Variables 281
  - Examples 284
- Application View 32
- Application Window 35
- Arrange 102
- Arranging Objects in Layers 309
- Associations 566
- associations 566
- Attribute Properties Propagation 670
- Attributes 103, 183
  - Attributes in LER Model
    - Create Attributes 232
    - Edit Attributes 233
  - Copy Attributes 111
  - Create Attributes 103, 183
  - Delete Attributes 114
  - Display Attributes 109
  - Edit Attributes 106
  - Move Attributes 113
  - Order Attributes 110
  - Parent Attributes 114, 202

- Set Attribute Color 109
- Unique Attributes 120

- Auto Complete 260
- Autolayout 308
  - Options 662
- Automatic FK Mapping 200
- Automation 457

## B

- Benefits 22

## C

- Call TDM From Other Applications 457
- Captions 319
- Cardinality 147, 190
- cardinality 239
- Categories 315
- Change Script 333
- Change Scripts
  - Temporary Tables 343
- Check After Script 664
- Check Constraint Rules 178
- Check Constraints 131, 207
- Colors
  - Attributes and Keys 326
- Colors and Alignment Toolbar 85
- Column to Column Alignment 312
- Command Line Parameters 580
- Compare Models 372
- Compare Procedures 173, 217
- Component Inspector 529
- Component Palette 530
- Compound Key 125
- Configuration 659
- Connection types 27
- Connections 436
- Connections by database 27
- Conversion Data Types 352
- Conversion Settings 352

- Convert Model 359
  - LER to PER 357
  - PER to LER 357
  - PER to PER 354
  - Simple Model Conversion 372
- Converting LER to PER 242
  - Data Type Conversion 247
  - Inheritance 254
  - M.N Relationships 248
  - Migration of Keys 249
- Copyright 3
- Create Package 491
- Creating Custom Properties 471
- CSV 290
- Custom Properties 471
  - Example 1 474
- Customize Forms 498
  - Component Inspector 529
  - Component Palette 530
  - Editable Forms and Frames 526
  - Form Explorer 531

## D

- Data Generation 626
- Data Types
  - User Data Types 135
- Data Warehouse Types 318
- Database Connections 436
- Database Support 25, 687
- Databases
  - Enabled/Disabled Databases 684
- DB2 9.5 (LUW)
  - Specifics 692
- DB2 9.7 (LUW)
  - Specifics 700
- DB2 10.1 (LUW)
  - Script Generation 716
  - Specifics 708
- DB2 10.5 (LUW)
  - Script Generation 728
  - Specifics 717

- DB2 11.1 (LUW)
  - Script Generation 732
  - Specifics 729
- DB2 UDB 8
  - Reverse Engineering 697, 705, 713, 725, 729
- DB2 z/OS 11
  - Reverse Engineering 740
  - Script Generation 742
  - Specifics 733
- DDL Script
  - Detailed Settings 299
  - Encoding 295
  - Example of Generated Script 303
  - Generating DDL Script 296
  - Generation Preparation 293
  - Referential Integrity 300
  - Script Preview 302
  - Select List 301
- Default Values 674
- Default Values (scripting) 501
- Defaults 175
  - Defaults in LER Model 257
- Designer 36
- Dialog Boxes 663
- Dictionaries (language) 677
- Dictionary Types 141
  - Export/Import 145
- Display Data Warehouse Type and Size 672
- Display Indexes 672
- Display Key and Index Marks 672
- Display Keys Graphically 672
- Display Level of Entities 322
- Display Modes 319
- Display Not Null Mark 672
- Display Toolbar 81
- Do Not Show Next Time 663
- Docking 39
- Domains 142

## E

- Eclipse Support 667

- EDB 10
  - Script Generation 769
- Edit Menu 61
- Enabled/Disabled Databases 684
- Encoding 295
- Entities 94, 182
  - Copy Entities 98
  - Create Entities 94, 182
  - Display Level 322
  - Edit Entities 95
  - Entities in LER Model 229
    - Options 231
  - Entity Options 101
  - Move Entities 100
- Eureka Log 666
- Excel 287
- Expert Mode
  - Eclipse 667
  - Options 665
- Expert Mode Menu 73
- Expert Mode Toolbar 86
- Export/Import
  - CSV Files 290
  - Export Image 291
  - Microsoft Excel 287
- External Objects 272

## F

- Feedback 24
- File Extensions 680
- File Menu 58
- Find Objects 271
- Fit to Page 402
- Foreign Key 124
  - Automatic Mapping 200
  - Mapping 199
  - Migration of Keys in LER Model 249
  - Understanding Foreign Keys 199
- Form Explorer 531
- Freeware 29
- Full Names 319
- Functions 173, 217

## G

- Gallery 614
  - Basic Actions 614
  - Gallery Explorer 619
  - Modifying Items 618
- General 660
- generate checkbox 96
- Generating DDL Script 293, 296
- Git 634
- Graphics 661
- Graphics Objects Toolbar 84
- Greenplum 4.1
  - Reverse Engineering 753
  - Specifics 743
- Greenplum 4.2
  - Reverse Engineering 759
  - Script Generation 760
  - Specifics 755
- Grid 56
- Grid (Workspace)
  - Grid Size 328
- Grid Toolbar 82

## H

- Handle Points 311
- Help Menu 76
- Help Search 88
- Hot Keys 42
- HTML Reports 419

## I

- IDF1X 317
- IE 317
- Image 279
- Image Export 291
- Index to Foreign Key in Child Entity 669
- Index to Foreign Key Name 670
- Indexes 129, 205
- Infer Relationships 630

- Ingres 9.3
  - Reverse Engineering 764
  - Script Generation 766
  - Specifics 761
- Ingres 10.0
  - Reverse Engineering 767
  - Specifics 767
- Inheritance
  - Create Inheritance 240
  - Edit Inheritance 240
- Inplace Editor 44
- Intelligence Central Toolbar 89
- Interface 31
- Internal Scripting 469

## J

- Join Line Distance 661
- JScript 469

## K

- Key Features 23
- Keys 116
  - Alternate Key 116, 186, 204
  - Compound Key 125
  - Create Keys 116, 186
  - Delete Keys 119
  - Display Keys 116
  - Edit Keys 118
  - Foreign Key 124
  - Foreign Key Mapping 125
  - Parent Key 119
  - Primary Key 116, 186
  - Unique Attributes 120

## L

- Large Models 575
- Layers 309
- Layout 31
- Layout Menu 68
- Layout Toolbar 85

- Legal Notices 1014
- Limitations 29
- Line Autolayout 323
- Linking Method 119
- Load Orphaned FK Constraints to Child Table After Script 663
- Load Toad for Oracle Connections as Connection in Toad Data Modeler 664
- Localized Reports 678
- Logical Model 224
  - Benefits of Logical Model 225
  - Objects 228
    - Format Objects 228
  - Options 667
- Logical Only 230
- Logo 279
- Loupe 328

## M

- Macros 448, 532
  - Macros and User Forms 550
  - Macros and User Forms - Sample 553
  - Productivity Pack 539
  - Rename Objects Pack 547
- Macros Menu 72
- Main Toolbar 78
- Materialized Views 169, 213
- Merge Models 344
  - Simple Merge 351
- Message Explorer 45
- Metamodel 558
  - Add New Properties 493
  - Aggregations
    - Create Aggregation 566
  - Associations
    - Create Association 566
  - Classes
    - Add Classes 560
    - Create Classes 560
    - Edit Classes 561
  - Generalizations
    - Create Generalizations 565
    - Edit Generalizations 566

- Methods
  - Create Methods 564
  - Edit Methods 564
- Properties
  - Create Properties 563
  - Edit Properties 563
- Microsoft Access 2007/2010
  - Reverse Engineering 777
  - Script Generation 777
  - Specifics 772, 775
- Microsoft Azure SQL
  - Specifics 779
- Microsoft Azure SQL Database
  - Reverse Engineering 780
  - Script Generation 783
- Microsoft SQL Server 2000
  - Reverse Engineering 845
  - Script Generation 841, 843, 848
- Microsoft SQL Server 2012
  - Script Generation 823
  - Specifics 784
- Microsoft SQL Server 2014
  - Script Generation 832, 838
  - Specifics 824, 839, 842
- Microsoft SQL Server 2016
  - Specifics 833
- Model
  - Large Models Tips 575
  - Logical Model 224
  - Options 667
  - Physical Model 93
  - Universal Model 224
- Model Actions 329
- Model Compare 372
- Model Conversion 359
- Model Explorer 47
- Model Menu 69
- Model Objects Toolbar 83
- Model Options 659
- Model Properties 91
- Model Statistics 93
- Model Toolbar 81
- Model Update 390

- Model Verification 621
- Modify Form 498
- MS Excel 287
- MySQL 5.0
  - Reverse Engineering 857
  - Script Generation 857
- MySQL 5.6
  - Script Generation 859, 862
  - Specifics 849
- MySQL 5.7
  - Reverse Engineering 859-860
  - Specifics 858, 860

## N

- Names 319
- Naming Conventions
  - About 596
  - Character/Word Replacement 604
  - Glossary 607
  - On Form Synchronization 599
  - Rules 602
  - Settings 601
  - Valid Character Set Syntax 604, 607
  - Valid Characters 604
  - Verification and Synchronization 598
- Naming Conventions Toolbar 82
- Navigation 328
- New in This Release 24
- Not Null and Mandatory Parent 125
- Notation 147, 190
  - Change Notation 317
- NotNull Property for PK and AK Attributes 117, 187

## O

- Object Navigator 49
- Object Types and Properties 50
- Object Viewer 54
- Objects
  - Add Objects 258
  - Align Objects 264
  - Copy Objects 266

- Delete Objects 269
- Edit Objects 259
- Format Objects 260
- Move Objects 267
- Rename Objects 266
- Select Objects 262
- Objects Menu 65
- On Form Synchronization 599
- On Form Verification 623
- Open Only One Instance of Forms 660
- Optimal Style On Move 69
- Options 659
  - Default Options 659
  - Dialog Boxes 663
  - General 660
  - Graphics 661
  - Import and Export 672
  - Logical Model 667
  - Model 667
  - Other Options 659
  - Paths 663
  - Physical Model 669
  - Print 665
  - Reports 664
  - Reverse Engineering 663
  - Scripting 664
  - Specific Databases 672
  - Toad for Oracle 664
  - Version Control System 665
- Oracle 9i
  - Reverse Engineering 915
  - Script Generation 912, 918
  - Specifics 912
- Oracle 11g R1
  - Specifics 864
- Oracle 11g R2
  - Script Generation 894
  - Specifics 887
- Oracle 12c
  - Script Generation 905, 911
  - Specifics 897, 906
- Order of Generated Objects 293
- OTPs 50

Owner Assignment 629

## P

- Package Explorer 464
- Packages 448
  - Create New Package 491
- Page Format 402
- Parent Key 119
- Paths 663
- PDF Printing 404
- PDF Reports 431
- PER to PER Conversion 354
- Permissions 134, 222
- Physical Model 93
  - Benefits of Physical Data Model 93
  - Options 669
- Plotter Printing 406
- PostgreSQL 8.2
  - Script Generation 950
- PostgreSQL 8.3
  - Reverse Engineering 769, 948
- PostgreSQL 9.4
  - Specifics 919
- Primary Key 116, 186
- Print 402
  - Page Setup 402
  - PDF Printing 404
  - Plotter Printing 406
  - Preview 403
  - Print Options 665
- Printing to PDF 404
- Procedures 170, 214
  - Compare Procedures 173, 217
- Product Improvement Program 24
- Productivity Pack 539
- Projects
  - Add Models to Project 409
  - Create New Project 407

## Q

Quick Search 611

Quick Tips 581

## R

- Recalculate Size 671
- Refactoring Utility 627
- Referential Integrity 157
  - Set up Referential Integrity Rules 196
- Relationship Lines 159, 310
- Relationship Names 314
- Relationships 152
  - Create Relationships 154, 188
  - Edit Relationships 155
  - Referential Integrity 157
  - Relationship Lines 159, 310
    - Line Autolayout 323
  - Relationship Options 164
  - Relationship Types 152
    - Identifying Relationship 152
    - M.N Relationships 153
    - Non-Identifying Relationship 153
    - Self-Relationship 153
  - Relationships in LER Model 236
- Rename 574
- Rename Objects Pack 547
- Renaming 627
- Reports
  - Alter Reports 372
  - Customizing XSLT Templates 415
  - HTML Reports 419
  - Localized Reports 678
  - Options 664
  - PDF Reports 431
  - RTF Reports 426
  - XSL Transformation 414
- Reverse Engineering
  - Options 663
  - Universal Model 446
- RTF Reports 426
- Rules 258
- Run Only One Instance of Application 660

## S

- Sample Scripts 496
- SAP ASE 16.0
  - Reverse Engineering 983
  - Script Generation 986
  - Specifics 979
- Save Models and Projects as Formatted XML Files 660
- Save Selection 50
- Schema
  - Schema Assignment 629
- Schema/Owner Assignment 629
- Schemas 178, 219
- Script Editor 456
  - Rollback 456
  - Scripting in Script Editor 468
- Script Examples 483
- Script Explorer 452
- Script Samples
  - Create New Objects 496
  - Dialogs 507
  - File System Scripts 514
  - Getting Settings Information 517
  - Iterate Entities and Attributes 517
- Scripting and Customization 447
  - Add Events 502
  - Call Script 519
  - Create New Package 491
  - Create Script 515
  - Internal Scripting 469
  - Modify HTML Reports 519
  - Script Editor 456
  - Script Examples 483
  - Script Explorer 452
  - Scripting Options 664
  - Scripting Window 451
- Scripting Window 451
  - Access Property Values 509
- Scripts 448
- Search 611
- Search Bar 77
- Selection 50

- Settings 659
  - Settings Menu 75
  - Shapes (Model Independent) 278
    - Caption of Categories 281
    - Image 279
    - Line 279
    - Note 278
    - Stamp 280
  - Shortcuts (keys) 42
  - Shortcuts (objects) 274
    - Create Shortcuts 275
    - Edit Shortcuts 276
    - Remove and List Shortcuts 276
  - Show Non-printable Characters in SQL Editors 669
  - Simple Change Script 343
  - Simple Model Conversion 372
  - Simple Model Merge 351
  - snap to objects 268
  - Snap to Objects 661
  - Snap to Objects Distance 661
  - SQLite 3.7
    - Reverse Engineering 961
    - Script Generation 963
    - Specifics 951
  - Subversion 635
    - Application Settings 641
    - Options 665
    - Project Settings 644
    - Subversion Actions 648
  - Supported Databases 25, 687
  - Sybase ASE 15.5
    - Reverse Engineering 969
    - Specifics 965
  - Sybase ASE 15.7
    - Reverse Engineering 976
    - Specifics 973
  - Sybase IQ 15.2
    - Reverse Engineering 993
    - Specifics 987
  - Sybase SQL Anywhere 11
    - Reverse Engineering 999, 1002
    - Script Generation 1004
    - Specifics 996, 1002
  - Synchronization 400
  - Synchronization of NotNull and Mandatory Parent 125
  - Synonyms 178
- ## T
- Target Database 181
  - Template Editor 568
  - Templates 567
    - Change Properties 571
    - Complete Template 570
    - Create Templates 571
    - Default Template 572
    - Delete Templates 572
    - Disable Template 572
    - Manage Templates 571
    - Predefined Templates 570
    - Template Body 570
    - Toad for Oracle Templates 573
  - Temporary Tables 343
  - Teradata 13
    - Reverse Engineering 1010
    - Script Generation 1013
    - Specifics 1005
  - Third-party Components 1014
  - To Do List 609
  - Toad for Oracle 582
    - Aliases 584
    - Auto Replacement Words 574
    - Connections 584
    - Import Templates 573
    - Import Toad for Oracle ER Diagrams 585
    - Open Projects 587
    - Options 664
    - Toad for Oracle Icons 588
    - Use Toad for Oracle as Default Editor 587
  - Toad for Oracle Integration 664
  - Toad Intelligence Central
    - Basic Actions 589
    - Notifications 595
    - Publishing Models/Reports 594
  - Toad World 24

Tools Menu 72  
Triggers 131, 208  
txp Files 680

## U

Undo/Redo Toolbar 83  
Unhide Line 314  
Unique Identifier 235  
    Editing Identifiers 235  
    Linking Method 236  
Universal Model 224  
Update Model 390  
Use Read Only Locking Mechanism 665  
Use Toad for Oracle Icons in Toad Data  
    Modeler 664  
User Data Types 135  
User Feedback 24  
User Groups 180, 221  
Users 178, 219  
    User Assignment 629  
Users Toolbar 82

## V

Verification and Synchronization 598  
Verification Log 622  
Verify Model 621  
Verison Manager  
    List of Projects, Files, Versions 657  
Version Control System 634, 665  
    (Step 1) Gettings Started with Apache  
        Subversion 635  
    (Step 2) Application Settings 641  
    (Step 3) Project Settings 644  
    (Step 4) Subversion Actions 648  
Version Manager 651  
    Options 654  
    Projects and Files 652  
Vertica 8  
    Specifics 1012  
Vertica Database 8.0  
    Reverse Engineering 1012  
View Menu 63

Views 166, 210  
Views Toolbar 80

## W

Warehouse 318  
Window Menu 75  
Window Toolbar 80  
Workspace 36  
    Format Workspaces and Objects 323

## Z

Zoom 328  
Zoom Toolbar 85  
Z-Order 310