

Quest® NetVault® Backup Plug-in *for MySQL*
12.2

ユーザーズ・ガイド



© 2019 日本クエスト・ソフトウェア株式会社

ALL RIGHTS RESERVED.

本書には、著作権によって保護されている機密情報が記載されています。本書に記載されているソフトウェアは、ソフトウェア・ライセンスまたは機密保持契約に基づいて提供されます。本ソフトウェアは、当該契約の条項に準拠している場合限り、使用または複製することができます。本書のいかなる部分も日本クエスト・ソフトウェア株式会社の書面による許可なしに、購入者の個人的な使用以外の目的で、複写や記録などの電子的または機械的ないかなる形式や手段によっても複製または転送することはできません。

本書には、Quest Software 製品に関連する情報が記載されています。明示的、黙示的、または禁反言などを問わず、本書または Quest Software 製品の販売に関連して、いかなる知的所有権のライセンスも付与されません。本製品の使用許諾契約の契約条件に規定されている場合を除き、QUEST SOFTWARE はいかなる責任も負わず、製品に関連する明示的、黙示的または法律上の保証（商品性、特定の目的に対する適合性、権利を侵害しないことに関する黙示的保証を含む）を否認します。QUEST SOFTWARE は、損害が生じる可能性について報告を受けたとしても、本ドキュメントの使用、または使用できないことから生じるいかなる、直接的、間接的、必然的、懲罰的、特有または偶発的な障害（無期限、利益の損失、事業中断、情報の損失も含む）に対しても責任を負わないものとします。Quest Software は、本書の内容の正確性または完全性について、いかなる表明または保証も行わず、通知なしにいつでも仕様および製品説明を変更する権利を有します。Quest Software は、本書の情報を更新する一切の義務を負いません。

本文書の使用に関してご不明な点がございましたら、下記までお問い合わせください。

日本クエスト・ソフトウェア株式会社
宛先：法律部門
東京都新宿区西新宿 6-10-1
日土地西新宿ビル 13F

日本国内および海外の事業所の情報に関しては、弊社の Web サイト (<https://www.quest.com/jp-ja>) を参照してください。

特許

高度なテクノロジーは Quest Software の誇りです。特許および出願中の特許がこの製品に適用される可能性があります。この製品に適用される特許に関する最新情報については、<https://www.quest.com/jp-ja/legal> の弊社 Web サイトを参照してください。

商標

Quest、Quest ロゴ、および NetVault は、日本クエスト・ソフトウェア株式会社の商標および登録商標です。Quest の商標の詳細な一覧については、<https://www.quest.com/jp-ja/legal/trademark-information.aspx> を参照してください。その他すべての商標および登録商標は各社に帰属します。

凡例

- **警告**：警告アイコンは、潜在的な資産の損害、個人の負傷または死亡の可能性を表しています。
- ! **注意**：注意アイコンは、指示に従わなかった場合に、ハードウェアの損傷やデータの損失につながる可能性があることを表しています。
- i **重要、メモ、ヒント、モバイル**、または**ビデオ**：情報アイコンは、サポート情報を表しています。

NetVault Backup Plug-in for MySQL ユーザーズ・ガイド
更新 - 2 0 1 9
ソフトウェア・バージョン - 12.2
MYG-101-12.2-JA-01

目次

NetVault Backup Plug-in for MySQL – はじめに	5
NetVault Backup Plug-in for MySQL : 概要	5
主な利点	5
機能概要	6
対象ユーザー	7
参考資料	7
プラグインのインストールと削除	8
インストールの前提条件	8
MySQL サーバーでのバイナリ・ログの有効化 (MySQL Standard/Community オプションのみ)	9
推奨構成の確認	10
プラグインのインストールまたはアップグレード	11
プラグインの削除	11
特定 MySQL インスタンスの削除	11
プラグインの設定	13
デフォルト設定の構成	13
既存インスタンス設定の更新	16
エラー条件のデフォルト・アクションの設定 (オプション)	17
データのバックアップ	19
データのバックアップ : 概要	19
バックアップ戦略の策定	22
バックアップの実行	26
バックアップ対象データの選択	27
バックアップ・オプションの設定	28
ジョブのファイナライズと実行	31
データのリストア	32
データのリストア : 概要	32
MySQL Standard/Community に利用可能なリストア方法の確認	32
MySQL Enterprise バックアップに利用可能なリストア・オプションの確認	33
MySQL におけるデータのリストア	34
リストア対象データの選択	34
リストア・オプションの設定	35
ジョブのファイナライズと実行	41
MySQL Standard/Community 用リストア・シナリオ例	42
MySQL Enterprise バックアップ用リストア・シナリオ例	67
高度な MySQL Standard/Community 用リストア手順	68
リストア中にデータベース名を変更する	69

同じサーバー上の別の MySQL インスタンスへリストアする	69
異なる MySQL サーバーへのリストア	71
MySQL レプリケーションの使用	74
MySQL レプリケーション環境でのプラグイン使用：概要	74
レプリケーションのサポートの有効化	74
レプリケーション・サーバーのバックアップ	75
レプリケーション設定のバックアップ	75
レプリケーション・サーバーのリストア	75
フェイルオーバー・クラスタ環境でのプラグインの使用	76
MySQL サーバー・フェイルオーバー・クラスタリング：概要	76
プラグインのインストールまたはアップグレード	77
インストールの前提条件	77
ソフトウェアのインストール	77
プラグインの設定	77
データのバックアップ	78
データのリストア	78
トラブルシューティング	79
テクニカル・サポート用リソース	80

NetVault Backup Plug-in for MySQL — はじめに

- [NetVault Backup Plug-in for MySQL : 概要](#)
- [主な利点](#)
- [機能概要](#)
- [対象ユーザー](#)
- [参考資料](#)

NetVault Backup Plug-in for MySQL : 概要

Quest® NetVault® Backup Plug-in for MySQL (Plug-in for MySQL) は、複数の MySQL ストレージ・エンジンのバックアップとリカバリを、複雑なスクリプト作成の必要もなく単一のジョブに統合します。[MySQL Enterprise バックアップ] オプション (MEB ベース・バックアップ方法) を使用する場合、本プラグインはバックアップ中の InnoDB テーブルに対するホット・バックアップをサポートします。[MySQL Standard/Community] オプション (mysqldump ベース・バックアップ方法) を使用する場合、本プラグインは読み取り専用アクセスでデータをオンラインに保ちながら、すべてのテーブルのウォーム・バックアップをサポートします。また、[MySQL Standard/Community] オプションを使用することにより、プラグインはさらに細かいリストアを実行する高度な特定時点 (PIT) リストア機能を提供します。この機能により正確な時点へのリストアが可能になるため、データの損失を大幅に低減できます。

主な利点

- **MySQL 環境に対する安心感を高め、リスクを低減** : Plug-in for MySQL には、複雑なバックアップ・スクリプトの作成は必要ありません。また、多様なリカバリ・シナリオに対応できる柔軟性を備えています。ユーザーは、MySQL に関する詳細な知識がなくても、コミットされたトランザクションがバックアップ中に失われないようにするバックアップ・ポリシーを実装したり、バイナリ・ログをパージする適切なタイミングを把握することができます。この知識はすべてプラグインに組み込まれています。

このほか、Plug-in for MySQL の柔軟なバックアップ機能には以下のものがあります。

- データをオンラインにした状態、すなわちアクセス可能な状態でフル、増分、および差分バックアップを実行
- 複数のストレージ・エンジンに渡って共通のユーザー・インターフェイスを採用
- テーブル・レベルおよび表示レベルまで保護
- 複数のストレージ・エンジンを単一のジョブに統合

プラグインを使用してバックアップ・ポリシーを実装すると、障害発生時に必要となるリカバリ作業をおろそかにすることなく、より重要なタスクに専念することができます。また、MySQL データが保護されていることが分かっているため、IT 管理者の安心感が高まります。

- **高速なリストアによりダウンタイムを短縮**：リストア対象、リストア元のバックアップ・セット、また該当する場合はリストアする時点または位置を選択するだけで、プラグインが自動的にリストアを実行するため、それ以上の操作は不要です。また、人的な操作への依存を低減することによってリストアを高速化するだけでなく、再起動を必要とする複雑なスクリプトの構文エラーの可能性も排除します。

このほか、Plug-in for MySQL は以下のリストア機能を備えています。

- 時間および位置に基づく特定時点のフルおよび増分リストア
 - インスタンス全体、個々のデータベース、または個々のテーブルと表示のリストア
 - リストア中のデータベース名の変更
 - 代替 MySQL インスタンスへのリストア
- **ビジネスの継続性を確保**：ビジネス上重要なアプリケーションのデータ保護でオフサイト・バックアップは重要です。本プラグインは幅広いバックアップ・デバイスと NetVault Backup との統合を有効に活用します。NetVault Backup では、バックアップの保存先バックアップ・デバイスを柔軟に選択することができます。バックアップをオンラインで仮想テープ・ライブラリ (VTL) に保存できます。また、そのジョブを複数の MySQL インスタンスや、その他の専用データベースで共有される物理テープ・ライブラリ、または一般的なバックアップ・ファイル用の物理テープ・ライブラリにも複製できます。
 - **高度な MySQL レプリケーション技術のサポート** — 『MySQL リファレンス・ガイド』で詳しく説明されているように、MySQL は、1 台のサーバーがマスタとして、その他の 1 台以上のサーバーがスレーブとして機能する、一方向の非同期レプリケーションをサポートしています。

単一マスタ・レプリケーションでは、マスタ・サーバーはそのバイナリ・ログに更新を書き込み、これらのファイルのインデックスを管理してログ・ローテーションを追跡します。バイナリ・ログ・ファイルは、スレーブ・サーバーに送信される更新の記録として機能します。スレーブはマスタに接続すると、スレーブが最後に正しく更新を行ったときにどこまでログを読み取ったかをマスタに通知します。スレーブは、それ以降に行われた更新を受信してブロックし、マスタが新たな更新を通知するまで待機します。

Plug-in for MySQL は、MySQL 環境が保護され、障害復旧に備えてオフサイトに保存されているという安心感を提供します。同時に、管理者が 24x7 体制で待機する必要がなくなります。経験が浅くてもリストアを開始できるため、ダウンタイムが短縮され、ビジネス継続性が高まります。

機能概要

- クラスタ対応の MySQL サーバー 5.6 をベースとした MySQL Cluster Network Database (NDB) 7.x のサポート。この機能ではバックアップに mysqldump ユーティリティを使用します。
- **[MySQL Standard/Community]** オプションでは、以下の機能がサポートされます。
 - フル・バックアップと増分バックアップ
 - 差分バックアップ
 - 個々のデータベース / テーブル・コピーのみのバックアップ
 - InnoDB、MyISAM、MERGE (または MRG_MyISAM)、Memory/Heap、Federated、Berkeley DB (BDB)、Archive、および CSV ストレージ・エンジンをサポート
 - 複数のストレージ・エンジンに渡って共通のユーザー・インターフェイスを採用
 - 時間および位置に基づく PIT リストア
 - データ損傷の前および後への PIT リストア
 - 個々のテーブル、データベース、またはインスタンス全体のリストア

- リストア中にデータベース名を変更
 - 代替インスタンスへのリストア
 - MySQL レプリケーションのスレーブおよびマスタ・インスタンスのバックアップをサポート
- **[MySQL Enterprise バックアップ]** オプションでは、以下の機能がサポートされます。
 - フル・バックアップと増分バックアップ
 - InnoDB、MyISAM、MERGE（または MRG_MyISAM）、Archive、および CSV ストレージ・エンジン
 - InnoDB テーブルのホット・バックアップ
 - 転送可能なテーブルスペース（TTS）バックアップ
 - 複数のストレージ・エンジンに渡って共通のユーザー・インターフェイスを採用
 - 個々のテーブル、データベース、またはインスタンス全体のリストア
 - TTS バックアップのリストア中での単一テーブルの名前変更
- ポイント・アンド・クリック WebUI

対象ユーザー

ルーチン的なバックアップ・オペレーションの作成および実行に MySQL データベース管理者の高度なスキルは要求されませんが、バックアップおよびリカバリ戦略を効率的に定義したり、高度なリカバリ・シナリオを実行する場合はこのスキルが要求されます。

参考資料

Quest は、本プラグインの設定時および使用中に以下のドキュメンテーションをすぐに利用できるよう準備しておくことをお勧めします。

- **MariaDB ドキュメンテーション** : <https://mariadb.com/kb/en/library/documentation/>
- **『MySQL <X> リファレンス・マニュアル』**（ここで、<X> は、MySQL サーバーにインストールされた MySQL バージョンを示します）。
 - **MySQL 5.7** : <http://dev.mysql.com/doc/refman/5.7/en/index.html>
 - **MySQL 5.6** : <http://dev.mysql.com/doc/refman/5.6/en/index.html>
 - **MySQL 5.5** : <http://dev.mysql.com/doc/refman/5.5/en/index.html>
- **NetVault Backup ドキュメンテーション**
 - Quest NetVault Backup インストール・ガイド : このガイドでは、NetVault Backup サーバーおよびクライアント・ソフトウェアのインストール方法について詳しく説明しています。
 - Quest NetVault Backup アドミニストレーターズ・ガイド : このガイドでは、NetVault Backup の使用方法と、すべてのプラグインで共通の機能について詳説します。
 - Quest NetVault Backup CLI リファレンス・ガイド : このガイドでは、コマンドライン・ユーティリティの詳細な説明を提供します。

これらのガイドは、<https://support.quest.com/technical-documents> からダウンロードできます。

プラグインのインストールと削除

- インストールの前提条件
- 推奨構成の確認
- プラグインのインストールまたはアップグレード
- プラグインの削除
- 特定 MySQL インスタンスの削除

インストールの前提条件

Plug-in for MySQL のインストールにあたり、MySQL サーバーとして使用するマシンに以下のソフトウェアがインストールされ、かつ正しく設定されていることを確認します。

- **NetVault Backup サーバーおよびクライアント・ソフトウェア** : MySQL サーバーとして構成されたマシン上に、少なくともクライアント・バージョンの NetVault Backup ソフトウェアをインストールする必要があります。
- **MySQL データベース・ソフトウェア**
- **MySQL サーバーでバイナリ・ログを有効にする** (MySQL Standard/Community オプションのみ) この設定により、MySQL サーバーの特定時点 (PIT) バックアップおよびリストアがサポートされます。詳細は、「MySQL サーバーでのバイナリ・ログの有効化 (MySQL Standard/Community オプションのみ)」を参照してください。
- **適切なバージョンの MySQL データベース・クライアント・パッケージ** : プラグインは、MySQL クライアント・パッケージとともにインストールしたコンポーネントと対話し、プラグインのより多くの機能へのアクセスを可能にします。このパッケージとともにインストールされるコンポーネントのバージョンは、インストールされている MySQL のバージョンと互換性があることが必要です。まず、以下の 2 つの MySQL コンポーネントをインストールし、そのバージョンを確認する必要があります。
 - **mysqldump** : このユーティリティは、さまざまなタイプの MySQL ストレージ・エンジンのバックアップ / リストアを可能にします。このコンポーネントのバージョンが、旧バージョンの Plug-in for MySQL とともに提供されたバージョン **ではなく**、ご利用中の MySQL のバージョンと互換性があることを確認する必要があります。
 - **mysqlbinlog** : このユーティリティは、データの PIT バックアップおよびリストアを可能にします。このコンポーネントがインストールされている MySQL バージョンと併用できる適切なバージョンであるかを確認する必要があります。
- **[MySQL Enterprise バックアップ] : [MySQL Enterprise バックアップ] オプション** (MEB ベース・バックアップ方法) をスタンドアロン (非クラスタ化) 環境で使用する場合、ご使用の環境が以下の要件を満たす必要があります。
 - Windows、Linux、および UNIX 環境において、MySQL サーバーのバージョンが 5.6 または 5.7 であること。
 - MySQL Enterprise バックアップ製品のバージョン 3.12 または 4.0 がインストールされていること。Commercial Edition を選択した状態の MySQL Enterprise Edition で MySQL Enterprise バック

アップが利用可能であること。インストール手順についての詳細は、MySQL Enterprise Edition 製品の利用可能なドキュメンテーションを参照してください。

- Windows Server 2008 R2、Windows Server 2012 R2、Windows Server 2016、および Red Hat Enterprise Linux (RHEL) 7.x 上で、MySQL 5.6 とともに、MEB オプションのバージョン 3.12 を使用できます。
- Windows Server 2008 R2、Windows 2012 R2、Windows Server 2016、および RHEL 6.x、7.x 上で、MySQL 5.7 とともに、MEB オプションのバージョン 4.0 を使用できます。

i **重要:** Windows Server 2008 R2またはWindows Server 2016を使用する場合、マシンのローカル管理者グループに**管理者グループのローカル・アカウントとメンバー**を追加するには、Windows Computer Managementを使用します。

RHEL 6.xを使用する場合、実行する前にライブラリが最新であることを確認します。

MySQL サーバーでのバイナリ・ログの有効化 (MySQL Standard/Community オプションのみ)

[MySQL Standard/Community] オプションを使用した特定時点バックアップ/リストアのサポートを構成する前に、MySQL のバイナリ・ログを有効にする必要があります。

LinuxまたはUNIXベースのMySQLサーバーにおけるログ・オンの有効化

- 1 MySQL インストール・ディレクトリにアクセスし、**my.cnf** という名前の MySQL 構成ファイルを特定します。
このファイルの名前と位置は、ご使用の MySQL 構成によって異なります。詳しくは、MySQL ドキュメンテーションを参照してください。
- 2 テキスト・エディタを使用してファイルを開き、**[mysqld]** セクションに移動します。
- 3 デフォルト MySQL ディレクトリを使用して MySQL バイナリ・ログを有効にするには、以下のエントリを追加します。

```
log-bin
```

i **重要:** バイナリ・ログを有効にするには、以下の構文を使用して**my.cnf**ファイルに追加した「**log-bin**」エントリを設定することができます。

```
log-bin=<NameOfDestinationFile>
```

バイナリ・ログの格納先ファイル名を指定する場合、ファイルへのフル・パス情報やファイルの拡張子は含めず、ファイルそのものの名前だけを指定するよう注意してください。プラグインのインストールを実行する前に、上記の手順と共にバイナリ・ログの有効化の詳細について『MySQLリファレンス・マニュアル』を参照してください。

- 4 変更を有効化するには、MySQL サーバーを再起動します。

WindowsベースのMySQLサーバーにおけるログ・オンの有効化

- 1 **MySQL Administrator** アプリケーションを起動します (詳しくは関連する MySQL ドキュメンテーションを参照してください)。

i | **重要:** MySQL Administratorがまだインストールされていない場合、Linux/UNIXシステムで設定ファイルを更新し、次にMySQLサービスを一旦停止してから再起動してバイナリ・ログを有効化します。

- 2 [MySQL Administrator] ウィンドウの左ペインで、[Startup Variables] をクリックします。
- 3 右ペインで、[ログ・ファイル] タブを選択します。
- 4 [Binary Logfile Name] に、バイナリ・ログ・ファイルの名前として一意の値を入力するか、デフォルト値である log-bin を使用するためにフィールドを空白のままにします。

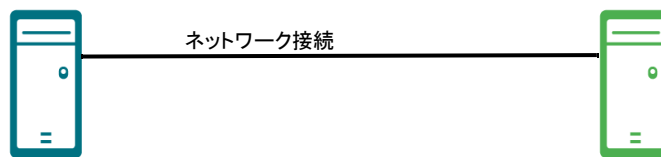
i | **重要:** バイナリ・ログの格納先ファイル名を指定する場合、ファイルへのフル・パス情報やファイルの拡張子は含めず、ファイルそのものの名前だけを指定するよう注意してください。プラグインのインストールを実行する前に、上記の手順と共にバイナリ・ログの有効化の詳細について『MySQLリファレンス・マニュアル』を参照してください。

- 5 MySQL Administrator アプリケーションを閉じます。
- 6 変更を有効化するには、MySQL サーバーを再起動します。

推奨構成の確認

1つのマシンを NetVault Backup サーバーおよび MySQL サーバーの両方として設定する（つまり、すべてのソフトウェアのインストールおよび設定を1つのマシンで実行する）ことはできますが、Quest ではこれらのエンティティを *別々の* マシンで実行することをお勧めします。

表1. 推奨構成



MySQLサーバー・マシン	NetVault Backupサーバー・マシン
インストールされているソフトウェアおよび構成	インストールされているソフトウェアおよび構成
<ul style="list-style-type: none">• MySQLデータベース・ソフトウェア(バージョン5.5以降)• NetVault Backupサーバーおよびクライアント・ソフトウェア• Plug-in for MySQL• バイナリ・ログを有効にする (MySQL Standard/Communityオプションのみ)• mysqldump/mysqlbinlogユーティリティ: インストールされているMySQLのバージョンとの互換性が必要• mysqlbackupユーティリティ: [MySQL Enterpriseバックアップ]オプションのみ	<ul style="list-style-type: none">• NetVault Backupサーバー・ソフトウェア• NetVault Backupクライアントとして追加されたMySQLサーバー: クライアント・マシンをNetVault Backupサーバーへ追加する手順については、『Quest NetVault Backupアドミニストレーターズ・ガイド』を参照してください。

i | **重要:** 本書の例の画像および手順では、この2台のマシン環境を使用して、前提条件をすべて満たしていることを前提としています。

プラグインのインストールまたはアップグレード

- 1 **【NetVault 設定ウィザード】** または **【クライアント管理】** ページにアクセスします。

i **メモ**： 選択されたクライアントがすべて同じタイプの場合、設定ウィザードを使用して複数のクライアントに同時にプラグインをインストールできます。複数のクライアントを選択する場合、プラグインのバイナリ・ファイルがターゲット・クライアントの OS とプラットフォームと互換性があることを確認する必要があります。**【クライアント管理】** ページでは、プラグインをインストールするクライアントを 1 つのみ選択できます。

- **【NetVault 設定ウィザード】** ページにアクセスするには：
 - a **【ナビゲーション】**パネルで、**【ガイド付き設定】**をクリックします。
 - b **【NetVault設定ウィザード】**ページで、**【プラグインのインストール】**をクリックします。
 - c 次のページで、利用可能なクライアントを選択します。
 - **【クライアント管理】** ページにアクセスするには：
 - a **【ナビゲーション】**パネルで、**【クライアント管理】**をクリックします。
 - b **【クライアント管理】**ページで、利用可能なマシンを選択して、**【管理】**をクリックします。
 - c **【クライアント表示】**ページで、**【プラグインのインストール】**ボタン(+)をクリックします。
- 2 **【プラグイン・ファイルの選択】** をクリックして、プラグインの .npl インストール・ファイルの場所（インストール用 CD や、Web サイトからファイルをダウンロードしたディレクトリなど）へ移動します。
インストール CD では、このソフトウェアのディレクトリ・パスはオペレーティング・システムによって異なります。
 - 3 ファイル **mys-x-x-x-x.npl**（ここで、xxxxx は、バージョン番号とプラットフォームを示します）を選択し、**【開く】** をクリックします。
 - 4 インストールを開始するには、**【プラグインのインストール】** をクリックします。
プラグインが正常にインストールされると、メッセージが表示されます。

プラグインの削除

- 1 **【ナビゲーション】** パネルで、**【クライアント管理】** をクリックします。
- 2 **【クライアント管理】** ページで、利用可能なクライアントを選択して、**【管理】** をクリックします。
- 3 **【クライアント表示】** ページの **【インストール済みソフトウェア】** テーブルで、**【Plug-in for MySQL】** を選択して **【プラグインのアンインストール】** ボタン (-) をクリックします。
- 4 **【確認】** ダイアログ・ボックスで、**【削除】** をクリックします。

特定 MySQL インスタンスの削除

MySQL インスタンスが正常に構成され、プラグインに追加されたら、これを削除することも可能です。

i | **重要:** この手順を実行する際は、注意が必要です。ただし、「[プラグインの設定](#)」で説明する手順に従って、インスタンスを再追加することも可能です。

- 1 [ナビゲーション] パネルで、[バックアップ・ジョブ作成] をクリックして、次に [セクション] リストの隣にある [新規作成] をクリックします。
- 2 セクション・ツリー内で適切なクライアント・ノードを開きます。
- 3 **Plug-in for MySQL** を開きます。
- 4 該当のインスタンスをクリックして、コンテキスト・メニューから [サーバーを削除] を選択します。
ただし、このコマンドを使用しても確認用ダイアログ・ボックスは特に表示されない点に注意してください。

プラグインの設定

- デフォルト設定の構成
- エラー条件のデフォルト・アクションの設定（オプション）

デフォルト設定の構成

Plug-in for MySQL は、単一の MySQL サーバーについて複数の MySQL インスタンスをサポートしています。各インスタンスは、用途に応じて設定する必要があります。使用可能な設定オプションは、MySQL サーバーで使用されている OS、および **【MySQL Standard/Community】** オプションまたは **【MySQL Enterprise バックアップ】** オプションを使用するかどうかによって異なります。

- 1 [ナビゲーション] パネルで、**【バックアップ・ジョブ作成】** をクリックして、次に **【セレクション】** リストの隣にある **【新規作成】** をクリックします。
- 2 セレクション・ツリー内で適切なクライアント・ノードを開きます。
- 3 **【Plug-in for MySQL】** をクリックして、コンテキスト・メニューから **【新しいサーバーの追加】** を選択します。
- 4 **【設定】** ダイアログ・ボックスで、利用可能なフィールドを記入します。
 - **【MySQL インスタンス名】**（必須）：MySQL インスタンスにつける名前を入力します（デフォルト値として、ローカル・ホスト名が使用されます）。ここで入力した値は、NetVault Backup WebUI で使用されます。Quest では、MySQL インスタンスの名前にはインスタンスが存在するマシンに関連する名前ではなく、一般的な名前を付けることを強く推奨しています。一般的な名前を付けることで、影響を受ける全クライアントに対するポリシーの管理と移植性が向上します。
たとえば、ローカル・ホスト名が `test_mysql_01_machine` で、MySQL インスタンスの推奨名が同じ場合は、インスタンス名を `local_mysql_server` などに変更します。そうすると、各クライアントのインスタンスを設定するときにインスタンス名 `local_mysql_server` を使用できます。したがって、すべてのクライアントのインスタンス名は `local_mysql_server` になります。
 - **【MySQL エディション】**（必須）：利用可能なオプションを選択します。**【MySQL Standard/Community】**、**【MySQL Enterprise バックアップ】**、または **【MariaDB Server】**（MariaDB サーバー）。使用するバージョンにより、このダイアログ・ボックスで変更可能なオプションは異なります。
 - **【ユーザー名】** および **【パスワード】**（必須）：ユーザー名とパスワードを入力します。MySQL インスタンスのデータベース内でバックアップおよびリストアのターゲットとなるテーブルの読み取り / 書き込み権限を持つアカウント（例：管理者権限を持つアカウント）を使用します。
 - i** **重要：** NetVault Backup は、選択されたインスタンスの MySQL データベースにアクセスするたびに、**【ユーザー名】** および **【パスワード】** の値を参照します（バックアップ時およびリストア時の両方）。MySQL インスタンスでこれらの一方の値が変更された場合、これらのフィールドで最新の値に更新する必要があります。更新しない場合、NetVault Backup でインスタンスにアクセスできなくなり、ジョブ要求が適切に実行されません。
 - **【MySQL ベース・ディレクトリ】**（必須）：MySQL インストール・ベース・ディレクトリ（MySQL プログラム・ファイルが保存されているベース・ディレクトリ）のフル・パスを入力します。

- **Windows ベースの MySQL サーバー** : MySQL ベース・ディレクトリは、Windows レジストリで「Location」値を照会すると見つかります。
- **Linux/UNIX ベースの MySQL サーバー** : MySQL をデフォルト・インストールした場合、MySQL ベース・ディレクトリは以下の場所になります。

"/var/lib/mysql"

- **[MySQL Bin ディレクトリ]** : MySQL の実行可能ファイルが保存されている、MySQL サーバー上のディレクトリのフル・パスを入力します。デフォルト・インストールした場合、MySQL ベース・ディレクトリは以下の場所になります。

- **Linux/UNIX ベースの MySQL サーバー** :

"<MySQL ベース・ディレクトリ>/bin"

- **Windows ベースの MySQL サーバー** :

"<MySQL ベース・ディレクトリ>\bin"

i | **メモ** : **mysqlbackup** ユーティリティのデフォルト・パスは、使用している環境と、場所をカスタマイズしたかどうかによって異なります。たとえば、Ubuntu 14 deb パッケージでは、このユーティリティはデフォルトで /usr/bin/mysqlbackup にインストールされます。

- **[Mysqldump パス]** : バックアップおよびリストア・プロセスで使用される **mysqldump** ユーティリティのフル・パスとファイル名を入力します。ターゲット・インスタンスに対して MySQL のデフォルト・インストールを実行している場合は、MySQL サーバー上で使用されている OS に基づいたファイルへの正しいパスがデフォルト値としてこのフィールドに表示されています。

- **Linux/UNIX ベースの MySQL サーバー** :

"<MySQL ベース・ディレクトリ>/bin/mysqldump"

- **Windows ベースの MySQL サーバー** :

"<MySQL ベース・ディレクトリ>\bin\mysqldump"

i | **重要** : [MySQL Binディレクトリ]または[Mysqldumpパス]が上記のデフォルト値に設定されている場合、このフィールドを空白のままにしても構いません。

- **[TCP ポート]** (Windows ベースの MySQL サーバーのみ) : 正しくアクセスするために、MySQL の各インスタンスに独自のポート値を指定します。MySQL のデフォルト・インストールではポート **3306** を使用するため、この値がデフォルトで表示されます。選択したインスタンスに対して別のポートを設定した場合は、適切な値を必ず入力する必要があります。

i | **重要** : 1台のMySQLサーバーに複数のインスタンスが存在する場合、それぞれに独自のポート値が割り当てられるため、この値を[ポート番号]フィールドに入力する必要があります。この値は、各インスタンスのmy.iniファイルで「port=」に表示される値と同じです。

- **[ソケット・ファイル・パス]** (Linux/UNIX ベースの MySQL サーバーのみ) : MySQL ソケット・ファイルのパスとファイル名を入力します。MySQL のデフォルト・インストールを実行した場合は、正しいパスがデフォルト値としてこのフィールドに表示されます。MySQL をデフォルト・インストールした場合、ソケット・ファイルは以下のディレクトリにあります。

"/tmp/mysql.sock"

- i** **重要:** MySQLの標準インストールをMySQLサーバーで実行した場合(デフォルトのインストール先ディレクトリを使用した場合は、[ソケット・ファイル・パス]にはデフォルトのパスをそのまま使用します。ただし、MySQLのインストール時にデフォルト以外のディレクトリを指定した場合は、ソケット・ファイルの正しい場所をこのフィールドに入力します。このパスを特定するには、MySQLサーバーのターミナル・セッションのプロンプトで次のコマンドを実行します。

“ show variables like ‘socket’ ”

[ソケット・ファイル・パス]に正しい値を入力していないと、本プラグインが正常に機能せず、バックアップ/リストアを実行することができません。

- **[デフォルト文字セット]:** デフォルト文字セットには、latin1 が設定されています。エンコーディングに異なる文字セット (UTF-8 など) を設定したい場合は、リストから選択します。

5 **[MySQL Standard/Community]** を使用している場合は、以下のフィールドを記入します。

- **[MyISAM バックアップ方法]:** ご使用の環境において MyISAM ストレージ・エンジンまたはテーブル・タイプが使用されている場合、利用可能なサブオプションを選択します。
 - **[テーブル・ファイルのロック & コピー]** (デフォルト選択): このオプションを使用してテーブル・ファイルのロック、フラッシュ、およびコピーについて標準バックアップ方法を使用することができます。

- i** **ヒント:** このプラグインは、InnoDBテーブルに対してMySQL--single-transactionオプションを使用しますが、MyISAMはこのオプションをサポートしていません。このオプションは、InnoDBなどのトランザクション・テーブルで使用する場合に便利ですが、MyISAMなどの他のタイプのテーブルではこの限りではありません。プラグインがフル・バックアップを完了すると、ターゲットのMySQLインスタンス内のすべてのテーブル・タイプに対して一貫した状態が要求されます。また、バックアップ・ジョブでは、バックアップされた各テーブルに対してmysqldumpが実行されます。つまり、すべてのテーブルが同時にバックアップされるわけではありません。このバックアップ方式により、完全にMySQLインスタンスをリストアする代わりに、必要に応じて、確実に特定のデータベース・オブジェクトを選択してリストアできます。

更新を防ぐためテーブルがロックされていない状態でオンライン・アプローチを使用する場合は、MySQLレプリケーション環境の使用を検討してください。この環境では、データベース・ユーザーはマスターMySQLサーバーと対話します。このサーバー上では、テーブルがロックされることはありません。バックアップはスレーブMySQLサーバーから取得されます。バックアップ・プロセス中、マスター・サーバーからのレプリケーションの更新は、バックアップが完了するまで一時停止されます。レプリケーション・セットアップでのプラグイン使用についての詳細は、「MySQLレプリケーションの使用」を参照してください。

- **[Mysqldump]:** テーブルを多用または頻繁にロードする場合、テーブルをコピーする代わりにこのオプションを選択して Mysqldump ユーティリティを使用します。この設定はパフォーマンスに影響する場合があります。
- **[MySQL レプリケーションを可能にする]:** このインスタンスに対してネイティブのMySQLレプリケーションを有効化するには、このチェック・ボックスを選択します。詳細は、「MySQLレプリケーションの使用」を参照してください。
 - **[スレーブ・インスタンス]:** MySQLレプリケーションを有効にしたインスタンスについて、そのインスタンスをスレーブ・インスタンスとして設定する場合に選択します。
 - **[マスタ・インスタンス]:** MySQLレプリケーションを有効にしたインスタンスについて、そのインスタンスをマスタ・インスタンスとして設定する場合に選択します。

- i** **重要:** レプリケーションの設定を目的としない場合は、このオプションを選択しないよう注意してください。バックアップが失敗します。

- **[特定時点リカバリを可能にする]** : 特定時点バックアップおよびリストアを有効化した場合、このチェック・ボックスを選択します。この機能を使用すると、データ損傷の前または後、あるいはその両方の特定時点へのリカバリが可能になります。この機能を使用する場合は、MySQL バイナリ・ログを有効化します。詳しくは、「[MySQL サーバーでのバイナリ・ログの有効化 \(MySQL Standard/Community オプションのみ\)](#)」を参照してください。
- **[バイナリ・ログ・インデックス・パス]** : **[特定時点リカバリを可能にする]** チェック・ボックスを選択した場合、このフィールドを使用してバイナリ・ログ・インデックス・ファイルへのフル・パスを指定します。MySQL をデフォルト・インストールした場合、このファイルは以下の場所にあります。

- **Linux/UNIX ベースの MySQL サーバー :**

<MySQL ベース・ディレクトリ>/data/< インスタンス名 >-bin.index

- **Windows ベースの MySQL サーバー :**

<MySQL ベース・ディレクトリ>\data\< インスタンス名 >-bin.index

i **重要:** 本プラグインは、[バイナリ・ログ・インデックス・パス]に指定されたファイルが存在するかどうかを設定中に判断します。ただし、指定されたファイルが実際にバイナリ・ログ・インデックスかどうかは、バックアップ・ジョブを開始するまで判断できません。指定されたファイル名が有効なバイナリ・ログ・インデックスではないことが判明した場合、ジョブは適切に実行されないことに注意してください。

- **[リレー・ログ・インデックス・パス]** : [スレーブ・インスタンス] を設定中の場合、このフィールドを利用してリレー・ログ・インデックス・ファイルへのフル・パスを入力し、バックアップに含めます。

- **Linux/UNIX ベースの MySQL サーバー :**

<MySQL ベース・ディレクトリ>/data/< インスタンス名 >-relay-bin.index

- **Windows ベースの MySQL サーバー :**

<MySQL ベース・ディレクトリ>\data\< インスタンス名 >-relay-bin.index

- 6 **[MySQL Enterprise バックアップ]** を使用している場合は、必須の **[Mysqbackup Path]** フィールドに、**mysqbackup** ユーティリティが存在しているディレクトリへの完全パスを入力します。

Linux 環境では、Quest は、**mysqbackup** ユーティリティを実行する代わりに、NetVault Backup スクリプトの .sh ファイルを使用することを推奨します。

たとえば、Linux を使用している場合、.sh ファイルへのデフォルト・パスには次が含まれます。

- MySQL Enterprise バックアップ 3.12 の場合 : /usr/netvault/plugins/mysql/mysqbackup-3.12.sh
- MySQL Enterprise バックアップ 4.0 の場合 : /usr/netvault/plugins/mysql/mysqbackup-4.0.sh

i **メモ :** .sh ファイルの代わりにユーティリティを使用することを選択した場合、**mysqbackup** ユーティリティのデフォルト・パスは、使用している環境と、場所をカスタマイズしたかどうかによって異なることに注意してください。たとえば、Ubuntu 14 deb パッケージでは、このユーティリティはデフォルトで /usr/bin/mysqbackup にインストールされます。

Windows 環境では **mysqbackup** ユーティリティを使用できます。設定を保存するには、**[OK]** をクリックします。

既存インスタンス設定の更新

MySQL のインスタンスを正しく設定し、プラグインに追加したら、以下の手順に従って設定オプションを編集することができます。

- 1 [ナビゲーション] パネルで、[バックアップ・ジョブ作成] をクリックして、次に [セレクション] リストの隣にある [新規作成] をクリックします。
- 2 セレクション・ツリー内で適切なクライアント・ノードを開きます。
- 3 [Plug-in for MySQL] ノードを開きます。
- 4 該当するインスタンスをクリックし、コンテキスト・メニューから [設定] を選択します。

[設定] ダイアログ・ボックスに以前のすべての設定が表示されます。必要に応じてこれらを修正することができます。

i **重要:** [設定] ウィンドウの編集画面では、[MySQLインスタンス名] フィールドは無効になります。これは情報を表示する目的のみのフィールドであり、現在選択されているMySQLインスタンス名が表示されます。

エラー条件のデフォルト・アクションの設定 (オプション)

通常、MySQL バックアップ・ジョブでは、複数のストレージ・エンジン、データベース、テーブルが使用されます。場合によっては、バックアップ・ジョブの実行中に、サポートされていないストレージ・エンジンが検出されたりデータベース/テーブルにアクセスできないことがあります。この場合、一部のアイテムが正しくバックアップされない一方で、バックアップ・ジョブで選択された残りのアイテムについては問題なくバックアップされます。MySQL データベース管理者は、このような状況が発生したときにどのようなアクションを実行するかを決定する必要があります。

- バックアップ・ジョブを警告付きで完了するか、警告なしで完了するか、または失敗とするか
- 正しく完了しなかったアイテムのバックアップを保持するか、削除するか

プラグインでは、バックアップおよびリストア・ジョブのデフォルト・オプションを設定できます。これらのオプションは、ジョブごとに上書きできます。

- 1 [ナビゲーション] パネルで、[設定変更] をクリックします。
- 2 [設定] ページで、利用可能な [サーバー設定] または [クライアント設定] をクリックします。
- 3 [クライアント設定] を選択した場合は、適切なクライアントを選択して、[次へ] をクリックします。
- 4 [NetVault サーバー設定] または [クライアント設定] ページで、[プラグイン・オプション] をクリックします。

[Plug-in for MySQL] セクションには、以下の項目が記載されています。

- **[ロックされたテーブル]** : この条件は、バックアップの対象として選択されているテーブルがプラグイン以外のクライアント・セッションによってロックされているために、バックアップできない場合に満たされます。
 - **[手動で選択されたテーブルが使用不能]** : この条件は、個々のテーブルを、バックアップ・ジョブの定義以降に削除されたなど何らかの理由でバックアップできない場合に満たされます。
 - **[手動で選択されたデータベースが使用不能]** : この条件は、個々のデータベースを、バックアップ・ジョブの定義以降に削除されたなど何らかの理由でバックアップできない場合に満たされます。
 - **[サポートされていないストレージ・エンジン]** : この条件は、バックアップ中に、プラグインでサポートされていないストレージ・エンジン・タイプのテーブルが検出された場合に満たされます。
- 5 これらの各オプションに対し、以下の4つの設定からいずれかを選択できます。
 - **[警告で終了 - 保存セットは保持されます]** : ジョブが [バックアップが警告付きで完了] というステータスを返し、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。

- **【警告なしで完了 — 保存セットは保持されました】** : ジョブが完了し、**【バックアップ完了】** というステータスが返されます。エラーは NetVault Backup バイナリ・ログに記録され、**【ジョブ・ステータス】** ページでは無視されます。バックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
- **【失敗 — セーブセットは保持されます】** : バックアップ・ジョブから **【バックアップ・ジョブ失敗】** というステータスが返されますが、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
- **【失敗 — セーブセットは保持されません】** : バックアップ・ジョブから **【バックアップ・ジョブ失敗】** というステータスが返され、バックアップされたオブジェクトのセーブセットは保持されません。選択された一部のオブジェクトが正しくバックアップされた場合も削除されます。

i | **重要:** 個別のバックアップ・ジョブ・レベルで選択したデフォルト・アクションを選択して上書きすることが可能です。

6 設定を保存するには、**【適用】** をクリックします。

データのバックアップ

- データのバックアップ：概要
- バックアップの実行

データのバックアップ：概要

バックアップを完了する前に、以下のトピックの情報を確認します。

- [MySQL Standard/Community に関する重要注意事項](#)
- [MySQL Enterprise バックアップに関する重要注意事項](#)
- [バックアップ戦略の策定](#)

i **重要:** Questでは、データベース名に特殊文字を使用しないことを強くお勧めします。データベース名に次の文字が含まれている場合、プラグインはそのデータベースをリストアできません。\$ ^ = @ # % +

バックアップ・データはMySQLサーバーから直接NetVault Backupメディア・マネージャにストリーミングされるため、前述の文字を使用するとシーケンス・コマンドとして解釈され、バックアップ・データの整合性に影響を与える場合があります。

MySQL Standard/Communityに関する重要注意事項

[MySQL Standard/Community] オプションを使用する場合、以下のガイドラインと情報について確認する必要があります。

- 半角英数字およびアンダースコアを除くその他すべての文字は、特殊文字として扱われます。
- [MIXED バイナリ・ログ形式の使用](#)

半角英数字およびアンダースコアを除くその他すべての文字は、特殊文字として扱われます。

ハイフンなどの特殊文字を含むデータベースが環境内に存在する場合、以下の制限事項に注意する必要があります。

- データベース名にハイフンが含まれていても、MyISAMを除くすべてのテーブル・タイプがバックアップされます。これらのバックアップは、mysqldump コマンドが常に上記のテーブル・タイプに使用されるためです。
- データベース名にハイフンが含まれると、**MyISAM バックアップ方法**がバージョン 4.2 で導入された **Mysqldump** オプションに設定されている場合、MyISAM テーブルがバックアップされます。この場合、バックアップおよびリストアのパフォーマンスにマイナスとなる影響を及ぼす場合があります。
- **MyISAM バックアップ方法**にデフォルトの **[テーブル・ファイルのロック & コピー]** オプションを使用し、データベース名にハイフンが使用されていると、MyISAM テーブルはバックアップされません。これは、本プラグインがMySQL コマンドを迂回し、直接テーブル・ファイルのコピーを試行するためです。本プラグインは、テーブル・ファイルが特定できないことを示すエラー・メッセージを出力し、バックアップ・ジョブがセーブセットを作成することなく失敗します。

以前のバージョンでは、本プラグインはデータベース・ディレクトリが存在するか検証を試み、失敗すると警告メッセージを出力してから次のデータベースのバックアップを継続していました。この場合、バックアップは警告付きで完了し、その他すべてのデータベースを含むセーブセットを作成します。

Mysqldump オプションを使用したパフォーマンスが最適でないなど、何らかの理由でオリジナルの動作を維持して **[テーブル・ファイルのロック & コピー]** オプションを使用する場合、本プラグイン設定ファイル「**nvmysql.cfg**」の **ValidateDatabaseDirectory** パラメータを以下のように手動で「**TRUE**」に設定することができます。

```
[MySQL:ValidateDatabaseDirectory]
Value=TRUE
```

その後、新規動作を適用しようとした場合、「**nvmysql.cfg**」ファイルでこのパラメータを「**FALSE**」に変更するか削除することができます。

- データベースまたはデータベース・オブジェクト（テーブルやビューなど）を作成するための SQL ステートメントが含まれる増分または差分バックアップをリストアすると、該当するデータベースまたはデータベース・オブジェクトが存在する場合に失敗します。この問題を解決するには、1 つまたは複数のデータベースまたはデータベース・オブジェクトを作成または削除した後にフル・バックアップを実行します。この手順により、後続の増分バックアップまたは差分バックアップに **CREATE** または **DROP SQL** ステートメントが含まれなくなります。

MIXED バイナリ・ログ形式の使用

MIXED バイナリ・ログ形式が使用されている場合、MySQL は **USE** ステートメントの使用を強制しません。したがって、Quest では、すべてのデータベース・ユーザーとプログラムで変更されたテーブルが **USE** によって選択されたデータベース内に存在すること、およびクロスデータベースの更新が実行されないことを確認することをお勧めします。このガイドラインがご使用の環境に適していない場合は、Quest では、MIXED バイナリ・ログ形式を使用しないことをお勧めします。

- i** | **重要:** 増分および差分バックアップ・ジョブは、MIXED バイナリ・ログ形式が使用されている場合、警告を表示して完了します。

ご使用の環境で MIXED バイナリ・ログ形式が使用されている場合、PIT リカバリ中にバイナリ・ログ・エントリが再生されないことがあります。リカバリ中、プラグインは **--database** オプションを指定して **mysqlbinlog** を使用し、リストア・ジョブで選択したデータベースに関連するエントリだけを再生します。**--database** が使用されない場合、すべてのエントリが再生され、すべてのデータベースに影響します。MIXED バイナリ・ログ形式を使用する場合、**--database** オプションを使用した **mysqlbinlog** が一部またはすべてのエントリを再生しないようにエントリが書き込まれます。詳細は、[「https://dev.mysql.com/doc/refman/5.7/en/mysqlbinlog.html#option_mysqlbinlog_database」](https://dev.mysql.com/doc/refman/5.7/en/mysqlbinlog.html#option_mysqlbinlog_database) を参照してください。

--database オプションを使用した MIXED バイナリ・ログ形式が正しく機能するためには、データベースを選択する **USE** ステートメントの下で、データベースに対する特定の更新のすべてのトランザクションを実行する必要があります。

増分または差分バックアップがリストアされず、**mysqlbinlog** が MySQL サーバーからの現在のバイナリ・ログを適用した場合も、同じ状況が発生します。この状況は、バイナリ・ログがバックアップに保存された方法ではなく、バイナリ・ログが書き込まれた方法によって発生します。

- i** | **重要:** 変更するテーブルが、**USE** ステートメントで指定されたデータベースに属していれば、MySQL コマンド・プロンプトで生成されたトランザクションに適用されます。また、MySQL サーバーのデータベースと対話するスクリプト、プログラム、その他のアプリケーションによって生成されたトランザクションにも適用されます。

次の例では、MIXED がリカバリ動作にどのように影響するかをさまざまな状況ごとに示しています。

- **例 1:** この例では、データ行が **my_database** の **my_table** に挿入されます。**USE** ステートメントはないため、使用中のデータベースは **mysql** データベースなどのデフォルト・データベースです。**binlog_format** が MIXED に設定されている場合、**mysqlbinlog** がバイナリ・ログに **--database my_database** オプションを適用すると、次のトランザクションが再生されません。

```
-bash-$ mysql
mysql> insert into my_database.my_table (C1,C2) values(1,now());
Query OK, 1 row affected (0.01 sec)
```

- **例 2** : この例では、データ行が **my_database** の **my_table** に挿入されます。USE ステートメントは存在しますが、別のデータベースが指定されています。つまり、USE ステートメントで **my_databas** が選択されていません。binlog_format が MIXED に設定されている場合、**mysqlbinlog** がバイナリ・ログに **-database my_database** オプションを適用すると、次のトランザクションが再生されません。

```
-bash-$ mysql
mysql> use mysql
Database changed
mysql> insert into my_database.my_table (C1,C2) values(2,now());
Query OK, 1 row affected (0.04 sec)
```

- **例 3** : この例では、データ行が **my_database** の **my_table** に挿入され、USE ステートメントで **my_database** が選択されています。binlog_format が MIXED に設定されている場合、**mysqlbinlog** がバイナリ・ログに **-database my_database** オプションを適用すると、次のトランザクションが再生されます。

```
-bash-$ mysql
mysql> use my_database
Database changed
mysql> insert into my_database.my_table (C1,C2) values(3,now());
Query OK, 1 row affected (0.04 sec)
```

- **例 4** : この例では、2つの挿入クエリがあります。最初の挿入は、USE ステートメントで選択されたデータベースとは異なる **my_database** に対して行われます。2番目の挿入は、USE ステートメントで選択された **my_database** の範囲内で行われます。binlog_format が MIXED に設定されている場合、**my_database** が USE ステートメントで指定されていないため、最初の挿入は再生されません。しかし、2番目の挿入は USE ステートメントで **my_database** が指定されているため、再生されます。

```
-bash-$ mysql
mysql> use mysql
Database changed
mysql> insert into my_database.my_table (C1,C2) values(4,now());
Query OK, 1 row affected (0.01 sec)
mysql> use my_database
Database changed
mysql> insert into my_database.my_table (C1,C2) values(5,now());
Query OK, 1 row affected (0.04 sec)
```

MySQL Enterprise バックアップに関する重要注意事項

[MySQL Enterprise バックアップ] オプションを使用する場合、以下のガイドラインと情報について確認する必要があります。

- MySQL は、重要なデータには InnoDB テーブルを使用することを推奨しています。これは、バックアップ・プロセスが高速で、信頼性および拡張性が高いためです。MySQL Enterprise バックアップは MySQL テーブルについてさまざまなバックアップを提供し、InnoDB テーブルに関するバックアップを最適化します。このオプションは、すべての InnoDB テーブルのホット・バックアップを実行します。データベースの実行中においてもホット・バックアップが実行されるため、バックアップ中に現行データベース操作を停止する必要はありません。さらに、バックアップ・プロセス中にデータベースに加わった変更が含まれます。この動作は、環境内のデータベースをオンラインに保ち、その拡張性もサポートされている必要がある場合、バックアップの完了に必要な時間に影響を及ぼすため重要です。
- このオプションを使用すると、MyISAM テーブルとその他の非 InnoDB テーブルは、ウォーム・バックアップを使用して、最後にバックアップされます。ウォーム・バックアップにおいて、データベースは稼働し続けますが、バックアップが完了してもテーブルは読み込み専用アクセスに設定されます。
- ホット・バックアップ・フェーズにおいてデータの大半がバックアップされたか確認したら、InnoDB を新規テーブル用デフォルト・ストレージ・エンジンとし、既存のテーブルを変換して InnoDB ストレージ

ジ・エンジンで利用可能にすることを検討するようお勧めします。MySQL Server 5.5 以降では、InnoDB がデフォルトです。

- 増分バックアップは主に InnoDB テーブル用であり、非 InnoDB テーブルは読み取り専用、あるいは、それほど頻繁な更新を必要としない用途に向いています。InnoDB ファイルについて、最後のバックアップ以降に変更が起こった場合、ファイル全体が含まれます。
- 本プラグインを使用する場合、以下の条件に適合すると、MySQL インスタンス内のすべての InnoDB テーブルがバックアップされます。

- ただし、テーブルがバックアップ用に明示的に選択され、テーブルが InnoDB ストレージ・エンジンまたはその類ではない場合に限りです。

例:2つのデータベースを含むMySQLインスタンスが配置されています(DB1およびDB2)。各データベースには2つのテーブルが含まれます。DB1にはT1_InnoDBとT1_MyISAM、DB2にはT2_InnoDBとT2_MyISAMが含まれます。T1_MyISAMとT2_MyISAMをバックアップすると、T1_InnoDBとT2_InnoDBもバックアップに含まれます。InnoDBテーブルの1つを含めると、InnoDBテーブルのみがバックアップされます。データベースの1つを選択すると、データベース内のテーブルのみがバックアップされます。

- いくつかまたはすべてのデータベースがバックアップに選択されると、すべての関連 InnoDB テーブルがバックアップから除外されます。

例:2つのデータベースを含むMySQLインスタンスが配置されています(DB1およびDB2)。各データベースには2つのテーブルが含まれます。DB1にはT1_InnoDBとT1_MyISAM、DB2にはT2_InnoDBとT2_MyISAMが含まれます。DB1とDB2をバックアップし、T1_InnoDBとT2_InnoDBを除外すると、T1_InnoDBとT2_InnoDBもバックアップに含まれます。2つのInnoDBテーブルのうち1つだけを除外すると、InnoDBテーブルのみがバックアップされます。

この説明は、MySQL Enterprise バックアップ (`mysqlbackup` ユーティリティ) の現在の動作を表していますが、MySQL の将来のリリース (3.12 以降) では変更される可能性があります。

- MySQL 5.6 以降では、`innodb_file_per_table` 設定オプションはデフォルトで有効化されています。`innodb_file_per_table` オプションが無効化された状態で作成されたすべての InnoDB テーブルは、InnoDB システム・テーブルスペース内に格納されますが、バックアップから除外することはできません。InnoDB テーブルをテーブルスペース外に配置する必要がある場合、`innodb_file_per_table` オプションを有効化した状態で、MySQL 内で InnoDB テーブルを作成する必要があります。各 `.ibd` ファイルには、1つのテーブルのデータとインデックスのみが含まれます。

バックアップ戦略の策定

MySQL バックアップ戦略を定義する際、以下の点を明確にしておきます。

- **[MySQL Standard/Community]** オプションまたは **[MySQL Enterprise バックアップ]** オプションのどちらを使用するか。環境に両方のバージョンを導入している場合でも、プラグインでは1つの計画のみを使用することができます。MEB ベース方法または `mysqldump` ベース方法のいずれかを使用します。両方を併用することはできません。

MEB ベース・オプションを使用すると、バックアップに選択したすべてのデータベース・オブジェクトに対して `mysqlbackup` ユーティリティまたは適用可能な NetVault Backup スクリプトが1度実行され、ジョブ・ログ内に `mysqlbackup` ログが出力されます。データのバックアップには2つのステージが含まれます。最初のステージでは、すべての InnoDB テーブルがコピーされます。2番目のステージでは、すべてのテーブル・タイプがコピーされます。InnoDB テーブルのホット・バックアップをサポートするだけでなく、MEB ベース・オプションはバックアップ・パフォーマンスを向上させます。

`mysqldump` ベース・オプションを使用する場合、各テーブル、トリガ、およびストアド・プロシージャに対してコマンドが実行されます。ホット・バックアップはサポートされません。

- フル・バックアップ時にインスタンス全体の読み取り専用アクセスが必要なことを認識した上で、フル・バックアップをどのような頻度で実行するか

- バックアップの速度とリストアの速度のどちらを重視するか
- 許容できる最大データ損失量

上記の点を明確にしておくこと、実装するバックアップ・タイプおよび頻度を定義する際に役立ちます。

- [MySQL Standard/Community 用バックアップ・タイプの確認](#)
- [MySQL Enterprise バックアップ用バックアップ・タイプの確認](#)
- [MySQL Standard/Community 用バックアップ・シーケンス例](#)

MySQL Standard/Community用バックアップ・タイプの確認

[MySQL Standard/Community] オプションを使用する場合、本プラグインは `mysqldump` を使用して以下のタイプのバックアップを実行します。

- フル・バックアップ
- 増分バックアップ
- 差分バックアップ
- 個々のデータベース/テーブル・コピーのみのバックアップ
- データベース全体のコピー・バックアップ

各 MySQL インスタンスのデータ保護要件に適したバックアップ・シーケンスを選択するには、まずこれらのバックアップの違いを理解する必要があります。

フル・バックアップ

[MySQL Standard/Community] オプション用フル・バックアップの場合、本プラグインは `mysqldump` ユーティリティを使用して、インスタンスに含まれるすべてのデータベースをバックアップします。フル・バックアップはほぼすべてのリストア・シナリオの起点になるため、あらゆるバックアップ戦略の基盤となります。プラグインで生成されたフル・バックアップを使用して、インスタンス全体、個々または複数のデータベース、個々または複数のテーブルをリストアできます。

[フルまたはインクリメンタル・バックアップ後、バイナリ・ログをパージ] オプションを使用して、フルまたは増分バックアップ後にバイナリ・ログをパージする必要があります。[MySQL レプリケーションを可能にする] が無効化され、[特定時点リカバリを可能にする] が有効化されている標準の MySQL サーバー設定でプラグインを使用する場合、このオプションはデフォルトで有効化されています。プラグインをクラスタに接続すると、このオプションは無効化されます。バイナリ・ログのパージはプラグインの外で管理する必要があります。

i **重要:** NetVault Backupサーバーがクラスタ化されたMySQLサーバーと標準のMySQLサーバー両方を管理するような混合環境では、標準のMySQLサーバー用に作成したバックアップ・オプション・セットをMySQLベースのクラスタで再利用しないでください。

[フルまたはインクリメンタル・バックアップ後、バイナリ・ログをパージ] オプションを選択しない場合、プラグインは設定ファイルで**最終バックアップのログ**を追跡します。これにより必要に応じてバイナリ・ログを手動でパージすることができます。例えば、MySQL レプリケーション環境で、バイナリ・ログがスレーブ・インスタンスにレプリケートされるまで、バイナリ・ログをマスタ・インスタンスからパージしたくない場合などは、ユーザーはバイナリ・ログを手動でパージする必要があります。

増分バックアップ

増分バックアップでは、最後のフルまたは増分バックアップ以降に生成されたバイナリ（トランザクション）・ログをバックアップし、次にバイナリ・ログをパージします。バイナリ・ログはインスタンスに基づくため、すべてのデータベースのトランザクション・ログがまとめてバックアップされ、パージされます。

増分バックアップはメディア障害またはデータ損傷の発生後のデータ損失を低減する上で重要です。増分バックアップを使用すると、不正な更新やテーブルの削除などのデータ損傷の前および後の時点にリストアできます。フル・バックアップとは異なり、増分バックアップはバックアップ中に読み取り専用アクセスを必要としません。

MySQL の増分バックアップを実行するには、バイナリ・ログを有効にする **-log-bin** オプションを使用して MySQL インスタンスを開始する必要があります。この手順は、「[MySQL サーバーでのバイナリ・ログの有効化 \(MySQL Standard/Community オプションのみ\)](#)」で概説しています。詳しくは、『MySQL リファレンス・マニュアル』のバイナリ・ログに関するセクションを参照してください。

上記で説明したように、**[フルまたはインクリメンタル・バックアップ後、バイナリ・ログをパージ]** オプションを使用して、フルまたは増分バックアップ後にバイナリ・ログをパージする必要があります。このオプションを使用しない場合は、本プラグインは設定ファイルで**最終バックアップのログ**を追跡します。これにより必要に応じてバイナリ・ログを手動でパージすることができます。

差分バックアップ

差分バックアップでは、最後のフルまたは増分バックアップ以降に生成されたバイナリ（トランザクション）・ログをバックアップします。ただし、この形式のバックアップでは、完了時にバイナリ・ログが**パージされません**。このため、以降の差分バックアップのサイズが大きくなり、その時間も長くなります。各差分バックアップには、前の差分バックアップにも含まれていたバイナリ・ログだけでなく、前の差分バックアップ以降に生成されたバイナリ・ログも含まれることになるため、サイズが大きくなり、その時間も長くなります。たとえば、月曜日から金曜日までの差分バックアップを伴って、日曜日にフル・バックアップの実行がスケジュールされている場合、月曜日の差分には日曜日のフル・バックアップ以降生成されたトランザクション・ログ・ファイルが含まれます。一方、火曜日の差分には、月曜日に生成されたバイナリ・ログ・ファイルおよび火曜日に生成されたバイナリ・ログ・ファイルが含まれます。水曜日の差分バックアップには、月曜日、火曜日、および水曜日のバイナリ・ログが含まれる、というようになります。

増分バックアップと同様に、差分バックアップを使用すると、メディア障害またはデータ損傷が発生した場合のデータ損失を低減でき、障害/損傷の前および後の時点にリストアできます。フル・バックアップとは異なり、差分バックアップはバックアップ中に読み取り専用アクセスを必要と**しません**。

差分バックアップを実行するには、バイナリ・ログを有効にする **-log-bin** オプションを使用して MySQL インスタンスを開始する必要があります。この手順は、「[MySQL サーバーでのバイナリ・ログの有効化 \(MySQL Standard/Community オプションのみ\)](#)」で概説しています。詳しくは、『MySQL リファレンス・マニュアル』のバイナリ・ログに関するセクションを参照してください。

増分バックアップと差分バックアップの比較

増分バックアップでは、バイナリ・ログがバックアップ後にパージされ、最後の増分バックアップ後に作成されたバイナリ・ログのみがバックアップされるため、以降の増分バックアップの実行時間は短くなります。ただし、増分バックアップを使用するリストア・シーケンスでは、フル・バックアップから障害時点までに実行されたすべての増分バックアップを継続してリストアする必要があります。このため、複数のリストア・ジョブを開始するためにデータベース管理者に必要な操作が多くなり、このプロセスではリストアに長い時間がかかる可能性があります。

差分バックアップでは、バイナリ・ログがバックアップ後にパージされず、最後のフル・バックアップ後に作成されたすべてのバイナリ・ログがバックアップの対象となるため、以降の各差分バックアップの実行時間は長くなります。ただし、差分バックアップを使用するリストア・シーケンスでは、フル・バックアップのリストア後に差分バックアップを1つのみリストアするだけで済みます。このため、リストア・プロセスに必要なデータベース管理者の操作が少なくなり、このプロセスではリストア時間は短くなります。

個々のデータベース/テーブル・コピーのみのバックアップ

場合によっては、データベース全体の包括的なバックアップおよびリストア手順に影響を与えることなく、特殊な目的でバックアップを実行しなければならないことがあります。たとえば、バックアップをテスト環境のソースにしたり、レプリケーション・スレーブ・インスタンスの初期同期用に使用する場合があります。個々のデータベース/テーブル・コピーのみのバックアップは、このような特殊な目的のために設計されており、MySQL 環境をコピーすることができます。**コピーのみのバックアップ**は、設定されたバックアップ・シーケンスから独立しているため、フル、増分、または差分バックアップのリカバリ可能性には影響しません。ただし、フル・バックアップの代わりとして使用することは**できません**。

データベース全体のコピー・バックアップ

個々のデータベース/テーブル・コピーのみのバックアップで説明したように、**[データベース全体のコピー・バックアップ]** オプションは、選択されたデータベースに対応するすべての InnoDB テーブルを含む、MySQL

データベースのコピーを作成するため、特殊な目的にのみ使用されます。**コピー・バックアップ**は、設定されたバックアップ・シーケンスから独立しているため、フル、増分、または差分バックアップのリカバリ可能性には影響しません。ただし、フル・バックアップの代わりとして使用することは**できません**。

i | **重要:** このオプションは、選択したデータベースのすべてのテーブルがInnoDBテーブルである場合にのみ使用できます。

個々のデータベース/テーブル・コピーのみのバックアップとデータベース全体の コピー・バックアップ

選択したデータベースごとに、データベースのテーブルが1つだけ選択されていても、[データベース全体のコピー・バックアップ] オプションによってデータベース全体がバックアップされます。このオプションでは、バックアップする個々のデータベースを選択できますが、個々のテーブルを選択することはできません。また、このオプションでは、InnoDB テーブルのバックアップのみがサポートされます。

[個々のデータベース/テーブル・コピーのみのバックアップ] オプションでは、個々のデータベースと個々のテーブルを選択できます。また、バックアップには InnoDB および MyISAM テーブルを含めることができます。ただし、通常、バックアップは [データベース全体のコピー・バックアップ] オプションのほうが、[個々のデータベース/テーブル・コピーのみのバックアップ] オプションよりも早く完了します。

MySQL Enterpriseバックアップ用バックアップ・タイプの確認

[MySQL Enterprise バックアップ] オプションについて、本プラグインは選択されたすべてのデータベース・オブジェクトに対して `mysqlbackup` コマンドを1度実行し、フルおよび増分タイプのバックアップをアーカイブします。フル、増分、および TTS。

フル・バックアップ

[MySQL Enterprise バックアップ] オプション用フル・バックアップの場合、本プラグインは `mysqlbackup` ユーティリティまたは適用可能な NetVault Backup スクリプトを使用して、インスタンスに含まれる**選択されたすべてのデータベース・オブジェクト**をバックアップします。フル・バックアップはほぼすべてのリストア・シナリオの起点になるため、あらゆるバックアップ戦略の基盤となります。プラグインで生成されたフル・バックアップを使用して、インスタンス全体、個々または複数のデータベース、個々または複数のテーブルをリストアできます。

増分バックアップ

InnoDB テーブルについて、最後のフルまたは増分バックアップ以降に変更が加わったデータのみがバックアップされます。非 InnoDB テーブルの場合、最後のフルまたは増分バックアップ以降に何かテーブル内で変更された場合、テーブル全体がバックアップされます。

転送可能なテーブルスペース(TTS)バックアップ

TTS バックアップを行う場合、プラグインはフル・バックアップを実行し、`--use-tts` MySQL オプションを追加します。

i | **重要:** Questでは、バックアップ・プランとは別に、TTSバックアップをスタンドアロン・バックアップとしてのみ生成することを強く推奨します。TTSバックアップは部分バックアップであるため、フル・バックアップまたは増分バックアップ戦略の置き換えや補完に使用することはできません。また、ディザスタ・リカバリ操作に使用することもできません。

TTS バックアップを生成する場合は、次の制限事項に注意する必要があります。

- TTS バックアップは、MySQL Server 5.6 以降でのみサポートされます。
- バックアップには、InnoDB テーブルのみが含まれます。
- バックアップには、`innodb_file_per_table` オプションを有効にして作成されたテーブルのみが含まれません。

- 共有テーブルスペースでパーティションが作成された場合、パーティション・テーブルのバックアップは失敗します。
- バックアップには、バイナリ・ログまたはリレー・ログは含まれません。

--use-tts オプションの使用に関するその他の制限については、<https://dev.mysql.com/doc/mysql-enterprise-backup/4.0/en/backup-partial-options.html> を参照してください。

MySQL Standard/Community用バックアップ・シーケンス例

以下に、さまざまなバックアップ・シーケンスを示します。

- **フル・バックアップのみ**：ビジネス要件で前日までのデータ保護が保証されており、読み取り専用アクセスが毎日許容される場合（勤務時間外など）、フル・バックアップのみを毎日実行すれば十分でしょう。データベース管理者がリカバリできるのは、データベースの最後のフル・バックアップ時点までですが、MySQL サーバーに存在しているバイナリ・ログを使用して PIT リカバリを実行することができます。
- **フル・バックアップと増分バックアップ**：ビジネス要件で前日までのデータ保護が保証されているが、ターゲット MySQL インスタンスへの読み取り専用アクセスが断続的にのみ許容され（週 1 回または隔週の勤務時間外など）、**バックアップ時間をできる限り短縮する必要がある場合**、フル・バックアップと増分バックアップの組合せが最適です。たとえば、毎週日曜日の夜 11:00 にフル・バックアップが実行され、さらに月曜日から土曜日の午後 11:00 にトランザクション・ログ・ファイルのバックアップが実行されているとします。この場合、各増分バックアップには、前夜のバックアップ、すなわち日曜日の夜に実行されたフル・バックアップまたはいずれかの増分バックアップ以降に生成されたバイナリ・ログが含まれます。

このバックアップ・タイプ・シーケンスのリストアには、より長い時間がかかることに注意してください。たとえば、火曜日にリカバリを実行する場合、日曜日のフル・バックアップと月曜日の増分バックアップをリストアする必要があります。また、木曜日にリカバリを実行する場合は、日曜日のフル・バックアップに続いて、月曜日、火曜日、および水曜日の増分バックアップをリストアする必要があります。バックアップ時間は短くなりますが、複数のリストア・ジョブを実行するために必要な操作が多くなるため、リストア時間は長くなる可能性があります。

- **フル・バックアップと差分バックアップ**：ビジネス要件で前日までのデータ保護が保証されているが、ターゲット MySQL インスタンスへの読み取り専用アクセスが断続的にのみ許容され（週 1 回または隔週の勤務時間外など）、**リストア時間をできる限り短縮する必要がある場合**、フル・バックアップと差分バックアップの組合せが最適です。たとえば、フル・バックアップを毎週日曜日の夜 11:00 に実行し、差分バックアップを月曜日から土曜日の午後 11:00 に実行します。各差分バックアップには、最後のフル・バックアップ以降に生成されたバイナリ・ログが含まれます。前述のように、このプロセスでの全体的なバックアップ時間は長くなります。

リカバリする必要のある特定時点に関わらず、必要なリストア・ジョブの数は同じです。たとえば、火曜日にリカバリを実行する場合、日曜日のフル・バックアップと月曜日の差分バックアップをリストアする必要があります。また、木曜日にリカバリを実行する場合は、日曜日のフル・バックアップに続いて水曜日の差分バックアップをリストアする必要があります。以降の差分バックアップはサイズが大きくなり、時間も長くなりますが、実行する必要のあるリストア・ジョブの数は少なくなるため、リストア時間は短くなります。

バックアップの実行

Plug-in for MySQL を使用してバックアップを実行するには、以下のトピックで説明する手順に従います。

- [バックアップ対象データの選択](#)
- [バックアップ・オプションの設定](#)
- [ジョブのファイナライズと実行](#)

バックアップ対象データの選択

バックアップ・ジョブを作成するには、セット（バックアップ・セレクション・セット、バックアップ・オプション・セット、スケジュール・セット、ターゲット・セット、および詳細設定セット）を使用する必要があります。

バックアップ・セレクション・セットは、増分および差分バックアップに必要です。フル・バックアップを実行中にバックアップ・セレクション・セットを作成してから、フル、増分、差分バックアップに使用します。増分または差分バックアップにセレクション・セットが使用されていない場合、バックアップ・ジョブがエラーをレポートします。詳しくは、『Quest NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

i | **ヒント:** 既存のセットを使用するには、[バックアップ・ジョブ作成]をクリックして、[選択]リストからセットを選択します。

1 [ナビゲーション] パネルで、[バックアップ・ジョブ作成] をクリックします。

[ガイド付き設定] リンクからウィザードを開始することもできます。[ナビゲーション] パネルで、[ガイド付き設定] をクリックします。[NetVault 設定ウィザード] ページで、[バックアップ・ジョブ作成] をクリックします。

2 [ジョブ名] に、ジョブの名前を指定します。

ジョブの進捗状況の監視やデータのリストアップ時にジョブを識別しやすくするため、分かりやすい名前を割り当てます。ジョブ名には英数字と英数字以外の文字を使用できませんが、非ラテン語系の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

3 [選択] リストの隣にある、[新規作成] をクリックします。

4 プラグインのリストで [Plug-in for MySQL] を開くと、MySQL サーバが表示されます。

5 利用可能データを選択します。

- バックアップ・ジョブ内で選択したインスタンスからすべての MySQL データベースを選択するには、[全てのデータベース] ノードを選択します。
- より細かく選択するには、[全てのデータベース] ノード内のデータベースを個別に表示させます。さらに、個々のデータベースを開き、その中のテーブルを表示して個別に選択し、バックアップ・ジョブに追加することもできます。
- アイテムをバックアップから明示的に除外するには、親レベルのアイテムを選択し、子のアイテムを個別にクリックして、緑のチェックマークを赤色の X に変更し、除外対象に指定します。

i | **重要:** [MySQL Standard/Community] オプションを使用したバックアップに細かいレベルのデータ・セットを選択する場合、[バックアップ・オプション] タブでバックアップ・タイプに [個々のデータベース/テーブルのコピーのみ] を選択します。他の形式のバックアップ (フル、増分、または差分バックアップ) を選択した場合、細かいレベルの選択は無視されデータベース全体がバックアップされます。MySQL 5.5.x 以降の場合、スタッド・プロシージャ、関数、およびトリガは、[MySQL Standard/Community] オプションの [フル] および [個々のデータベース/テーブルのコピーのみ] で自動的にバックアップされます。

MySQL 5.5.x (以降) では、選択ツリーに「information_schema」データベースが表示されますが、このデータベースは選択できません。この問題が発生するのは、このデータベースに含まれるすべてのデータが動的に生成され、永久的に存在するものではないからです。このため、プラグインは、すべてのバックアップから information_schema データベースを自動的に除外します。

6 [保存] をクリックして、[新規セットの作成] ダイアログ・ボックスに名前を入力し、[保存] をクリックします。

名前には英数字と英数字以外の文字を使用できませんが、非ラテン語系の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

バックアップ・オプションの設定

次の手順には、バックアップ・オプション・セットの作成または既存のセットの選択が含まれています。**[MySQL Standard/Community]** または **[MySQL Enterprise バックアップ]** オプションのいずれを使用するかによって、**[バックアップ・オプション]** タブで各設定が利用可能になります。

MySQL Standard/Community用バックアップ・オプションの設定

バックアップする目的のアイテムを選択した状態で、実行するバックアップ・タイプを選択したり、失敗した場合に異なる動作を選択したりすることもできます。

i | **ヒント:** 既存のセットを使用するには、**[プラグイン・オプション]** リストで使用するセットを選択します。

- 1 **[プラグイン・オプション]** リストの隣にある、**[新規作成]** をクリックします。
- 2 利用可能なオプションを選択します。

i | **重要:** MySQLのターゲット・インスタンスとしてレプリケーションのマスター・インスタンスを指定した場合(このMySQLインスタンスの**[設定]**ダイアログ・ボックスで、**[MySQLレプリケーションを可能にする]**オプションと**[マスター・インスタンス]**オプションが選択されている状態)、フル、増分および差分形式のバックアップは選択することはできません。詳細は、「**MySQLレプリケーションの使用**」を参照してください。

- **[全てのデータベースをフルバックアップ]** (デフォルト選択) : 現在の MySQL インスタンスに含まれるすべてのデータベースおよびすべてのテーブルについて完全なフル・バックアップを実行するには、このオプションを選択します。このオプションを選択すると、バックアップ・セレクション・セットで選択したデータに関係なく、すべてがバックアップされます。
- **[増分バックアップ]** : 最後のフルまたは増分バックアップ以降に生成されたトランザクション・ログのみをバックアップする場合、このオプションを選択します。
- **[差分バックアップ]** : 最後のフルまたは増分バックアップ以降に生成されたすべてのトランザクション・ログをバックアップする場合、このオプションを選択します。以降の差分バックアップには、元のフル・バックアップの実行後に生成されたすべてのバイナリ・ログが含まれます。バックアップ完了後は、MySQL インスタンスのバイナリ・ログは**保持**されます。

i | **重要:** 増分および差分バックアップ・ジョブは、MIXEDバイナリ・ログ形式が使用されている場合、警告を表示して完了します。詳細は、「**MIXEDバイナリ・ログ形式の使用**」を参照してください。

! | **注意:** MIXEDバイナリ・ログ形式が使用されている場合、バイナリ・ログ・エントリがPITRリカバリ中に再生されないことがあるため、リカバリ用に選択したデータベースが、選択したポイントにロールバックされないことがあります。詳細は、「**MySQL Standard/Communityに関する重要注意事項**」および「**MIXEDバイナリ・ログ形式の使用**」を参照してください。

- **[個々のデータベース / テーブル・コピーのみ]** : 特殊な目的 (テスト環境を作成する場合など) で MySQL 環境をコピーする場合に、このオプションを選択します。データベースの全体的なバックアップおよびリストア手順には影響を与えません。この方法で作成したコピー・バックアップは、MySQL のフル・バックアップと増分 / 差分バックアップを併用したシナリオで設定されたシーケンスに影響しません (これらのバックアップは、バイナリ・ログに影響を与えません)。この形式のバックアップは、MySQL のフル・バックアップと増分 / 差分バックアップの併用シナリオで通常のバックアップ・シーケンスとして設定されたバックアップから独立しています。また、コピー・バックアップをフル・バックアップの代わりに使用することは**できません**。
- **[データベース全体のコピー・バックアップ]** : MySQL 環境の特殊な目的のコピーを作成して、選択したデータベースのすべてのコンテンツをバックアップするには、このオプションを選択します。たとえば、テスト環境を構築するためにこのオプションを選択します。このオプションは、

[個々のデータベース/テーブル・コピーのみ] オプションと似ていますが、このオプションでは、選択したすべてのデータベースと対応するテーブルがバックアップされます。このオプションは、選択したデータベースのすべてのテーブルが InnoDB テーブルである場合にのみ使用できます。この方法で作成したコピー・バックアップは、MySQL のフル・バックアップと増分/差分バックアップを併用したシナリオで設定されたシーケンスに影響しません（これらのバックアップは、バイナリ・ログに影響を与えません）。また、コピー・バックアップをフル・バックアップの代わりに使用することは **できません**。

i **重要:** このオプションは、選択したデータベースのすべてのテーブルが InnoDB テーブルである場合にのみ使用できます。

このオプションを選択すると、選択したデータベース全体および対応するすべてのテーブルがバックアップされます。これは、バックアップする特定のテーブルを選択した場合でも発生します。特定のテーブルをバックアップする場合は、[個々のデータベース/テーブル・コピーのみ] オプションを使用します。

- **[バックアップ中は、全てのテーブルを読み取りアクセスにロックしトランザクションの消失を防ぎます。]: [フルバックアップ]** を選択し、現在読み取り専用アクセスのインスタンス内のすべてのデータベースがロックされることによるトランザクションの消失を防ぎたい場合に、このオプションを選択します。このオプションを選択した場合、ユーザーはフル・バックアップ中にインスタンス全体でデータの挿入、更新、または削除を実行できません。このオプションが選択解除されると、プラグインは、テーブルがバックアップされた場合に **限り** バックアップ・プロセス中に各テーブルをロックします。このため、Quest は、インスタンスに関連するテーブルが含まれる場合、このオプションを選択してバックアップ・プロセス中に確実にすべてのテーブルがロックされるようにすることをお勧めします。
- **[フルまたはインクリメンタル・バックアップ後、バイナリ・ログをパージ]: [MySQL レプリケーションを可能にする]** が無効化され、**[特定時点リカバリを可能にする]** が有効化されている標準の MySQL サーバー設定でプラグインを使用する場合、このオプションはデフォルトで有効化されています。プラグインをクラスタに接続すると、このオプションは無効化されます。バイナリ・ログのパージはプラグインの外で管理する必要があります。Quest はこのオプションを使用することをお勧めしますが、ユーザーはバイナリ・ログに関する制御の度合いを決定することが可能です。

i **重要:** NetVault Backup サーバーがクラスタ化された MySQL サーバーと標準の MySQL サーバー両方を管理するような混合環境では、標準の MySQL サーバー用に作成したバックアップ・オプション・セットを MySQL ベースのクラスタで再利用しないでください。

- 3 各条件に対して利用可能なアクションを選択します（詳しくは、「**エラー条件のデフォルト・アクションの設定 (オプション)**」を参照してください）。

各条件を使用することにより、ジョブに対して実行するアクションを選択することができます。これらとは異なるデフォルト・アクションを選択していたとしても、現在のジョブについてこの手順で指定した値に自動的に設定されます。

- **[警告で終了 — 保存セットは保持されます]:** ジョブが **[バックアップが警告付きで完了]** というステータスを返し、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
- **[警告なしで完了 — 保存セットは保持されました]:** ジョブが完了し、**[バックアップ完了]** というステータスが返されます。エラーは NetVault Backup バイナリ・ログに記録され、**[ジョブ・ステータス]** ページでは無視されます。バックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
- **[失敗 — セーブセットは保持されます]:** バックアップ・ジョブから **[バックアップ・ジョブ失敗]** というステータスが返されますが、正常にバックアップされたアイテムからなるバックアップ・セーブセットが作成されます。
- **[失敗 — セーブセットは保持されません]:** バックアップ・ジョブから **[バックアップ・ジョブ失敗]** というステータスが返され、バックアップされたオブジェクトのセーブセットは保持されません。選択された一部のオブジェクトが正しくバックアップされた場合も削除されます。

- 4 **[Mysqldump オプション]** テキスト・ボックス内に、**mysqldump** ユーティリティを使用するジョブをリストします。

このオプションはダッシュ (-) またはダブルダッシュ (--) で開始する必要があります。また、(;|<>) などの文字を含めることはできません。

これらのオプションはまず `mysqldump` コマンドに追加され、次にプラグインが内部的に生成したオプションが続きます。この順番により、ここに入力したオプションが内部的に生成されたオプションと矛盾している場合、プラグインが生成したオプションが優先されます。

ジョブが失敗するようなエラーを `mysqldump` オプションが検出した場合、このエラーはジョブ・ログ内でエラー・ログ・メッセージに埋め込まれます。

以前、このタスクを実行するために MySQL オプション・ファイルをセットアップした場合、このテキスト・ボックスに入力したオプションは、オプション・ファイルに指定したオプションへ追加されます。本プラグインに既存の MySQL オプション・ファイルを無視させたい場合は、このテキスト・ボックスの先頭に `--no-defaults` と入力します。

使用している `mysqldump` バージョンでサポートされているオプションについて詳しくは、利用可能な MySQL ドキュメンテーションを参照してください。

! **注意:** この機能に、`--routines (-R)` または `--triggers` オプションを使用しないでください。このオプションを使用すると、データベース・テーブルのバックアップは完了するが、リストアが失敗するなど、バックアップの正常完了を阻害する場合があります。データベースのバックアップに必要なストアド・プロシージャやトリガがある場合、本プラグインは `--routines` や `--triggers` オプションとともに `mysqldump` コマンドを内部生成します。

5 **[保存]** をクリックして、セットを保存します。

6 **[新規セットの作成]** ダイアログ・ボックスで、セットの名前を指定して、**[保存]** をクリックします。

名前には英数字と英数字以外の文字を使用できますが、非ラテン語系の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

MySQL Enterprise バックアップ用バックアップ・オプションの設定

バックアップする目的のアイテムを選択した状態で、実行するバックアップ・タイプを選択したり、失敗した場合に異なる動作を選択したりすることもできます。

i **ヒント:** 既存のセットを使用するには、**[プラグイン・オプション]** リストで使用するセットを選択します。

1 **[プラグイン・オプション]** リストの隣にある、**[新規作成]** をクリックします。

2 利用可能なオプションを選択します。

- **[フル・バックアップ]** (デフォルト設定) : 現在の MySQL インスタンス内で設定されたすべてのデータベースおよびテーブルをバックアップするには、このオプションを選択します。
- **[増分バックアップ]** : 最後のフルまたは増分バックアップ以降に変更が加わったデータ (InnoDB テーブル用) またはテーブル全体 (非 InnoDB テーブル用) のみをバックアップする場合、このオプションを選択します。
- **[転送可能なテーブルスペース (TTS) バックアップ]** : MySQL TTS 機能を利用する部分バックアップを作成する場合は、このオプションを選択します。

3 **[保存]** をクリックして、セットを保存します。

4 **[新規セットの作成]** ダイアログ・ボックスで、セットの名前を指定して、**[保存]** をクリックします。

名前には英数字と英数字以外の文字を使用できますが、非ラテン語系の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

ジョブのファイナライズと実行

- 1 [スケジュール]、[ターゲット・ストレージ]、および [詳細設定] リストを使用して、その他の必要なオプションを設定します。
- 2 [保存] または [保存 & 実行] の、どちらか適切な方をクリックします。

i | **ヒント:** すでに作成および保存しているジョブを実行するには、[ナビゲーション]パネルで [ジョブ定義管理] を選択し、目的のジョブを選択して、[今すぐ実行] をクリックします。

[ジョブ・ステータス] ページで進捗状況を監視したり、[ログ参照] ページでログを表示したりできます。詳しくは、『Quest NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

データのリストア

- データのリストア：概要
- MySQL におけるデータのリストア
- 高度な MySQL Standard/Community 用リストア手順

データのリストア：概要

このトピックでは、プラグインのリストア・プロセスと利用可能なすべての機能について説明します。さらに、「MySQL Standard/Community 用リストア・シナリオ例」および「MySQL Enterprise バックアップ用リストア・シナリオ例」では、さまざまなリストア・タイプの例が用意されています。Quest では、これらのトピックを熟読し、さまざまなリストア・タイプに対して利用可能な機能と適用方法について確認するようお勧めします。

- MySQL Standard/Community に利用可能なリストア方法の確認
- MySQL Enterprise バックアップに利用可能なリストア・オプションの確認

MySQL Standard/Community に利用可能なリストア方法の確認

リストアを正常に実行するには、利用可能なリストア・タイプについてよく理解する必要があります。

フル・リストアまたは個々のデータベース/テーブル・コピーのみのリストア

プラグインでフル・バックアップまたは個々のデータベース/テーブル・コピーのみのバックアップを実行すると、MySQL の `mysqldump` ユーティリティを使用して、テーブルの作成とデータ追加に使用される SQL ステートメントがバックアップ・メディアに直接送信されます。プラグインがこれらのいずれかの形式で作成されたバックアップをリストアするときに、SQL ステートメントがバックアップ・メディアから直接読み取られ、自動的に実行されます。

増分または差分リストア

プラグインは増分または差分バックアップを実行するときに、MySQL バイナリ・ログ・インデックスを使用して、バックアップ・メディアにコピーする必要があるバイナリ・ログを判断します。これらのバックアップをリストアすると、バイナリ・ログがテンポラリ・ディレクトリ (`NETVAULT_HOME/tmp/MySQL`) にリストアされます。その後、`mysqlbinlog` によって、バイナリ・ログに記録された各トランザクション用の SQL ステートメントが生成され、自動的に実行されます。このプロセスを「バイナリ・ログの適用」と言います。

増分および差分リストアの実行中に、バイナリ・ログに記録されたすべてのトランザクションを適用することも、特定時点までのトランザクションを適用 (PIT リカバリ) することもできます。PIT リカバリは、開発者が

誤ってテーブルを削除したり誤った更新を実行するなど、データ損傷の直前の時点にリカバリする場合に便利です。

時間に基づくPoint-in-Time (PIT)リカバリ

PIT リカバリは、増分または差分リストアの実行中にリストア対象のバイナリ・ログを使用して実行できます。時間に基づく PIT リカバリは、データの損傷時刻が分かっている場合に便利です。たとえば、開発者が午前 06:00:00 にテーブルを削除した場合、中止時刻を午前 05:55:00 に設定して PIT リカバリを実行できます。

通常、時間に基づく PIT リカバリは、1つの手順から成るプロセスです。[バイナリ・ログをリストアし適用する] ([オプション] タブの [バイナリ・ログをリストアし適用する] を選択) の対象として選択したバイナリ・ログを増分または差分バックアップからリストアし、不要なトランザクションの直前の中止時刻を指定します。

位置に基づくPoint-in-Time (PIT)リカバリ

データが損傷した実際の時刻が分からない場合、またはより正確なリカバリを必要とする場合、位置に基づくポイント・イン・タイム・リカバリを使用します。たとえば、開発者がデータベースからテーブルを削除したが、テーブルを削除した正確な時刻が分からない場合は、位置に基づく PIT リカバリを使用します。

位置に基づく PIT リカバリは、3つの手順から成るプロセスです。

- 1 [オプション] タブの [テンポラリ・ディレクトリへのログをリストアし、時間あるいは位置を特定する] オプションを選択して、増分または差分バックアップから MySQL サーバーのテンポラリ・ディレクトリにバイナリ・ログをリストアします。
- 2 MySQL の `mysqlbinlog` ユーティリティを使用して、不要なトランザクションの位置を特定します。詳しくは、『MySQL リファレンス・マニュアル』の「任意時点のリカバリ」セクションを参照してください。
- 3 同じ増分または差分バックアップを再度リストアします。ただし、今度は [テンポラリ・ディレクトリからのバイナリ・ログを適用する] リストア・オプションを選択し、不要なトランザクションの直前の停止位置を指定します。

MySQL Enterprise バックアップに利用可能なリストア・オプションの確認

MEB ベース・バックアップ方法では、フル・リストアまたは増分リストアを実行できます。TTS バックアップ・オプションを使用した場合は、TTS リストアを完了するオプションもあります。TTS リストア・プロセスでは、部分リストアと呼ばれる特定のテーブルをリストアする追加オプションがあり、指定したテーブルのいずれかの名前も変更できます。

i | 重要: TTSバックアップまたはリストアを使用する機能には制限があるため、Questでは、TTSオプションを慎重に使用することをお勧めします。

TTS バックアップをリストアする場合は、次の制約事項に注意する必要があります。

- サーバーとの接続を確立する必要があるため、宛先の MySQL サーバーが動作していることを確認します。
- リストアするテーブルが宛先サーバーに存在しないことを確認します。
- バックアップが実行された元の MySQL サーバーで使用されていたのと同じページ・サイズが宛先サーバーで使用されていることを確認します。
- 宛先サーバーで `innodb_file_per_table` オプションが有効になっていることを確認します。
- リストアしようとしている InnoDB ファイル (.ibd ファイル) が、宛先サーバーの `innodb_file_format` 変数の値と一致しない場合、リストアは失敗します。

詳細は、「<https://dev.mysql.com/doc/mysql-enterprise-backup/4.0/en/restore-use-tts.html>」を参照してください。

MySQL におけるデータのリストア

Plug-in for MySQL を使って標準リストアを実行するには、以下のトピックで説明する手順に従います。

- リストア対象データの選択
- リストア・オプションの設定
- ジョブのファイナライズと実行
- MySQL Standard/Community 用リストア・シナリオ例
- MySQL Enterprise バックアップ用リストア・シナリオ例

リストア対象データの選択

- 1 [ナビゲーション] パネルで、[リストア・ジョブ作成] をクリックします。
- 2 [リストア・ジョブ作成 - セーブセットの選択] ページで、[プラグイン・タイプ] リストから [Plug-in for MySQL] を選択します。
- 3 セーブセットのテーブルに表示されている項目をさらにフィルタリングするには、[クライアント]、[日付]、[ジョブ ID] リストを使用します。

表にはセーブセット名（ジョブ・タイトルとセーブセット ID）、作成日時、およびサイズが表示されます。デフォルトで、リストは [作成日] 列でソートされます。

- 4 セーブセットの表で、適切な項目を選択します。

セーブセットを選択すると、以下の情報が [セーブセット情報] に表示されます。ジョブ ID、ジョブ・タイトル、サーバー名、クライアント名、プラグイン名、セーブセットの日時、リタイア設定、増分バックアップかどうか、アーカイブかどうか、セーブセットのサイズ、スナップショットベースのバックアップかどうかなど。
- 5 [次へ] をクリックします。
- 6 [セレクション セット作成] ページで、リストアするデータを選択します。

リストアの対象として選択可能な最初のノードは、リカバーするバックアップのタイプに基づいて異なります：

- **フル・バックアップまたは個々のデータベース/テーブル・コピーのみのバックアップ**：ルート・ノードが [全てのデータベース] として表示されます。これは、実際のデータベース/テーブル・データがバックアップに含まれたからです。

i **重要：** ルート・ノードは [全てのデータベース] という名前ですが、ターゲットの MySQL インスタンスに対して現存するデータベースをすべて包括していません。このノードを選択すると、バックアップ・ジョブとして実際に選択されたデータ・アイテムのみがリストアされます（つまり、リストア対象としてこのノードを選択すると、MySQL インスタンス内に現存するすべてのデータベースはリストアされず、単にバックアップに含まれたデータベースのみがリストアされます）。

- **増分または差分バックアップ**：ルート・ノードは [バイナリ・ログ] として表示されます。これは、この形式のバックアップに、前のバックアップの実行以降に行われたトランザクション（バイナリ・ログ）が含まれているからです。

- 7 アイテム別のリストアを行うには、ルート・ノードをダブルクリックして開き、バックアップに含まれた個々のデータベースを表示します。

また、個々のデータベースを開いて、テーブルを表示し選択することもできます。

i **重要：** MySQL は、データベース情報の格納にさまざまなファイル形式を使用します。リストアしたデータベースが確実に機能するよう、必ずリストア・プロセスに .frm ファイルを含めるよう確認してください。

リストア・オプションの設定

[MySQL Standard/Community] または [MySQL Enterprise バックアップ] オプションのいずれを使用するかによって、[オプション] タブに各オプションが表示されます。

- [MySQL Standard/Community 用リストア・オプションの設定](#)
- [MySQL Enterprise バックアップ用リストア・オプションの設定](#)

MySQL Standard/Community用リストア・オプションの設定

[セクション・セット作成] ページで、[プラグイン・オプションの編集] をクリックして、[特定時点リカバリを可能にする] タブおよび [リストア先] タブで以下のパラメータを設定します。リストアに選択されたバックアップ・タイプによって、さまざまなオプションが表示されます。

- [フルまたは個々のデータベース・リストア・オプション](#)
- [増分または差分データベース・リストア・オプション](#)

フルまたは個々のデータベース・リストア・オプション

フル・バックアップまたは個別データベース / テーブルのコピーのみバックアップのいずれかをリストアするには、以下の手順に従います。

- 1 以下のガイドラインを利用して、[リカバリ時] タブで利用可能なオプションを選択します。
 - **[現在のバイナリ・ログで PIT リカバリを実行する]** : MySQL サーバー上の MySQL バイナリ・ログ・ディレクトリに保持されているバイナリ・ログを使用して、選択したデータ・オブジェクトについて**特定時点形式**のリストアを実行する場合に選択します。このオプションを選択すると、このタブ上のすべてのオプションが使用可能になります。
 - **[Point In Time (特定時点) タイプ]** : このセクションで特定時点リカバリに利用可能な形式を選択します。
 - **[時間に基づく PIT]** (デフォルト選択) : このオプションを選択して、選択したデータを**指定した時間**へリストアします (**[時間に基づく Point-in-Time (PIT) リカバリ]** で詳説)。このオプションを選択すると、**[時間に基づく PIT の詳細]** セクションが有効になります。
 - **[位置に基づく PIT]** : このオプションを選択して、**選択したデータを不要なトランザクションの直前に存在した特定の停止位置**へリストアします (**[位置に基づく Point-in-Time (PIT) リカバリ]** で詳説)。このオプションを選択すると、**[位置に基づく PIT の詳細]** セクションが有効になります。
 - **[時間に基づく PIT の詳細]** : **[時間に基づく PIT]** を選択した場合、以下の利用可能なオプションを選択します。
 - **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]** : 不要なトランザクションの**前**に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した時刻より**後**に実行されたトランザクションはすべて失われます。付随する **[中止日 / 時間]** フィールドに、目的の日付と時刻 (24 時間形式) を指定します。
 - **[誤った / 不良の SQL ステートメントの後に、リカバリを可能にする]** : 不要なトランザクションの**後**に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した時刻より**前**に実行されたトランザクションはすべて失われます。付随する **[開始日 / 時間]** フィールドに、目的の日付と時刻 (24 時間形式) を指定します。特定の開始日と時刻に加え、トランザクションの中止日と時刻を設定することもできます。
 - **[なし]** (デフォルト) : 指定した日付と時刻の後に実行されたすべてのトランザクションをリストアする場合は、このラジオ・ボタンを選択したままにします。

- **[具体的な日付]**: 指定した時間範囲に実行されたトランザクションのみを含めるには、このラジオ・ボタンを選択します。付随する時刻および日付フィールドに目的の中止時刻を入力します(24時間形式)。

i 重要: リストアされたバイナリ・ログと現在のバイナリ・ログの両方に対してPITリカバリを有効にした場合、中止時刻が、リストアされたバイナリ・ログまたは現在のバイナリ・ログのどちらにあるかを特定する必要はありません。MySQLは、指定された時刻で自動的に中止/開始し、指定された最終的な中止時点より後のバイナリ・ログをすべて無視します。

これらのオプションを両方有効にすることもできます。特定の時間範囲に不要なトランザクションが実行された場合は、これらのオプションを両方使用します。たとえば、2011年1月29日の午前11:00から午前11:15までの間に収集されたデータが不要の場合、**[誤った/不良のSQLステートメントの前に、リカバリを可能にする]**オプションを有効にし、**[中止日/時間]**として**11:00 - 2011年1月29日**を入力します。また、**[誤った/不良のSQLステートメントの後に、リカバリを可能にする]**オプションを選択し、**[開始日/時間]**として**11:15 - 2007年1月29日**を入力します。この結果、2011年1月29日の11:00から11:15までの間に実行されたすべてのトランザクションがリストアから除外されます。

- **[位置に基づく PIT の詳細]**: **[位置に基づく PIT]** を選択した場合、以下の利用可能なオプションを選択します。
 - **[誤った/不良のSQLステートメントの前に、リカバリを可能にする]**: 不要なトランザクションの**前**に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した位置より**後**に実行されたトランザクションはすべて失われます。このオプションには、以下の関連オプションがあります。
 - **[停止位置]**: このフィールドに、バイナリ・ログ内の不要なトランザクションよりも**前**の位置を入力します。たとえば、不要なトランザクションの位置が805の場合、804を入力します。
 - **[終了位置を含むバイナリ・ログ]**: このドロップダウンを使用して、**[停止位置]**フィールドに指定した停止位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合(またはここに目的のファイルが表示されない場合)、**[その他]**オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。
 - **[誤った/不良のSQLステートメントの後に、リカバリを可能にする]**: 不要なトランザクションの**後**に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した位置より**前**に実行されたトランザクションはすべて失われます。このオプションにも、以下の関連オプションがあります。
 - **[開始位置]**: このフィールドに、バイナリ・ログ内の不要なトランザクションよりも**後**の位置を入力します。たとえば、不要なトランザクションの位置が805の場合、806を入力します。
 - **[開始位置を含むバイナリ・ログ]**: このドロップダウンを使用して、**[開始位置]**フィールドに指定した開始位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合(またはここに目的のファイルが表示されない場合)、**[その他]**オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。
 - **[停止位置]: [なし]**(デフォルト選択) - **[誤った/不良のSQLステートメントの後に、リカバリを可能にする]**に指定した**開始位置**の後に実行された**すべての**トランザクションをリカバリするには、このラジオ・ボタンを選択したままにします。
 - **[停止位置]: [具体的な位置]**: バイナリ・ログの特定の位置範囲に実行されたトランザクションのみを含めるには、このオプションを選択します。**[具体的な位置]**オプションに付随するフィールドに目的の停止位置を入力し、**[終了位置を含むバイナリ・ログ]**ドロップダウン・リストで適切なバイナリ・ログ・ファイルを選択します(別のファイルを使用する場合は、このドロップダウンから**[その他]**を選

択し、付随するテキスト・ボックスにファイル名を指定します)。**[開始位置]**に指定した位置から**[具体的な位置]**フィールドに指定した位置までの間に実行されたトランザクションのみがリストアの対象となります。

- i | 重要:** これらのオプションを両方有効にすることもできます。特定の位置範囲に不要なトランザクションが実行された場合は、これらのオプションを両方使用します。たとえば、位置805から位置810までの間に収集されたデータに不要なトランザクションが含まれている場合、**[誤った/不良のSQLステートメントの前に、リカバリを可能にする]**オプションを有効にし、**[停止位置]**として805を入力してから、付随するオプションで、バイナリ・ログ・ファイルを呼び出すよう設定します。また、**[誤った/不良のSQLステートメントの後に、リカバリを可能にする]**オプションを有効にし、**[開始位置]**として810を入力してから、付随するオプションで、バイナリ・ログ・ファイルを呼び出すよう設定します。この結果、指定したバイナリ・ログ・ファイルの805から810までの間に記録されたすべてのトランザクションがリストアから除外されます。停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定する必要があります。

2 以下のガイドラインに従い、**[リストア先]** タブで利用可能なオプションを選択します。

- **同一 MySQL インスタンスへのリストアを実行する場合:** リストアのターゲットがバックアップされた元のインスタンスと同じ場合には、これらのフィールドはブランクのままにします。NetVault Backup は **[設定]** ダイアログ・ボックス内の値セットを使用します。詳細は、「**プラグインの設定**」を参照してください。
- **別の MySQL インスタンスへのリストアを実行する場合:** 選択したデータのリストアを別のインスタンスへ移動する場合には、新しいインスタンスへのリストア・アクセスを許可するために **[ユーザー名]** と **[パスワード]** の各フィールドにログイン情報を入力する必要があります。さらに、新しいインスタンスに設定された NetVault Backup 名を **[インスタンス名]** フィールドに入力します (**[設定]** ダイアログ・ボックスで **[MySQL インスタンス名]** に設定した名前を入力します。詳しくは、「**プラグインの設定**」を参照してください)。

- i | 重要:** リストアを別のMySQLインスタンスへ移動する前に、「**異なるMySQLサーバーへのリストア**」を参照して詳細な手順を確認する必要があります。

増分または差分データベース・リストア・オプション

増分または差分バックアップのいずれかをリストアするには、以下の手順に従います。

1 以下のガイドラインを利用して、**[リカバリ時]** タブで利用可能なオプションを選択します。

- **[PIT リカバリを実行する]:** 選択したデータ・アイテムについて**特定時点形式**のリストアを実行するには、このオプションを選択します。このオプションを選択すると、このタブ上のすべてのオプションが使用可能になります。

増分および差分リストアでは、リストアを完了するためにバイナリ・ログが使用されます。このため、この形式のバックアップをリストアする場合は、選択したデータベースに関連するバイナリ・ログをどのようにリカバリするかを決定します。以下の方法の1つを選択します。

- **[バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)]:** 1つのリストア・ジョブで、バックアップ・デバイスからのバイナリ・ログをリストアし、かつバイナリ・ログに記録されたトランザクションを適用する場合に選択します。現在 MySQL バイナリ・ログ・ディレクトリに保存されているバイナリ・ログを使用して**特定時点リカバリ**を実行する場合、**[現在のバイナリ・ログを含む]** チェック・ボックスを選択します。このプロセスは、増分/差分バックアップに保存されたバイナリ・ログのトランザクションをリストアして適用した**後**に実行されます。
- **[テンポラリ・ディレクトリへのログをリストアし、時間あるいは位置を特定する]:** このバイナリ・ログのリストア方法を選択した場合、リストア・ジョブによって、選択した増分/差分バックアップに関連するバイナリ・ログが MySQL サーバー上の**テンポラリ・ディレクトリ** (**[NETVAULT_HOME/tmp/MySQL/]**) にリストアされます。このオプションにより、**mysqlbinlog** ユーティリティを使用して、リカバリされたログを調べ、データが損傷した時刻/位置を特定できるようになります。

- **【停止位置】**:このフィールドに、バイナリ・ログ内の不要なトランザクションより**前**の位置を入力します。たとえば、不要なトランザクションの位置が805の場合、804を入力します。
- **【終了位置を含むバイナリ・ログ】**:このドロップダウンを使用して、**【停止位置】**フィールドに指定した停止位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合(またはここに目的のファイルが表示されない場合)、**【その他】**オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。
- **【誤った/不良のSQLステートメントの後に、リカバリを可能にする】**:不要なトランザクションの**後**に起こったすべてのトランザクションをリストアするには、このオプションを選択します。このオプションのみを選択した場合、ここに指定した位置より**前**に実行されたトランザクションはすべて失われます。このオプションにも、以下の関連オプションがあります。
 - **【開始位置】**:このフィールドに、バイナリ・ログ内の不要なトランザクションより**後**の位置を入力します。たとえば、不要なトランザクションの位置が805の場合、806を入力します。
 - **【開始位置を含むバイナリ・ログ】**:このドロップダウンを使用して、**【開始位置】**フィールドに指定した開始位置が含まれるバイナリ・ログ・ファイルを選択します。別のファイルが必要な場合(またはここに目的のファイルが表示されない場合)、**【その他】**オプションを選択し、付随するテキスト・ボックスに目的のファイル名を入力します。
 - **【停止位置】**:**【なし】**(デフォルト):**【誤った/不良のSQLステートメントの後に、リカバリを可能にする】**に指定した**開始位置**の後に実行された**すべての**トランザクションをリカバリするには、このラジオ・ボタンを選択したままにします。
 - **【停止位置】**:**【具体的な位置】**:バイナリ・ログの特定の位置範囲に実行されたトランザクションのみを含めるには、このオプションを選択します。**【具体的な位置】**オプションに付随するフィールドに目的の停止位置を入力し、**【終了位置を含むバイナリ・ログ】**ドロップダウン・リストで適切なバイナリ・ログ・ファイルを選択します(別のファイルを使用する場合は、このドロップダウンから**【その他】**を選択し、付随するテキスト・ボックスにファイル名を指定します)。**【開始位置】**に指定した位置から**【具体的な位置】**フィールドに指定した位置までの間に実行されたトランザクションのみがリストアの対象となります。

i 重要: これらのオプションを両方有効にすることもできます。特定の位置範囲に不要なトランザクションが実行された場合は、これらのオプションを両方使用します。たとえば、位置805から位置810までの間に収集されたデータに不要なトランザクションが含まれている場合、**【誤った/不良のSQLステートメントの前に、リカバリを可能にする】**オプションを有効にし、**【停止位置】**として**805**を入力してから、付随するオプションで、バイナリ・ログ・ファイルを呼び出すよう設定します。また、**【誤った/不良のSQLステートメントの後に、リカバリを可能にする】**オプションを有効にし、**【開始位置】**として**810**を入力してから、付随するオプションで、バイナリ・ログ・ファイルを呼び出すよう設定します。この結果、指定したバイナリ・ログ・ファイルの805から810までの間に記録されたすべてのトランザクションがリストアから除外されます。停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定する必要があります。

2 以下のガイドラインに従い、**【リストア先】** サブタブで利用可能なオプションを選択します。

このタブには、**【リストア先詳細】** セクションが表示されます。このフィールドにアカウント情報を入力して、MySQL のターゲット・インスタンスへのリストア・アクセスを許可します。必要なリストア・タイプに応じて、以下のようにオプションを使用します。

- **同一 MySQL インスタンスへのリストアを実行する場合**: リストアのターゲットがバックアップされた元のインスタンスと同じ場合には、これらのフィールドは空白のままにします。NetVault Backup は**【設定】** ダイアログ・ボックス内の値セットを使用します。詳細は、「**プラグインの設定**」を参照してください。

- 別の MySQL インスタンスへのリストアを実行する場合：選択したデータのリストアを別のインスタンスへ移動する場合には、新しいインスタンスへのリストア・アクセスを許可するために【ユーザー名】と【パスワード】の各フィールドにログイン情報を入力する必要があります。さらに、新しいインスタンスに設定された NetVault Backup 名を【インスタンス名】フィールドに入力します（【設定】ダイアログ・ボックスで【MySQL インスタンス名】に設定した名前を入力します。詳しくは、「プラグインの設定」を参照してください）。

i | **重要:** リストアを別のMySQLインスタンスへ移動する前に、「異なるMySQLサーバーへのリストア」を参照して詳細な手順を確認する必要があります。

MySQL Enterpriseバックアップ用リストア・オプションの設定

【セクション・セット作成】 ページで、【プラグイン・オプションの編集】 をクリックして、【オプション】 タブで適切なパラメータを設定します。

i | **重要:** リストアを実行する前に、【MySQL Enterpriseバックアップ】オプションを使用して作成されたフル・バックアップ内に含まれるすべてのデータを（少なくとも一時的に）格納するのに十分な空き容量がデフォルトのNetVault BackupのTempディレクトリにあることを確認します。【一般】オプションを使用して、十分な空き容量を持つ格納場所にデフォルト設定を変更することができ、これによりマップ済みドライブやネットワーク・ファイル・システム(NFS)、またはSMBマウント等も使用できるようになります。【ナビゲーション】パネルで【設定変更】をクリックし、【クライアント設定】をクリックして、次に【システムおよびセキュリティ】セクションの【一般】をクリックします。

- 【フル・リストア】：利用可能なオプションを選択します。
 - 【Raw フル・バックアップをリストア、抽出し、ログを適用してTemp ディレクトリ内に準備済みフル・バックアップを生成】（デフォルト選択）：このオプションを選択して、MySQL サーバーのデータ・リポジトリ・ディレクトリ階層に対応する一時位置へフル・バックアップをリストアします。このオプションを使用する場合は、どのバックアップをリストアするか識別していることが前提です。まだの場合は、以下の2つのオプションを使用することができます。
 - 【フル・バックアップ・イメージをTemp ファイルにリストア】：バックアップのコンテンツをリストして、次のオプションを実行する必要があるか識別する場合はこのオプションを選択します。
 - 【Temp ファイルからRaw フル・バックアップを抽出し、ログを適用してTemp ディレクトリ内に準備済みフル・バックアップを生成】：前述のオプションを使用した結果について、どのバックアップをリストアする必要があるか識別するには、このオプションを選択します。このオプションにより、フル・バックアップがMySQLサーバーのデータ・リポジトリ・ディレクトリ階層に対応する一時格納場所へリストアされます。
 - 【MySQLサーバーをシャットダウンし、準備済みフル・バックアップをMySQLサーバー・リポジトリへコピー・バック】（標準のフル・リストアに使用可能なオプション）：MySQLサーバーをシャットダウンし、リストア済みコンテンツを一時格納場所から元の位置へコピー・バックする準備が整った際、このオプションを選択します。
 - 【準備済みフル・バックアップをMySQLサーバー・リポジトリへコピー・バック】（TTSフル・リストアに使用可能なオプション）：リストア済みコンテンツを一時格納場所から元の位置へコピー・バックする際に、このオプションを選択します。以下の2つのオプションも使用可能です。
 - 【テーブルを含める】：部分リストアを実行する場合は、このフィールドに正規表現を入力して、リストアに含めるテーブルの命名パターンを記述します。このフィールドに入力すると、プラグインによって `--include-tables` MySQL コマンドが発行されます。
 - 【テーブル名の変更】：【テーブルを含める】フィールドに入力してリストアするテーブルを指定した場合は、このフィールドを使用して、指定したテーブルのいずれかの名前を変更できます。テーブル名を変更するには、`original_name to new_name` 表現を使用します。このフィールドに入力すると、プラグインによって `--rename` MySQL コマンドが発行されます。

i **重要:** リストアにTTSバックアップを選択して[MySQLサーバーをシャットダウンし、準備済みフル・バックアップをMySQLサーバー・リポジトリへコピー・バック]オプションを使用する場合、プラグインは[リストア・セレクション]ダイアログ・ボックスで選択した項目を無視します。プラグインは、[テーブルを含める]フィールドに指定されたテーブルと対応するデータベースのみをリストアします。

- [バックアップ・イメージの検証]: 抽出したデータに対して検証コマンドの実行を本プラグインに指示する場合、このチェック・ボックスを選択します。
- [バックアップ・イメージのリスト]: 出力ログ内のバックアップ・コンテンツをリストするには、このオプションを選択します。
- [増分リストア]: 利用可能なオプションを選択します。
 - [増分バックアップをリストア、抽出し、Temp ディレクトリ内の準備済みフル・バックアップに適用] (デフォルト選択): 増分バックアップをリストアする場合、このオプションを選択します。このオプションを使用する場合は、どのバックアップをリストアするか識別していることが前提です。まだの場合は、以下の2つのオプションを使用することができます。
 - [増分バックアップ・イメージを Temp ファイルにリストア]: バックアップのコンテンツをリストして、次のオプションを実行する必要があるか識別する場合はこのオプションを選択します。
 - [Temp ファイルから増分バックアップを抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成]: 前述のオプションを使用した結果について、どのバックアップをリストアする必要があるか識別するには、このオプションを選択します。
 - [MySQL サーバーをシャットダウンし、準備済みフル・バックアップを MySQL サーバー・リポジトリへコピー・バック]: MySQL サーバーをシャットダウンし、リストア済みコンテンツを一時格納場所から元の位置へコピー・バックする準備が整った際、このオプションを選択します。
 - [バックアップ・イメージの検証]: 抽出したデータに対して検証コマンドの実行を本プラグインに指示する場合、このチェック・ボックスを選択します。
 - [バックアップ・イメージのリスト]: 出力ログ内のバックアップ・コンテンツをリストするには、このオプションを選択します。

ジョブのファイナライズと実行

最終ステップには、[スケジュール]、[ソース・オプション]、および [詳細設定] ページの追加オプション設定、ジョブの実行、および [ジョブ・ステータス] と [ログ参照] ページからの進捗状況の監視が含まれています。これらのページとオプションは、すべての NetVault Backup プラグインに共通しています。詳しくは、『Quest NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

- 1 設定を保存するには、[OK]、続いて [次へ] をクリックします。
- 2 デフォルト設定を使用しない場合は、[ジョブ名] に、ジョブの名前を指定します。

進捗状況を監視する際にジョブを識別しやすくするため、分かりやすい名前を割り当てます。ジョブ名には英数字と英数字以外の文字を使用できますが、非ラテン語系の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

i **重要:** ターゲットOSのファイル名としてサポートされていない特殊文字を使用しないよう注意してください。たとえば、Windowsでは、\、*、@などの文字は使用できません。これは、Plug-in for MySQLがデータを一時的にリストアするために、[ジョブ・タイトル]と同じ名前のフォルダを作成しようとするからです。

- 3 [クライアント指定] リストで、データをリストアするマシンを選択します。

i **ヒント:** [選択]をクリックして、[クライアント指定選択]ダイアログ・ボックスから適切なクライアントを検索、選択することもできます。

- 4 [スケジュール]、[ソース・オプション]、および [詳細設定] リストを使って、その他の必要なオプションを設定します。
- 5 [保存] または [保存 & 実行] の、どちらか適切な方をクリックします。

[ジョブ・ステータス] ページで進捗状況を監視したり、[ログ参照] ページでログを表示したりできます。詳しくは、『Quest NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

i 重要: LinuxまたはUNIX環境でMySQL Enterpriseバックアップを使用している場合は、リストアされたデータのファイル所有権および権限情報が、データのバックアップ前と一致していることを確認します。**mysqlbackup**ユーティリティは、バックアップ・プロセス中にこの情報を記録しないため、リストアの完了後に情報が異なっている場合があります。詳細は、「https://docs.oracle.com/cd/E17952_01/mysql-enterprise-backup-3.11-en/bugs.backup.html」を参照してください。

MySQL Standard/Community 用リストア・シナリオ例

障害またはデータ損傷から正しくリカバリするには、ジョブの設定時に、リストア対象として選択するデータおよび [オプション] タブのオプションに関してさまざまな設定を行う必要があります。以降のトピックでは、さまざまなタイプのリカバリ例を示し、必要となるオプションについて説明します。

- フル・バックアップのみによるリストア・シナリオ
- フルおよび増分バックアップによるリストア・シナリオ
- フルおよび差分バックアップによるリストア・シナリオ
- MIXED バイナリ・ログ形式が使用され、クロスデータベースの更新が発行された場合の PIT リストア

フル・バックアップのみによるリストア・シナリオ

以下の例で、MySQL 管理者は毎晩午後 11 : 00 にフル・バックアップを実行するバックアップ・ポリシーを設定しました。

フル・バックアップのリストアと時間に基づく特定時点リカバリ

データベース管理者は月曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、データベース管理者の出勤前の **月曜日午前 6:00** に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

方法1: 誤ったステートメントの *前* をリカバリする

データベース管理者は、drop table コマンドが実行される **直前** の特定時点までをリカバリすることを決定しました。つまり、データベース管理者は日曜日の夜に実行されたフル・バックアップをリストアし、現在のバイナリ・ログを使用して PIT リカバリを実行する必要があります。

- 1 **日曜日の夜からのフル・リストアを選択する:** データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の [オプション] タブのオプション設定:** データベース管理者は以下のオプションを選択します。
 - **[現在のバイナリ・ログで PIT リカバリを実行する]:** このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - **[時間に基づく PIT]:** リストア・タイプとして選択します。

- **【誤った / 不良の SQL ステートメントの前に、リカバリを可能にする】**：このオプションを選択し、**【中止日 / 時間】** を「5:59」、「2007 年 1 月 8 日」（月曜日の日付の午前 6:00 の 1 分前）に設定します。

3 ジョブを開始します。

方法2: 誤った / 不良の SQL ステートメントの 前および 後をリカバリする

データベース管理者は、drop table コマンドが実行される **直前**の特定時点までをリカバリすることを決定しました。また、誤った / 不良の SQL ステートメントの **後**の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。

- 1 **日曜日の夜からのフル・リストアを選択する**：データベース管理者は **【リストア・ジョブ作成 – セーブセットの選択】** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の【オプション】タブのオプション設定**：データベース管理者は以下のオプションを選択します。
 - **【現在のバイナリ・ログで PIT リカバリを実行する】**：このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - **【時間に基づく PIT】**：リストア・タイプとして選択します。
 - **【誤った / 不良の SQL ステートメントの前に、リカバリを可能にする】**：このオプションを選択し、**【中止日 / 時間】** を「5:59」、「2007 年 1 月 8 日」（月曜日の日付の午前 6:00 の 1 分前）に設定します。
 - **【誤った / 不良の SQL ステートメントの後に、リカバリを可能にする】**：Order テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より **後**の時刻と日付を **【開始日 / 時間】** オプションに入力します。最後に、指定したバイナリ・ログ・ファイルを最後までリカバリするために、**【開始日 / 時間】** オプションで **【なし】** ラジオ・ボタンを選択しました。

フル・リストアと位置に基づく特定時点リカバリ

データベース管理者は月曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、データベース管理者の出勤前の **月曜日午前 6:00** に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

方法1: 誤ったステートメントの 前をリカバリする

データベース管理者は、drop table コマンドが実行される **直前**の特定時点までをリカバリすることを決定しました。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。このプロセスを行うには、データベース管理者は日曜日のフル・バックアップをリストアし、現在のバイナリ・ログを使用して PIT リカバリを実行する必要があります。

- 1 **【リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する】**：リストアしない drop table SQL ステートメントの **位置**を特定するために、この手順を NetVault Backup の外で実行します（この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください）。この処理で、データベース管理者は drop table ステートメントが **MYSQLSVR-bin.000009** バイナリ・ログの **805** の位置にあることを特定しました。
- 2 **日曜日の夜からのフル・リストアを選択する**：データベース管理者は **【リストア・ジョブ作成 – セーブセットの選択】** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 3 **リストア関連の【オプション】タブのオプション設定**：データベース管理者は以下のオプションを選択します。

- **[現在のバイナリ・ログで PIT リカバリを実行する]**：このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
- **[位置に基づく PIT]**：リストア・タイプとして選択します。
- **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]**：このオプションを選択し、**[停止位置]** を「804」（mysqlbinlog で特定した位置の**前**の位置）に設定します。**[終了位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、ターゲット・バイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。

i | **重要**：停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

4 ジョブを開始します。

方法2: 誤った / 不良の SQL ステートメントの**前**および**後**をリカバリする

データベース管理者は、drop table コマンドが実行される**直前**の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の**後**の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。このプロセスを行うには、データベース管理者は日曜日のフル・バックアップをリストアし、現在のバイナリ・ログを使用して PIT リカバリを実行する必要があります。

- 1 **[リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する]**：リストアしない drop table SQL ステートメントの**位置**を特定するために、この手順を NetVault Backup の外で実行します（この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください）。この処理で、データベース管理者は drop table ステートメントが **MYSQLSVR-PM-bin.000009** バイナリ・ログの **805** の位置にあることを特定しました。
- 2 **日曜日の夜からのフル・リストアを選択する**：データベース管理者は **[リストア・ジョブ作成 - セーブセットの選択]** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 3 **リストア関連の [オプション] タブのオプション設定**：データベース管理者は以下のオプションを選択します。
 - **[現在のバイナリ・ログで PIT リカバリを実行する]**：このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - **[位置に基づく PIT]**：リストア・タイプとして選択します。
 - **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]**：このオプションを選択し、**[停止位置]** を「804」（mysqlbinlog で特定した位置の**前**の位置）に設定します。**[終了位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、ターゲット・バイナリ・ファイルの名前（MYSQLSVR-PM-bin.000009 など）をテキスト・ボックスに入力しました。
 - **[誤った / 不良の SQL ステートメントの後に、リカバリを可能にする]**：このオプションを選択し、**[開始位置]** を「806」（mysqlbinlog で特定した位置の**後**の位置）に設定します。**[開始位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、ターゲット・バイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。最後に、指定したバイナリ・ログ・ファイルを最後までリカバリするために、**[停止位置]** オプションで **[なし]** ラジオ・ボタンを選択しました。

i | **重要**：停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

4 ジョブを開始します。

フルおよび増分バックアップによるリストア・シナリオ

DBA はフル・バックアップを毎週 **日曜日の午後 11 時**に、増分バックアップを **月～土曜日の午後 11 時**に実行するバックアップ・ポリシーを作成しました。DBA は増分バックアップを実行しているため、各増分バックアップの実行後バイナリ・ログは **削除**されます。このプロセスにより、全体的なバックアップ時間は短くなりますが、リストアにはより多くの時間と手順が必要になります。

フルおよび増分リストアのみ

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、データベース管理者の出勤前の木曜日午前に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、最後の増分バックアップ、つまり **水曜日の夜**に実行されたバックアップの時点までを完全にリカバリすることを決定しました。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順2: 月曜日からの増分リストア

- 1 **月曜日の夜からの増分バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順3: 火曜日からの増分リストア

- 1 **火曜日の夜からの増分バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順4: 水曜日からの増分リストア

- 1 **[水曜日の夜からの増分バックアップを選択する]** : データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始します**。

フル・リストアと時間に基づく特定時点リカバリ

以降の例では、フル/増分バックアップ・シナリオを取り上げます。データベース管理者はデータを特定時点にリカバリしようと考えています。

方法1: リストアされたバイナリ・ログのみを使用して誤った/不良のSQLステートメントの**前**をリカバリする

データベース管理者は木曜日の午前9:00に、ユーザーが**Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**水曜日の午後8:00**に開発者が無意識のうちに**Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が**水曜日の午後8:00**にテーブルを削除した**直前**の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順2: 月曜日からの増分リストア

- 1 **月曜日の夜からの増分バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順3: 火曜日からの増分リストア

- 1 **火曜日の夜からの増分バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順4: 水曜日からの時間に基づくPITリストア

- 1 **[水曜日の夜からの増分バックアップを選択する]** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の [オプション] タブのオプション設定** : データベース管理者は以下のオプションを選択します。
 - **[PIT リカバリを実行する]** : このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にします。
 - **[バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)]** : このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定します。
 - **[時間に基づく PIT]** : リストア・タイプとして選択します。
 - **[誤った/不良のSQLステートメントの前に、リカバリを可能にする]** : このオプションを選択し、[中止日/時間] を「19:59」、「2007年1月10日」(水曜日の日付の午後8:00の1分前)に設定します。
- 3 **ジョブを開始します**。

方法2: リストアされたバイナリ・ログのみを使用して誤った/不良のSQLステートメントの**前**および**後**をリカバリする

データベース管理者は木曜日の午前9:00に、ユーザーが**Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**水曜日の午後8:00**に開発者が無意識のうちに**Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、午後8:00にDrop Table コマンドが実行された**直前**の特定時点までリカバリすることを決定しました。また、**Orders** テーブルが削除された時点の**後**の特定時点からバックアップされたバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。このことから、以下の手順を実行します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順2: 月曜日からの増分リストア

- 1 **月曜日の夜からの増分バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順3: 火曜日からの増分リストア

- 1 **火曜日の夜からの増分バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順4: 水曜日からの時間に基づくPITリストア

- 1 **[水曜日の夜からの増分バックアップを選択する]** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の [オプション] タブのオプション設定** : データベース管理者は以下のオプションを選択します。
 - **[PIT リカバリを実行する]** : このオプションを選択してPIT リカバリを指定し、関連するすべてのオプションを有効にします。
 - **[バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)]** : このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定します。
 - **[時間に基づく PIT]** : リストア・タイプとして選択します。
 - **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]** : このオプションを選択し、[中止日 / 時間] を「19:59」、「2007年1月10日」(水曜日の日付の午後8:00の1分前)に設定します。

- **【誤った/不良の SQL ステートメントの後に、リカバリを可能にする】**：Order テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より後の時刻と日付を **【開始日/時間】** オプションに入力します。最後に、バックアップに含まれるバイナリ・ログ・ファイルを最後までリカバリするために、**【開始日/時間】** オプションで **【なし】** ラジオ・ボタンを選択しました。

3 ジョブを開始します。

方法3: リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った/不良のSQLステートメントの**前**をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前 6:00** に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が**木曜日の午前 6:00** にテーブルを削除した**直前**の特定時点までデータベースをリストアするようリカバリを実行する必要があります。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする**：どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順2: 月曜日からの増分リストア

- 1 **月曜日の夜からの増分バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする**：どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順3: 火曜日からの増分リストア

- 1 **火曜日の夜からの増分バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする**：どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順4: 水曜日からの時間に基づくPITリストア

- 1 **【水曜日の夜からの増分バックアップを選択する】**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の【オプション】タブのオプション設定**：データベース管理者は以下のオプションを選択します。
 - **【PIT リカバリを実行する】**：このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にします。
 - **【バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)】**：このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定します。
 - **【現在のバイナリ・ログを含む】**：水曜日にバックアップが完了してから Drop Table コマンドを発行するまでの間に発生したエントリを適用するために、このオプションを有効にします。

- **[時間に基づく PIT]** : リストア・タイプとして選択します。
- **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]** : このオプションを選択し、**[中止日 / 時間]** を「5:59」、「2007 年 1 月 11 日」(月曜日の日付の午前 6:00 の 1 分前) に設定します。

3 ジョブを開始します。

方法4: リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った / 不良の SQL ステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前 6:00** に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される**直前**の特定時点までをリカバリすることを決定しました。また、**Orders** テーブルが削除された時点の**後**の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。このことから、以下の手順を実行します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する** : データベース管理者は **[リストア・ジョブ作成 - セーブセットの選択]** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順2: 月曜日からの増分リストア

- 1 **月曜日の夜からの増分バックアップを選択する** : データベース管理者は **[リストア・ジョブ作成 - セーブセットの選択]** ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順3: 火曜日からの増分リストア

- 1 **火曜日の夜からの増分バックアップを選択する** : データベース管理者は **[リストア・ジョブ作成 - セーブセットの選択]** ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順4: 水曜日からの時間に基づく PIT リストア

- 1 **[水曜日の夜からの増分バックアップを選択する]** : データベース管理者は **[リストア・ジョブ作成 - セーブセットの選択]** ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の [オプション] タブのオプション設定** : データベース管理者は以下のオプションを選択します。
 - **[PIT リカバリを実行する]** : このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にします。
 - **[バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)]** : このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定します。

- **【現在のバイナリ・ログを含む】**：水曜日にバックアップが完了してから Drop Table コマンドを発行するまでの間に発生したエントリを適用するために、このオプションを有効にします。
- **【時間に基づく PIT】**：リストア・タイプとして選択します。
- **【誤った / 不良の SQL ステートメントの前に、リカバリを可能にする】**：このオプションを選択し、**【中止日 / 時間】** を「5:59」、「2007 年 1 月 11 日」（月曜日の日付の午前 6:00 の 1 分前）に設定します。
- **【誤った / 不良の SQL ステートメントの後に、リカバリを可能にする】**：**Order** テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より後の時刻と日付を **【開始日 / 時間】** オプションに入力します。最後に、**現在のバイナリ・ログ・ファイル**を最後までリカバリするために、**【中止日 / 時間】** オプションで **【なし】** オプションを選択しました。

3 ジョブを開始します。

フル・リストアと位置に基づく特定時点リカバリ

以降の例では、フル / 増分バックアップ・シナリオを取り上げます。データベース管理者は、より確実な方法で時刻を特定し、データを特定時点にリカバリしようと考えています。このリカバリは、MySQL バイナリ・ログ・ファイル内で特定した位置の値を使用して行います。

方法1: リストアされたバイナリ・ログのみを使用して誤った / 不良の SQL ステートメントの 前をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**水曜日の午後 8:00** に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される **直前**の特定時点までをリカバリすることを決定しました。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。このプロセスを行うには、データベース管理者は、日曜日のフル・バックアップと、月曜日および火曜日に実行された増分バックアップをリストアしてから、水曜日に実行された増分バックアップを使用して位置に基づく PIT リカバリを実行する必要があります。以下にこのプロセスを示します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする**：どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順2: 月曜日からの増分リストア

- 1 **月曜日の夜からの増分バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする**：どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順3: 火曜日からの増分リストア

- 1 **火曜日の夜からの増分バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする**：どのオプションも使用しません。

- 3 ジョブを開始して完了するまで待ちます。

手順4:バックアップされたバイナリ・ログをリストアして誤ったSQLステートメントの位置を特定する

この手順では、水曜日の夜の増分バックアップに記録されたバイナリ・ログについてテンポラリ・ロケーションへのリストアのみを実行します。このプロセスにより、データベース管理者はログで、Orders テーブルが削除された時のマークが付けられた位置を見つけることができます。

- 1 [水曜日の夜からの増分バックアップを選択する]: データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定: データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]: このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - [テンポラリ・ディレクトリへのログをリストアし、時間あるいは位置を特定する]: 水曜日の夜の増分バックアップに含まれるバイナリ・ログ・ファイルについてリストアのみを実行するために、このオプションを選択します。
 - [時間に基づく PIT]: タイプとして選択します。ただし、[時間に基づく PIT の詳細] セクションのその他すべてのオプションは **選択解除** したままにします。

- 3 ジョブを開始して完了するまで待ちます。

手順5:リストアされたバイナリ・ログでDrop Tableコマンドの位置を特定する

[リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する]: リストアしない drop table SQL ステートメントの **位置** を特定するために、この手順を NetVault Backup の外で実行します (この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください)。この処理で、データベース管理者は、drop table ステートメントが、MySQL サーバーのテンポラリ・ロケーションにリストアされた **MYSQLSVR-bin.000009** バイナリ・ログの **805** の位置にあることを特定しました (また、これらの両方の値をメモしました)。

手順6:位置に基づくPITリストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 [水曜日の夜からの増分バックアップを選択する]: データベース管理者は再度 [リストア・ジョブ作成 - セーブセットの選択] タブで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定: データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]: このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - [テンポラリ・ディレクトリからのバイナリ・ログを適用する]: 一連の操作の最後の手順で、テンポラリ・ロケーションにリストアしたバイナリ・ログをターゲットにするために、このオプションを選択します。リストアしたバイナリ・ログ・ファイルで Drop Table コマンドが記録されている位置を特定したため、これと同じバイナリ・ログ・ファイルがプラグインで使用されるようにこのオプションを選択します。
 - [誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]: このオプションを選択し、[停止位置] を「804」(バイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の **前**にある位置) に設定します。[終了位置を含むバイナリ・ログ] ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル (MYSQLSVR-bin.000009) を選択しました。

- 3 ジョブを開始します。

方法2: リストアされたバイナリ・ログのみを使用して誤った/不良のSQLステートメントの**前**および**後**をリカバリする

データベース管理者は木曜日の午前9:00に、ユーザーがOrdersテーブルで「table not found (テーブルが見つかりません)」エラーに遭遇しているという通知を受けました。調査の結果、**水曜日の午後8:00**に開発者が無意識のうちにOrdersテーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される**直前**の特定時点までをリカバリすることを決定しました。また、Ordersテーブルが削除された時点の**後**の特定時点からバックアップされたバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。このプロセスを行うには、データベース管理者は、日曜日のフル・バックアップと、月曜日および火曜日に実行された増分バックアップをリストアしてから、水曜日に実行された増分バックアップを使用して位置に基づく PIT リカバリを実行する必要があります。以下にこのプロセスを示します。

手順1: 日曜日からのフル・リストア

- 1 日曜日の夜からのフル・バックアップを選択する: データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする: どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順2: 月曜日からの増分リストア

- 1 月曜日の夜からの増分バックアップを選択する: データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする: どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順3: 火曜日からの増分リストア

- 1 火曜日の夜からの増分バックアップを選択する: データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする: どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順4: バックアップされたバイナリ・ログをリストアして誤ったSQLステートメントの位置を特定する

この手順では、水曜日の夜の増分バックアップに記録されたバイナリ・ログについてテンポラリ・ロケーションへのリストアのみを実行します。この手順により、データベース管理者はログで、Ordersテーブルが削除された時のマークが付けられた位置を見つけることができます。

- 1 [水曜日の夜からの増分バックアップを選択する]: データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定: データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]: このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。

- **[テンポラリ・ディレクトリへのログをリストアし、時間あるいは位置を特定する]**：水曜日の夜の増分バックアップに含まれるバイナリ・ログ・ファイルについてリストアのみを実行するために、このオプションを選択します。
- **[時間に基づく PIT]**：タイプとして選択します。ただし、**[時間に基づく PIT の詳細]** セクションのその他すべてのオプションは**選択解除**したままにします。

3 ジョブを開始して完了するまで待ちます。

手順5: リストアされたバイナリ・ログで Drop Table コマンドの位置を特定する

[リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する]：リストアしない drop table SQL ステートメントの**位置**を特定するために、この手順を NetVault Backup の外で実行します（この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください）。この処理で、データベース管理者は、drop table ステートメントが、MySQL サーバーのテンポラリ・ロケーションにリストアされた **MYSQLSVR-bin.000009** バイナリ・ログの **805** の位置にあることを特定しました（また、これらの両方の値をメモしました）。

手順6: 位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 **[水曜日の夜からの増分バックアップを選択する]**：データベース管理者は再度 **[リストア・ジョブ作成 - セーブセットの選択]** タブで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の [オプション] タブのオプション設定**：データベース管理者は以下のオプションを選択します。
 - **[PIT リカバリを実行する]**：このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - **[テンポラリ・ディレクトリからのバイナリ・ログを適用する]**：一連の操作の最後の手順で、テンポラリ・ロケーションにリストアしたバイナリ・ログをターゲットにするために、このオプションを選択します。リストアしたバイナリ・ログ・ファイルで Drop Table コマンドが記録されている位置を特定したため、これと同じバイナリ・ログ・ファイルがプラグインで使用されるようにこのオプションを選択します。
 - **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]**：このオプションを選択し、**[停止位置]** を「**804**」（バイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の**前**にある位置）に設定します。**[終了位置を含むバイナリ・ログ]** ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル（**MYSQLSVR-bin.000009**）を選択しました。
 - **[誤った / 不良の SQL ステートメントの後に、リカバリを可能にする]**：このオプションを選択し、**[開始位置]** を「**806**」（バイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の**後**にある位置）に設定します。**[終了位置を含むバイナリ・ログ]** ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル（**MYSQLSVR-bin.000009**）を選択しました。最後に、指定したバイナリ・ログ・ファイルを最後までリカバリするために、**[開始日 / 時間]** オプションで **[なし]** ラジオ・ボタンを選択しました。

i | 重要: 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

3 ジョブを開始します。

方法3: リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った / 不良の SQL ステートメントの**前**をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前 6:00** に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が**木曜日の午前 6:00** にテーブルを削除した**直前**の特定時点までデータベースをリストアするようリカバリを実行する必要があります。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。このプロセスを行うには、データベース管理者は、日曜日のフル・バックアップと、月曜日および火曜日に実行された増分バックアップをリストアしてから、水曜日に実行された増分バックアップを使用して位置に基づく PIT リカバリを実行する必要があります。以下にこのプロセスを示します。

手順1: 日曜日からのフル・リストア

- 1 日曜日の夜からのフル・バックアップを選択する：データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順2: 月曜日からの増分リストア

- 1 月曜日の夜からの増分バックアップを選択する：データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順3: 火曜日からの増分リストア

- 1 火曜日の夜からの増分バックアップを選択する：データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順4: 現在のバイナリ・ログで Drop Table コマンドの位置を特定する

[リストアされたバイナリ・ログに対して `mysqlbinlog` ユーティリティを使用する]：リストアしない `drop table` SQL ステートメントの**位置**を特定するために、この手順を NetVault Backup の外で実行します（この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください）。この処理で、データベース管理者は `drop table` コマンドが現在のバイナリ・ログ `MYSQLSVR-bin.000009` の `805` の位置にあることを特定しました。

手順5: 位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 [水曜日の夜からの増分バックアップを選択する]：データベース管理者は再度 [リストア・ジョブ作成 – セーブセットの選択] タブで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定：データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]：このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - [バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）]：バックアップに含まれているバイナリ・ログ・ファイルをプラグインで使用するよう指定するために、このオプションを選択します。

- **【現在のバイナリ・ログを含む】**：データベース管理者はこのオプションを選択し、NetVault Backup で現在のバイナリ・ログも使用して、水曜日の夜の増分バックアップの**後**に実行されたすべてのデータベース・トランザクションを適用するよう指定します。この手順により、水曜日の夜に増分バックアップが完了してから drop table コマンドが発行されるまでの間に実行されたすべてのトランザクションがリカバリされます。
- **【誤った/不良の SQL ステートメントの前に、リカバリを可能にする】**：このオプションを選択し、**【停止位置】**を 804（現在のバイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の**前**にある位置）に設定します。**【終了位置を含むバイナリ・ログ】**ドロップダウンを**【OTHER FILE】**に設定し、現在のバイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。

方法4: リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った/不良の SQL ステートメントの**前**および**後**をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが Orders テーブルで「table not found（テーブルが見つかりません）」エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前 6:00**に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が**木曜日の午前 6:00**にテーブルを削除した**直前**の特定時点までデータベースをリストアするようリカバリを実行する必要があります。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。このプロセスを行うには、データベース管理者は、日曜日のフル・バックアップと、月曜日および火曜日に実行された増分バックアップをリストアしてから、水曜日に実行された増分バックアップを使用して位置に基づく PIT リカバリを実行する必要があります。以下にこのプロセスを示します。

手順1: 日曜日からのフル・リストア

- 1 日曜日の夜からのフル・バックアップを選択する：データベース管理者は**【リストア・ジョブ作成 - セーブセットの選択】**ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順2: 月曜日からの増分リストア

- 1 月曜日の夜からの増分バックアップを選択する：データベース管理者は**【リストア・ジョブ作成 - セーブセットの選択】**ページで、月曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順3: 火曜日からの増分リストア

- 1 火曜日の夜からの増分バックアップを選択する：データベース管理者は**【リストア・ジョブ作成 - セーブセットの選択】**ページで、火曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順4: 現在のバイナリ・ログで Drop Table コマンドの位置を特定する

【リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する】：リストアしない drop table SQL ステートメントの**位置**を特定するために、この手順を NetVault Backup の外で実行します（この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください）。この処理で、データベース管理者は drop table コマンドが現在のバイナリ・ログ **MYSQLSVR-bin.000009** の **805** の位置にあることを特定しました。

手順5: 位置に基づくPITリストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 **【水曜日の夜からの増分バックアップを選択する】**: データベース管理者は再度 **【リストア・ジョブ作成 - セーブセットの選択】** タブで、水曜日の夜に実行された増分バックアップに対応するバックアップ・セーブセットを選択します。
 - 2 **リストア関連の【オプション】タブのオプション設定**: データベース管理者は以下のオプションを選択します。
 - **【PIT リカバリを実行する】**: このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - **【バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）】**: バックアップに含まれているバイナリ・ログ・ファイルをプラグインで使用するよう指定するために、このオプションを選択します。
 - **【現在のバイナリ・ログを含む】**: データベース管理者はこのオプションを選択し、NetVault Backup で現在のバイナリ・ログも使用して、水曜日の夜の増分バックアップの**後**に実行されたすべてのデータベース・トランザクションを適用するよう指定します。この手順により、水曜日の夜に増分バックアップが完了してから drop table コマンドが発行されるまでの間に実行されたすべてのトランザクションがリカバリされます。
 - **【誤った / 不良の SQL ステートメントの前に、リカバリを可能にする】**: このオプションを選択し、**【停止位置】** を **804** (現在のバイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の**前**にある位置) に設定します。**【終了位置を含むバイナリ・ログ】** ドロップダウンを **【OTHER FILE】** に設定し、現在のバイナリ・ファイルの名前 (**MYSQSVR-bin.000009 など**) をテキスト・ボックスに入力しました。
 - **【誤った / 不良の SQL ステートメントの後に、リカバリを可能にする】**: このオプションを選択し、**【開始位置】** を **「806」** (現在のバイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の**後**にある位置) に設定します。**【終了位置を含むバイナリ・ログ】** ドロップダウンを **【OTHER FILE】** に設定し、現在のバイナリ・ファイルの名前 (**MYSQSVR-bin.000009 など**) をテキスト・ボックスに入力しました。最後に、現在のバイナリ・ログ・ファイルを最後までリカバリするために、**【停止位置】** オプションで **【なし】** ラジオ・ボタンを選択しました。
- i | 重要:** 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

フルおよび差分バックアップによるリストア・シナリオ

DBA はフル・バックアップを毎週**日曜日の午後 11 時**に、差分バックアップを**月～土曜の午後 11 時**に実行するバックアップ・ポリシーを作成しました。差分バックアップを実行しているため、この形式のバックアップを実行するたびにバイナリ・ファイルが保存されます。この場合、バックアップ時間は長くなりますが、全体的なリストア時間は短くなります。

フルおよび差分リストアのみ

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、データベース管理者の出勤前の木曜日午前に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、最後の差分バックアップ、つまり**水曜日の夜**に実行されたバックアップの時点までを完全にリカバリすることを決定しました。

手順1:日曜日からのフル・リストア

- 1 日曜日の夜からのフル・バックアップを選択する：データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順2:水曜日からの増分リストア

- 1 水曜日の夜からの差分バックアップを選択する：データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：[オプション] タブのどのオプションも使用しません。
- 3 ジョブを開始します。

i **重要:** データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません。差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。

フル・リストアと時間に基づく特定時点リカバリ

以降の例では、フル/差分バックアップ・シナリオを取り上げます。データベース管理者はデータを特定時点にリカバリしようと考えています。

方法1:リストアされたバイナリ・ログのみを使用して誤った/不良のSQLステートメントの前をリカバリする

データベース管理者は木曜日の午前9:00に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**水曜日の午後8:00**に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が**水曜日の午後8:00**にテーブルを削除した**直前**の特定時点までデータベースをリストアするようリカバリを実行する必要があります。このことから、以下の手順を実行します。

手順1:日曜日からのフル・リストア

- 1 日曜日の夜からのフル・バックアップを選択する：データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順2:水曜日からの時間に基づくPITリストア

- 1 水曜日の夜からの差分バックアップを選択する：データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

i **重要:** データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません。差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。

- 2 リストア関連の [オプション] タブのオプション設定：データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]：このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にします。
 - [バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）]：このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定します。
 - [時間に基づく PIT]：リストア・タイプとして選択します。
 - [誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]：このオプションを選択し、[中止日 / 時間] を「19:59」、「2007 年 1 月 10 日」（水曜日の日付の午後 8:00 の 1 分前）に設定します。
- 3 ジョブを開始します。

方法2:リストアされたバイナリ・ログのみを使用して誤った/不良のSQLステートメントの前および後をリカバリする

データベース管理者は木曜日の午前 9:00 に、ユーザーが Orders テーブルで「table not found (テーブルが見つかりません)」エラーに遭遇しているという通知を受けました。調査の結果、**水曜日の午後 8:00** に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、午後 8:00 に Drop Table コマンドが実行された**直前**の特定時点までリカバリすることを決定しました。また、Orders テーブルが削除された時点の**後**の特定時点からバックアップされたバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。このことから、以下の手順を実行します。

手順1:日曜日からのフル・リストア

- 1 日曜日の夜からのフル・バックアップを選択する：データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 すべてのリストア関連オプションをデフォルトのままにする：どのオプションも使用しません。
- 3 ジョブを開始して完了するまで待ちます。

手順2:水曜日からの時間に基づくPITリストア

- 1 水曜日の夜からの差分バックアップを選択する：データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

i **重要:** データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません。差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。

- 2 リストア関連の [オプション] タブのオプション設定：データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]：このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にします。
 - [バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）]：このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定します。
 - [時間に基づく PIT]：リストア・タイプとして選択します。

- **【誤った / 不良の SQL ステートメントの前に、リカバリを可能にする】**：このオプションを選択し、**【中止日 / 時間】** を「19:59」、「2007 年 1 月 10 日」（水曜日の日付の午後 8:00 の 1 分前）に設定します。
- **【誤った / 不良の SQL ステートメントの後に、リカバリを可能にする】**：**Order** テーブルの削除後に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より後の時刻と日付を **【開始日 / 時間】** オプションに入力します。最後に、リストアしたバイナリ・ログ・ファイルを最後までリカバリするために、**【開始日 / 時間】** オプションで **【なし】** ラジオ・ボタンを選択しました。

3 ジョブを開始します。

方法3: リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った / 不良の SQL ステートメントの前をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前 6:00** に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が**木曜日の午前 6:00** にテーブルを削除した**直前**の特定時点までデータベースをリストアするようリカバリを実行する必要があります。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする**：どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順2: 水曜日からの時間に基づく PIT リストア

- 1 **水曜日の夜からの差分バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

i 重要: データベース管理者は**月曜日**および**火曜日**の夜の差分バックアップをリストアする必要があります。**ありません。**差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。

- 2 **リストア関連の【オプション】タブのオプション設定**：データベース管理者は以下のオプションを選択します。
 - **【PIT リカバリを実行する】**：このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にします。
 - **【バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）】**：このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定します。
 - **【現在のバイナリ・ログを含む】**：水曜日にバックアップが完了してから Drop Table コマンドを発行するまでの間に発生したエントリを適用するために、このオプションを有効にします。
 - **【時間に基づく PIT】**：リストア・タイプとして選択します。
 - **【誤った / 不良の SQL ステートメントの前に、リカバリを可能にする】**：このオプションを選択し、**【中止日 / 時間】** を「5:59」、「2007 年 1 月 11 日」（月曜日の日付の午前 6:00 の 1 分前）に設定します。

3 ジョブを開始します。

方法4: リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った/不良のSQLステートメントの**前**および**後**をリカバリする

データベース管理者は木曜日の午前9:00に、ユーザーがOrdersテーブルで「table not found (テーブルが見つかりません)」エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前6:00**に開発者が無意識のうちにOrdersテーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される**直前**の特定時点までをリカバリすることを決定しました。また、Ordersテーブルが削除された時点の**後**の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。これにより、削除されたテーブルに加え、可能な限り多くのトランザクションをリカバリすることができます。このことから、以下の手順を実行します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順2: 水曜日からの時間に基づくPITリストア

- 1 **水曜日の夜からの差分バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

i 重要: データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません。差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。

- 2 **リストア関連の [オプション] タブのオプション設定** : データベース管理者は以下のオプションを選択します。
 - **[PIT リカバリを実行する]** : このオプションを選択して PIT リカバリを指定し、関連するすべてのオプションを有効にします。
 - **[バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)]** : このオプションを選択し、バックアップに含まれているバイナリ・ログを使用するよう指定します。
 - **[現在のバイナリ・ログを含む]** : 水曜日にバックアップが完了してから Drop Table コマンドを発行するまでの間に発生したエントリを適用するために、このオプションを有効にします。
 - **[時間に基づく PIT]** : リストア・タイプとして選択します。
 - **[誤った/不良の SQL ステートメントの前に、リカバリを可能にする]** : このオプションを選択し、[中止日/時間] を「5:59」、「2007年1月11日」(月曜日の日付の午前6:00の1分前)に設定します。
 - **[誤った/不良の SQL ステートメントの後に、リカバリを可能にする]** : **Order** テーブルの**削除後**に実行されたトランザクションをリカバリするために、このオプションを選択し、削除より**後**の時刻と日付を【開始日/時間】オプションに入力します。最後に、現在のバイナリ・ログ・ファイルを最後までリカバリするために、[中止日/時間] オプションで [なし] オプションを選択しました。
- 3 **ジョブを開始します**。

フル・リストアと位置に基づく特定時点リカバリ

以降の例では、フル/増分バックアップ・シナリオを取り上げます。データベース管理者は、より確実な方法で時刻を特定し、データを特定時点にリカバリしようとしています。このプロセスは、MySQL バイナリ・ログ・ファイル内で特定した位置の値を使用しています。

方法1: リストアされたバイナリ・ログのみを使用して誤った/不良のSQLステートメントの**前**をリカバリする

データベース管理者は木曜日の午前9:00に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**水曜日の午後8:00**に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される **直前**の特定時点までをリカバリすることを決定しました。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。以下にこのプロセスを示します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます**。

手順2: バックアップされたバイナリ・ログをリストアして誤ったSQLステートメントの位置を特定する

この手順では、水曜日の夜の差分バックアップに記録されたバイナリ・ログについてテンポラリー・ロケーションへのリストアのみを実行します。このプロセスにより、データベース管理者はログで、**Orders** テーブルが削除された時のマークが付けられた位置を見つけることができます。

- 1 **水曜日の夜からの差分バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **リストア関連の [オプション] タブのオプション設定** : データベース管理者は以下のオプションを選択します。
 - **[PIT リカバリを実行する]** : このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - **[テンポラリー・ディレクトリへのログをリストアし、時間あるいは位置を特定する]** : このオプションを選択し、水曜日の夜の差分バックアップに含まれるバイナリ・ログ・ファイルについてリストアのみを実行するよう指定します。
 - **[時間に基づく PIT]** : タイプとして選択します。ただし、**[時間に基づく PIT の詳細]** セクションのその他すべてのオプションは **選択解除** したままにします。
- 3 **ジョブを開始して完了するまで待ちます**。

手順3: リストアされたバイナリ・ログで Drop Table コマンドの位置を特定する

[リストアされたバイナリ・ログに対して **mysqlbinlog** ユーティリティを使用する] : リストアしない drop table SQL ステートメントの **位置** を特定するために、この手順を NetVault Backup の外で実行します (この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください)。この処理で、データベース管理者は、drop table ステートメントが、MySQL サーバーのテンポラリー・ロケーションにリストアされた **MYSQLSVR-bin.000009** バイナリ・ログの **805** の位置にあることを特定しました (また、これらの両方の値をメモしました)。

手順4: 位置に基づくPITリストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された差分バックアップを使用して PIT リストアを実行します。

- 1 **水曜日の夜からの差分バックアップを選択する** : データベース管理者は再度 [リストア・ジョブ作成 - セーブセットの選択] タブで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

i **重要**: データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません。差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。

- 2 **リストア関連の [オプション] タブのオプション設定** : データベース管理者は以下のオプションを選択します。
 - **[PIT リカバリを実行する]** : このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - **[テンポラリ・ディレクトリからのバイナリ・ログを適用する]** : 一連の操作の最後の手順で、テンポラリ・ロケーションにリストアしたバイナリ・ログをターゲットにするために、このオプションを選択します。リストアしたバイナリ・ログ・ファイルで Drop Table コマンドが記録されている位置を特定したため、これと同じバイナリ・ログ・ファイルがプラグインで使用されるようにこのオプションを選択します。
 - **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]** : このオプションを選択し、**[停止位置]** を「804」（バイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の *前*にある位置）に設定します。**[終了位置を含むバイナリ・ログ]** ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル（MYSQLSVR-bin.000009）を選択しました。

- 3 **ジョブを開始します。**

方法2: リストアされたバイナリ・ログのみを使用して誤った / 不良の SQL ステートメントの *前* および *後* をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが Orders テーブルで「table not found (テーブルが見つかりません)」エラーに遭遇しているという通知を受けました。調査の結果、**水曜日の午後 8:00** に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される *直前* の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の *後* の特定時点からバックアップされたバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。以下にこのプロセスを示します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順2: バックアップされたバイナリ・ログをリストアして誤った SQL ステートメントの位置を特定する

この手順では、水曜日の夜の増分バックアップに記録されたバイナリ・ログについてテンポラリ・ロケーションへのリストアのみを実行します。このプロセスにより、データベース管理者はログで、Orders テーブルが削除された時のマークが付けられた位置を見つけることができます。

- 1 水曜日の夜からの差分バックアップを選択する：データベース管理者は [リストア・ジョブ作成 – セーブセットの選択] ページで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。
- 2 リストア関連の [オプション] タブのオプション設定：データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]：このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - [テンポラリ・ディレクトリへのログをリストアし、時間あるいは位置を特定する]：このオプションを選択し、水曜日の夜からの差分バックアップに含まれるバイナリ・ログ・ファイルについてリストアのみを実行するよう指定します。
 - [時間に基づく PIT]：タイプとして選択します。ただし、[時間に基づく PIT の詳細] セクションのその他すべてのオプションは **選択解除** したままにします。
- 3 ジョブを開始して完了するまで待ちます。

手順3: リストアされたバイナリ・ログで Drop Table コマンドの位置を特定する

[リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する]：リストアしない drop table SQL ステートメントの **位置** を特定するために、この手順を NetVault Backup の外で実行します（この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください）。この処理で、データベース管理者は、drop table ステートメントが、MySQL サーバーのテンポラリ・ロケーションにリストアされた **MYSQLSVR-bin.000009** バイナリ・ログの **805** の位置にあることを特定しました（また、これらの両方の値をメモしました）。

手順4: 位置に基づくPITリストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された増分バックアップを使用して PIT リストアを実行します。

- 1 水曜日の夜からの差分バックアップを選択する：データベース管理者は再度 [リストア・ジョブ作成 – セーブセットの選択] タブで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。
 - ❗ **重要:** データベース管理者は月曜日および火曜日の夜の差分バックアップをリストアする必要がありません。差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。
- 2 リストア関連の [オプション] タブのオプション設定：データベース管理者は以下のオプションを選択します。
 - [PIT リカバリを実行する]：このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - [テンポラリ・ディレクトリからのバイナリ・ログを適用する]：一連の操作の最後の手順で、テンポラリ・ロケーションにリストアしたバイナリ・ログをターゲットにするために、このオプションを選択します。リストアしたバイナリ・ログ・ファイルで Drop Table コマンドが記録されている位置を特定したため、これと同じバイナリ・ログ・ファイルがプラグインで使用されるようにこのオプションを選択します。
 - [誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]：このオプションを選択し、[停止位置] を「804」（バイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の **前**にある位置）に設定します。[終了位置を含むバイナリ・ログ] ドロップダウンを使用して、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル（MYSQLSVR-bin.000009）を選択しました。
 - [誤った / 不良の SQL ステートメントの後に、リカバリを可能にする]：このオプションを選択し、[開始位置] を「806」（バイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の **後**にある位置）に設定します。[終了位置を含むバイナリ・ログ] ドロップダウンを使用し

て、テンポラリ・ディレクトリにリストアしたバイナリ・ログ・ファイル（`MYSQLSVR-bin.000009`）を選択しました。最後に、指定したバイナリ・ログ・ファイルを最後までリカバリするために、[停止位置] オプションで [なし] ラジオ・ボタンを選択しました。

i | **重要:** 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている**実際の位置**を指定してください。

3 ジョブを開始します。

方法3: リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った/不良のSQLステートメントの**前**をリカバリする

データベース管理者は木曜日の午前 9:00 に、ユーザーが **Orders** テーブルで「**table not found (テーブルが見つかりません)**」エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前 6:00** に開発者が無意識のうちに **Orders** テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、開発者が**木曜日の午前 6:00** にテーブルを削除した**直前**の特定時点までデータベースをリストアするようリカバリを実行する必要があります。また、開発者がテーブルを削除した時刻の推定以上に正確なリカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。以下にこのプロセスを示します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する** : データベース管理者は [リストア・ジョブ作成 - セーブセットの選択] ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする** : どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順2: 現在のバイナリ・ログで Drop Table コマンドの位置を特定する

[リストアされたバイナリ・ログに対して `mysqlbinlog` ユーティリティを使用する] : リストアしない drop table SQL ステートメントの**位置**を特定するために、この手順を NetVault Backup の外で実行します（この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください）。この処理で、データベース管理者は drop table コマンドが `MYSQLSVR-bin.000009` バイナリ・ログの **805** の位置にあることを特定しました。

手順3: 位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された差分バックアップを使用して PIT リストアを実行します。

- 1 **水曜日の夜からの差分バックアップを選択する** : データベース管理者は再度 [リストア・ジョブ作成 - セーブセットの選択] タブで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

i | **重要:** データベース管理者は**月曜日**および**火曜日**の夜の差分バックアップをリストアする必要が**ありません**。差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日の夜のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。

- 2 **リストア関連の [オプション] タブのオプション設定** : データベース管理者は以下のオプションを選択します。
 - **[PIT リカバリを実行する]** : このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
 - **[バイナリ・ログをリストアし適用する (時間あるいは位置が、既に判明している場合、使用される)]** : このオプションを選択し、バックアップに含まれていたバイナリ・ログ・ファイルをプラグインで使用するよう指定します。

- **【現在のバイナリ・ログを含む】**：このオプションを選択し、NetVault Backup で現在のバイナリ・ログを使用し、水曜日の夜の差分バックアップの**後**に実行されたすべてのデータベース・トランザクションを適用するよう指定します。この手順により、水曜日の夜に差分バックアップが完了してから drop table コマンドが発行されるまでの間に実行されたすべてのトランザクションがリカバリされます。
- **【誤った / 不良の SQL ステートメントの前に、リカバリを可能にする】**：このオプションを選択し、**【停止位置】** を 804（現在のバイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の**前**にある位置）に設定します。**【終了位置を含むバイナリ・ログ】** ドロップダウンを **【OTHER FILE】** に設定し、現在のバイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。

方法4: リストアされたバイナリ・ログと現在のバイナリ・ログを使用して誤った / 不良の SQL ステートメントの**前**および**後**をリカバリする

データベース管理者は木曜日の午前 9 : 00 に、ユーザーが **Orders** テーブルで「table not found（テーブルが見つかりません）」エラーに遭遇しているという通知を受けました。調査の結果、**木曜日の午前 6:00** に開発者が無意識のうちに Orders テーブルを削除したために、このテーブルが存在しなくなっていることが判明しました。

データベース管理者は、drop table コマンドが実行される**直前**の特定時点までをリカバリすることを決定しました。また、Orders テーブルが削除された時点の**後**の特定時点から現在のバイナリ・ログの最後まで、残りのテーブルに実行されたトランザクションをリカバリしようと考えました。また、開発者がテーブルを削除した時刻の推定以上に正確なりカバリを実行することを決定しました。このことから、データベース管理者は位置に基づくリカバリを使用することにしました。以下にこのプロセスを示します。

手順1: 日曜日からのフル・リストア

- 1 **日曜日の夜からのフル・バックアップを選択する**：データベース管理者は **【リストア・ジョブ作成 - セーブセットの選択】** ページで、日曜日の夜に実行されたフル・バックアップに対応するバックアップ・セーブセットを選択します。
- 2 **すべてのリストア関連オプションをデフォルトのままにする**：どのオプションも使用しません。
- 3 **ジョブを開始して完了するまで待ちます。**

手順2: 現在のバイナリ・ログで Drop Table コマンドの位置を特定する

【リストアされたバイナリ・ログに対して mysqlbinlog ユーティリティを使用する】：リストアしない drop table SQL ステートメントの**位置**を特定するために、この手順を NetVault Backup の外で実行します（この処理の手順およびこのユーティリティの使用方法については、『MySQL リファレンス・マニュアル』を参照してください）。この処理で、データベース管理者は drop table コマンドが現在のバイナリ・ログ **MYSQLSVR-bin.000009** の **805** の位置にあることを特定しました。

手順3: 位置に基づく PIT リストアを実行する

リストアされたバイナリ・ログで特定した位置をもとに、水曜日に実行された差分バックアップを使用して PIT リストアを実行します。

- 1 **水曜日の夜からの差分バックアップを選択する**：データベース管理者は再度 **【リストア・ジョブ作成 - セーブセットの選択】** タブで、水曜日の夜に実行された差分バックアップに対応するバックアップ・セーブセットを選択します。

i 重要：データベース管理者は**月曜日**および**火曜日**の夜の差分バックアップをリストアする必要が**ありません**。差分バックアップを実行するよう選択すると、毎晩のバックアップに日曜日のフル・バックアップ以降のバックアップが累積されます。つまり、水曜日の夜のバックアップには、日曜日のフル・バックアップ以降に生成された月曜日、火曜日、および水曜日のすべてのバイナリ・ログが含まれます。

- 2 **リストア関連の【オプション】タブのオプション設定**：データベース管理者は以下のオプションを選択します。

- **[PIT リカバリを実行する]**：このオプションを選択し、このリストア形式と、関連するすべてのオプションを有効にします。
- **[バイナリ・ログをリストアし適用する（時間あるいは位置が、既に判明している場合、使用される）]**：このオプションを選択し、バックアップに含まれていたバイナリ・ログ・ファイルをプラグインで使用するように指定します。
- **[現在のバイナリ・ログを含む]**：このオプションを選択し、NetVault Backup で現在のバイナリ・ログを使用し、水曜日の夜の差分バックアップの後に実行されたすべてのデータベース・トランザクションを適用するように指定します。この手順により、水曜日の夜に差分バックアップが完了してから drop table コマンドが発行されるまでの間に実行されたすべてのトランザクションがリカバリされます。
- **[誤った / 不良の SQL ステートメントの前に、リカバリを可能にする]**：このオプションを選択し、**[停止位置]** を 804（現在のバイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の *前*にある位置）に設定します。**[終了位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、現在のバイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。
- **[誤った / 不良の SQL ステートメントの後に、リカバリを可能にする]**：このオプションを選択し、**[開始位置]** を「806」（現在のバイナリ・ログの、mysqlbinlog で特定した Drop Table コマンドの位置の *後*にある位置）に設定します。**[終了位置を含むバイナリ・ログ]** ドロップダウンを **[OTHER FILE]** に設定し、現在のバイナリ・ファイルの名前（MYSQLSVR-bin.000009 など）をテキスト・ボックスに入力しました。最後に、現在のバイナリ・ログ・ファイルを最後までリカバリするために、**[停止位置]** オプションで **[なし]** ラジオ・ボタンを選択しました。

i | **重要:** 停止位置と開始位置には、不要なトランザクションの位置より大きい任意の数値ではなく、バイナリ・ログ・ファイルに記録されている *実際の位置* を指定してください。

MIXEDバイナリ・ログ形式が使用され、クロスデータベースの更新が発行された場合のPITリストア

i | **重要:** サイトにMIXEDバイナリ・ログ形式を使用していて、すべてのデータベース・ユーザーとプログラムがベストプラクティスに従い、変更されたテーブルが確実にUSEによって選択されたデータベース内にあり、かつクロスデータベースの更新が実行されない場合、このトピックはそのサイトに適用されません。（詳しくは、「[MIXEDバイナリ・ログ形式の使用](#)」を参照してください。）PITリストア・ジョブを実行すると、バイナリ・ログがジョブで選択されたデータベースの指定されたポイントに再生されます。

前述のように、環境内のユーザーとプログラムが、USEによって選択されていないデータベース内のテーブルを変更し、クロスデータベースの更新を実行する場合、PIT リストア・ジョブを実行したときに、指定の時間にトランザクションが再生されないことがあります。Quest では、すべてのデータベース・ユーザーとプログラムで変更されたテーブルが USE によって選択されたデータベース内に存在すること、およびクロスデータベースの更新が実行されないことを確認することをお勧めします。このガイドラインがご使用の環境に適していない場合は、Quest では、MIXED バイナリ・ログ形式を使用しないことをお勧めします。

i | **重要:** 次の手順では、`--database` オプションを指定せずに `mysqlbinlog` を使用します。したがって、バイナリ・ログのすべての内容が適用され、すべてのデータベースが変更される可能性があります。この手順を異なるMySQLサーバーに適用し、異なるMySQLサーバーから適切なデータを抽出することを検討してください。次の手順をプロダクションMySQLサーバーに適用すると、すべてのデータベースが指定したポイントにロールバックされます。すべてのMySQLサーバー・データベースを指定したポイントにロールバックする予定がない限り、プロダクション環境にこの手順を適用しないでください。

- 1 [ナビゲーション] パネルで、**[リストア・ジョブ作成]** をクリックします。
- 2 **[リストア・ジョブ作成 - セーブセットの選択]** ページで **[テーブル・フィルタリング]** をクリックし、**[フィルタの編集]** を選択します。
- 3 **[プラグイン・タイプ]** リストから **[Plug-in for MySQL]** を選択します。

- 4 セーブセット・テーブルで、バイナリ・ログとともに増分または差分バックアップを含むセーブセットを選択し、[次へ]をクリックします。
- 5 [セクション・セット作成] ページで、[バイナリ・ログ] を選択します。
バイナリ・ログは、すべての MySQL サーバー・データベースに共通です。
- 6 [セクション・セット作成] ページで、[プラグイン・オプションの編集] をクリックします。
- 7 [リカバリ時] タブで、[PIT リカバリを実行する] および [テンポラリ・ディレクトリへのログをリストアし、時間あるいは位置を特定する] オプションを選択します。
バイナリ・ログは、次の場所にあるテンポラリ・ディレクトリにリストアされます。
<NetVaultBackupInstallationDirectory>/tmp/mysql/<savesetName>
- 8 `mysqlbinlog` コマンド・プロンプトからバイナリ・ログを手動で適用するには、次のように入力します。

```
mysqlbinlog --stop-datetime="yyyy/mm/dd hh:mm:ss"  
"<NetVaultBackupInstallationDirectory>/tmp/mysql/<savesetName>" |  
mysql -u<user> -p<password>
```


例：

```
mysqlbinlog --stop-datetime="2018/06/06 15:09:00"  
"/usr/netvault/tmp/mysql/MySQL 59 - DIFF - DIFFERENTIAL (Saveset 86) 15.17 06  
Jun 2017/mysql-bin.000038" | mysql -uroot -p<password>
```
- 9 リストア・シーケンスに、リストアする必要がある複数の増分バックアップが含まれている場合は、増分バックアップごとにこの手順を繰り返します。
異なる増分バックアップ・セーブセットが、<NetVaultBackupInstallationDirectory>/tmp/mysql ディレクトリ内の異なるサブディレクトリにリストアされます。その後、`mysqlbinlog` コマンドを各ディレクトリに適用するか、すべてのバイナリ・ログを共通のディレクトリにコピーまたは移動して `mysqlbinlog` を実行できます。

MySQL Enterprise バックアップ用リストア・シナリオ例

障害またはデータ損傷からリカバリするには、ジョブの設定時に、リストア対象として選択するデータおよび [オプション] タブのオプションに関してさまざまな設定を行う必要があります。

- [フル・バックアップのみによるリストア・シナリオ](#)
- [フルおよび増分バックアップによるリストア・シナリオ](#)
- [TTS のみのリストア・シナリオ](#)
- [Linux および UNIX 環境での追加手順](#)

フル・バックアップのみによるリストア・シナリオ

- 1 リストアする準備済みフル・バックアップを生成するには、[オプション] タブで選択した [Raw フル・バックアップをリストア、抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成] オプションを選択したジョブを実行します。
- 2 MySQL をシャットダウンし、MySQL サーバー・リポジトリに準備済みバックアップをコピーするには、[オプション] タブで選択した [MySQL サーバーをシャットダウンし、準備済みフル・バックアップを MySQL サーバー・リポジトリへコピー・バック] オプションを選択したジョブを実行します。
- 3 コマンド・プロンプトで利用可能なコマンドを入力して MySQL を再起動します。

フルおよび増分バックアップによるリストア・シナリオ

- 1 リストアする準備済みフル・バックアップを生成するには、[オプション] タブで選択した [Raw フル・バックアップをリストア、抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成] オプションを選択したジョブを実行します。
- 2 必要な増分バックアップを準備済みフル・バックアップへ、それらがバックアップされた順に適用するには、[オプション] タブで選択した [増分バックアップをリストア、抽出し、Temp ディレクトリ内の準備済みフル・バックアップに適用] オプションを選択したジョブを実行します。
- 3 MySQL をシャットダウンし、MySQL サーバー・リポジトリに準備済みバックアップをコピーするには、[オプション] タブで選択した [MySQL サーバーをシャットダウンし、準備済みフル・バックアップを MySQL サーバー・リポジトリへコピー・バック] オプションを選択したジョブを実行します。
- 4 コマンド・プロンプトで利用可能なコマンドを入力して MySQL を再起動します。

TTSのみのリストア・シナリオ

- 1 リストアする準備済みフル・バックアップを生成するには、[オプション] タブで選択した [Raw フル・バックアップをリストア、抽出し、ログを適用して Temp ディレクトリ内に準備済みフル・バックアップを生成] オプションを選択したジョブを実行します。
- 2 準備済みフル・バックアップを MySQL サーバー・リポジトリにコピーし、1つのテーブルの名前を変更するには、次の場合にジョブを送信します。
 - オプションタブで、[MySQL サーバーをシャットダウンし、準備済みフル・バックアップを MySQL サーバー・リポジトリへコピー・バック] オプションを選択した。
 - [テーブルを含める] フィールドに正規表現パターンを入力し、リストアに含めるテーブルを指定した。
[テーブルを含める]フィールドの例:database_name\.
 - [テーブル名の変更] フィールドに名前変更のリクエストを入力した。
[テーブル名の変更]フィールドの例:original_name to new_name

LinuxおよびUNIX環境での追加手順

Linux または UNIX 環境で MySQL Enterprise バックアップを使用している場合は、リストアされたデータのファイル所有権および権限情報が、データのバックアップ前と一致していることを確認します。mysqlbackup スクリプトは、バックアップ・プロセス中にこの情報を記録しないため、リストアの完了後に情報が異なる場合があります。詳細は、「https://docs.oracle.com/cd/E17952_01/mysql-enterprise-backup-3.11-en/bugs.backup.html」を参照してください。

高度な MySQL Standard/Community 用 リストア手順

このトピックでは、[MySQL Standard/Community] オプション用プラグインを使用して実行することができるその他のリストア操作について説明します。

- リストア中にデータベース名を変更する
- 同じサーバー上の別の MySQL インスタンスへリストアする
- 異なる MySQL サーバーへのリストア

リストア中にデータベース名を変更する

NetVault Backup では、バックアップした MySQL データベースを選択し、リストア時に名前を変更して、既存のデータベースが上書きされないよう設定することができます。このプロセスは、既存のデータベースのコピーを作成する場合に便利です。このプロセスを実行するには、以下のトピックで説明する手順に従います。

i | 重要: リストアではデータベース全体の名称変更のみを行います。個別のテーブルの名称を変更しようとするとエラー・メッセージが表示されます。

リストア時の名前変更を実行する前に、この操作に関する以下の既知の制限事項と想定される用途のリストを確認する必要があります。

- 有効なリストア・シーケンスは、フル・バックアップまたは個々のデータベース/テーブル・コピーのみのバックアップに制限されます。
- 増分および差分リストアの実行中に行うことはできません。
- 別の MySQL インスタンスまたは MySQL サーバーへのリストアと組み合わせて使用できます。

1 [ナビゲーション] パネルで [リストア・ジョブ作成] をクリックして、[プラグイン・タイプ] リストから [Plug-in for MySQL] を選択し、適切なセーブセットを選択して [次へ] をクリックします。

詳細は、「[リストア対象データの選択](#)」を参照してください。

2 [セレクション・セット作成] ページで、名前を変更するデータベースをクリックし、コンテキスト・メニューから [名前の変更] を選択します。

3 [名前変更/移動] ダイアログ・ボックスの [名前変更] ボックスに新しい名前を入力して、[OK] をクリックします。

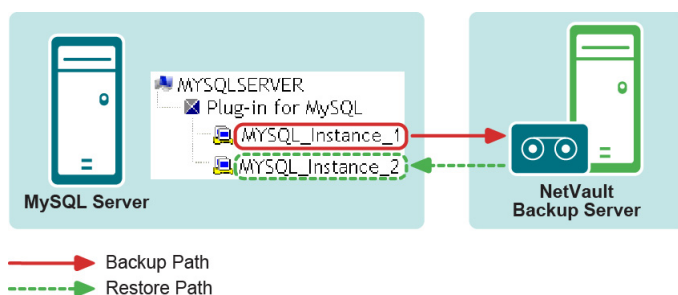
データベース・アイテムでは、変更された名前情報が括弧内に表示されます。

4 「[MySQL におけるデータのリストア](#)」の説明に従い、リストアを続行します。

同じサーバー上の別の MySQL インスタンスへリストアする

この形式の移動リストアで、Plug-in for MySQL のバックアップを、*同じ* MySQL サーバー・マシンであるが、そこで構成された MySQL の*別の*インスタンスへリストアします。

図 1. この形式のリストアでは、1つの MySQL インスタンスからバックアップされたデータを別のインスタンスへリカバリします。



このプロセスを実行するには、以下のトピックで説明する手順に従います。

既知の制限事項/想定される用途

移動リストアを実行する前に、この操作に関する以下の既知の制限事項と想定される用途のリストを確認する必要があります。

- 有効なリストア・シーケンスには、フル、増分、差分バックアップ、および個々のデータベース / テーブル・コピーのみのバックアップを含めることができます。
- リストア先インスタンスに適用できるのは、増分または差分バックアップからリストアされたバイナリ・ログのみです。つまり、ソース・インスタンスの現在のバイナリ・ログをリストア先インスタンスに適用することはできません。

前提条件

このタイプのリストアを設定して実行するには、以下の前提条件を満たす必要があります。

- **既存のマシンとターゲット・マシンのインストール構成を同一にする**：MySQL に関して、両方のマシンが以下のように設定されている必要があります。
 - 同じオペレーティング・システムがインストールされていること
 - 同じバージョンの MySQL がインストールされていること
- **Plug-in for MySQL で新しいターゲット・インスタンスで構成**：新しい MySQL インスタンスを追加するには、「[プラグインの設定](#)」で説明した手順を正しく実行する必要があります（ターゲット・インスタンスを **[NetVault Backup 選択]** ページの **[Plug-in for MySQL]** ノードの下で表示してアクセス可能にします）。

リストアの設定と開始

前提条件が満たされた状態で、以下の手順に従ってこの形式の移動リストア・ジョブを設定します。

- 1 [ナビゲーション] パネルで、**[リストア・ジョブ作成]** をクリックします。
- 2 **[リストア・ジョブ作成 – セーブセットの選択]** ページで、**[プラグイン・タイプ]** リストから **[Plug-in for MySQL]** を選択します。
- 3 セーブセットのテーブルに表示されている項目をさらにフィルタリングするには、**[クライアント]**、**[日付]**、**[ジョブ ID]** リストを使用します。

表にはセーブセット名（ジョブ・タイトルとセーブセット ID）、作成日時、およびサイズが表示されます。デフォルトで、リストは **[作成日]** 列でソートされます。

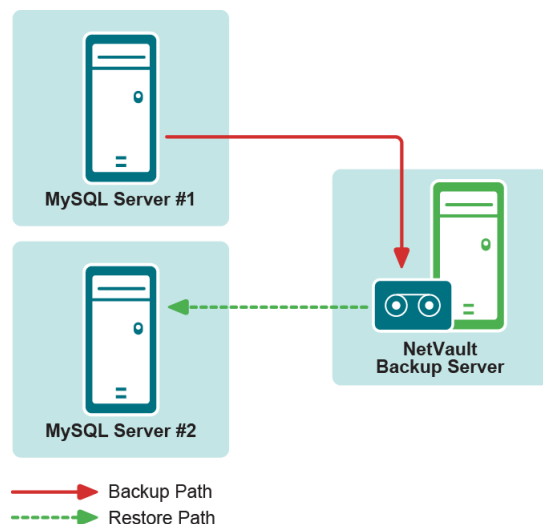
- 4 セーブセットの表で、適切な項目を選択します。
セーブセットを選択すると、以下の情報が **[セーブセット情報]** に表示されます。ジョブ ID、ジョブ・タイトル、サーバー名、クライアント名、プラグイン名、セーブセットの日時、リタイア設定、増分バックアップかどうか、アーカイブかどうか、セーブセットのサイズ、スナップショットベースのバックアップかどうかなど。
- 5 **[次へ]** をクリックします。
- 6 **[セレクション セット作成]** ページで、リストアするデータを選択します。
バックアップ対象だった個別の MySQL インスタンスを表示して、選択ツリーから目的のデータ・アイテムを探し、それらを選択します。
- 7 利用可能なデータベースを選択したら **[プラグイン・オプションの編集]** をクリックし、**[リストア先]** タブを選択します。
- 8 **[リストア先詳細]** セクションで、以下の情報を入力します。
 - **[ユーザー名]**：ターゲット MySQL インスタンスにアクセスするためのログイン・アカウントを入力します。
 - **[パスワード]**：ログイン・アカウントに関するパスワードを入力します。
 - **[インスタンス名]**：NetVault Backup の構成時の設定に基づいて、新しい MySQL ターゲット・インスタンスの NetVault Backup 名前を入力します。この名前は **[設定]** ダイアログ・ボックスの **[MySQL インスタンス名]** フィールドでインスタンスに設定した値です（詳しくは、「[プラグインの設定](#)」を参照してください）。

- 9 必要に応じて、**[リカバリ時]** タブで使用可能なその他のオプションを設定できますが、この形式のリストアを実行するために必ずしも必要ではありません。詳細は、「**リストア・オプションの設定**」を参照してください。
- 10 設定を保存するには、**[OK]**、続いて **[次へ]** をクリックします。
- 11 デフォルト設定を使用しない場合は、**[ジョブ名]** に、ジョブの名前を指定します。
- 進捗状況を監視する際にジョブを識別しやすくするため、分かりやすい名前を割り当てます。ジョブ名には英数字と英数字以外の文字を使用できますが、非ラテン語系の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。
- i** | **重要:** ターゲットOSのファイル名としてサポートされていない特殊文字を使用しないよう注意してください。たとえば、Windowsでは、\、\、*、@などの文字は使用できません。これは、Plug-in for MySQLがデータを一時的にリストアするために、**[ジョブ・タイトル]**と同じ名前のフォルダを作成しようとするからです。
- 12 **[クライアント指定]** リストで、データをリストアするマシンを選択します。
- i** | **ヒント:** **[選択]**をクリックして、**[クライアント指定選択]**ダイアログ・ボックスから適切なクライアントを検索、選択することもできます。
- 13 **[スケジュール]**、**[ソース・オプション]**、および **[詳細設定]** リストを使って、その他の必要なオプションを設定します。
- 14 **[保存]** または **[保存 & 実行]** の、どちらか適切な方をクリックします。
- [ジョブ・ステータス]** ページで進捗状況を監視したり、**[ログ参照]** ページでログを表示したりできます。詳しくは、『Quest NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

異なる MySQL サーバーへのリストア

プラグインでデータベースまたは個々のテーブルを同じ MySQL サーバー上の別の MySQL インスタンスにリストアできるのと同様に、リストア・プロセス中に**別の MySQL サーバー**をターゲットにすることができます。このオプションは、災害復旧オペレーションで使用されます。

図 2. この形式の移動リストアのデータ・パス例



このプロセスを実行するには、以下のトピックで説明する手順に従います。

既知の制限事項/想定される用途

リストアを別の MySQL サーバーに移動する前に、このオペレーションに関する以下の既知の制限事項と想定される用途のリストを確認する必要があります。

- 有効なリストア・シーケンスには、フル、増分、差分バックアップ、および個々のデータベース / テーブル・コピーのみのバックアップを含めることができます。
- リストア先インスタンスに適用できるのは、増分または差分バックアップからリストアされたバイナリ・ログのみです（ソース・インスタンスの現在のバイナリ・ログをリストア先の MySQL インスタンスに適用することはできません）。

ソフトウェアのインストール/設定の前提条件

このタイプのリストアを設定して実行するには、以下の前提条件を満たす必要があります。

- **既存のマシンとターゲット・マシンのインストール構成を同一にする**：MySQL に関して、両方のマシンが以下のように設定されている必要があります。
 - 同じオペレーティング・システムがインストールされていること
 - 同じバージョンの MySQL がインストールされていること
 - インストールとベース・ディレクトリが同一であること
 - MySQL データ・ディレクトリが同一であること
- **NetVault Backup ソフトウェアおよび Plug-in for MySQL をすべてのクライアントにインストールする**：NetVault Backup（クライアントまたはサーバー・バージョン）およびプラグインを、このプロセスで使用する **両方**のクライアント・マシン（**既存の MySQL マシン**と**新しいリストア・ターゲット**）にインストールし、設定する必要があります。
- **すべてのクライアント・マシンを NetVault Backup サーバーに追加する**：すべてのソフトウェア・インストール条件を満たした上で、NetVault Backup WebUI を使用して、ターゲットの NetVault Backup クライアント・マシン（**既存の MySQL マシン**および**新しいリストア・ターゲット**）を NetVault Backup サーバーに追加する必要があります。
- **新しいリストア・ターゲットに MySQL インスタンスが必ず存在する**：**新しいリストア・ターゲット**に MySQL インスタンスが存在している必要があります。このインスタンスが移動リストアのターゲットとなります。このインスタンスは MySQL で正しく設定および構成されるとともに、新しいリストア・ターゲットでプラグインのインストールに追加されている必要があります（「[プラグインの設定](#)」で説明した手順に従います）。

i **重要**: 新しいリストア・ターゲットのインスタンスの**[設定]**ダイアログ・ボックスから、以下の値をメモしておきます。**ユーザ名**、**パスワード**および**インスタンス名**

移動リストアの設定中、プラグインでターゲット MySQL インスタンスへの適切なアクセスを得られるよう、これらの値を**[オプション]**タブ内のフィールドに入力する必要があります。

リストアの実行

すべての前提条件を満たしたら、以下の手順に従って、MySQL のバックアップを別のマシンにリストアします。

- 1 [ナビゲーション] パネルで、**[リストア・ジョブ作成]** をクリックします。
- 2 **[リストア・ジョブ作成 - セーブセットの選択]** ページで、**[プラグイン・タイプ]** リストから **[Plug-in for MySQL]** を選択します。
- 3 セーブセットのテーブルに表示されている項目をさらにフィルタリングするには、**[クライアント]**、**[日付]**、**[ジョブ ID]** リストを使用します。

表にはセーブセット名（ジョブ・タイトルとセーブセット ID）、作成日時、およびサイズが表示されます。デフォルトで、リストは**[作成日]**列でソートされます。

- 4 セーブセットの表で、適切な項目を選択します。

セーブセットを選択すると、以下の情報が【**セーブセット情報**】に表示されます。ジョブ ID、ジョブ・タイトル、サーバー名、クライアント名、プラグイン名、セーブセットの日時、リタイア設定、増分バックアップかどうか、アーカイブかどうか、セーブセットのサイズ、スナップショットベースのバックアップかどうかなど。

- 5 【**次へ**】 をクリックします。

- 6 【**セレクション セット作成**】 ページで、リストアするデータを選択します。

バックアップ対象だった個別の MySQL インスタンスを表示して、選択ツリーから目的のデータ・アイテムを探し、それらを選択します。

- 7 利用可能なデータベースを選択したら【**プラグイン・オプションの編集**】 をクリックし、【**リストア先**】 タブを選択します。

- 8 【**リストア先詳細**】 セクションで、以下の情報を入力します。

- 【**ユーザー名**】: **新しいリストア・ターゲット**のターゲット・インスタンスに設定したユーザー名を入力します（【**設定**】 ダイアログ・ボックスの【**ユーザー名**】 フィールドで設定した名前）。
- 【**パスワード**】: **新しいリストア・ターゲット**のターゲット・インスタンスに設定したパスワードを入力します（【**設定**】 ウィンドウの【**パスワード**】 フィールドで設定したパスワード）。
- 【**インスタンス名**】: **新しいリストア・ターゲット**の MySQL インスタンスの NetVault Backup 名を入力します（【**設定**】 ウィンドウの【**MySQL インスタンス名**】 フィールドでインスタンスに設定した値）。

- 9 必要に応じて、【**リカバリ時**】 タブで使用可能なその他のオプションを設定できますが、

この形式のリストアを実行するために必ずしも必要ではありません。詳細は、「**リストア・オプションの設定**」を参照してください。

- 10 設定を保存するには、【**OK**】、続いて【**次へ**】 をクリックします。

- 11 デフォルト設定を使用しない場合は、【**ジョブ名**】 に、ジョブの名前を指定します。

進捗状況を監視する際にジョブを識別しやすくするため、分かりやすい名前を割り当てます。ジョブ名には英数字と英数字以外の文字を使用できますが、非ラテン語系の文字を含めることはできません。Linux の場合、名前は最大で 200 文字です。Windows の場合、長さ制限はありません。ただし、すべてのプラットフォームで、40 文字以内にするをお勧めします。

i | **重要:** ターゲットOSのファイル名としてサポートされていない特殊文字を使用しないよう注意してください。たとえば、Windowsでは、\、\、*、@などの文字は使用できません。これは、Plug-in for MySQLがデータを一時的にリストアするために、【**ジョブ・タイトル**】と同じ名前のフォルダを作成しようとするからです。

- 12 【**クライアント指定**】 リストで、データをリストアするマシンを選択します。

i | **ヒント:** 【**選択**】 をクリックして、【**クライアント指定選択**】ダイアログ・ボックスから適切なクライアントを検索、選択することもできます。

- 13 【**スケジュール**】、【**ソース・オプション**】、および【**詳細設定**】 リストを使って、その他の必要なオプションを設定します。

- 14 【**保存**】 または【**保存 & 実行**】 の、どちらか適切な方をクリックします。

【**ジョブ・ステータス**】 ページで進捗状況を監視したり、【**ログ参照**】 ページでログを表示したりできます。詳しくは、『Quest NetVault Backup アドミニストレーターズ・ガイド』を参照してください。

MySQL レプリケーションの使用

- MySQL レプリケーション環境でのプラグイン使用：概要
- レプリケーションのサポートの有効化
- レプリケーション・サーバーのバックアップ
- レプリケーション・サーバーのリストア

MySQL レプリケーション環境でのプラグイン使用：概要

レプリケーションを使用するときは、複製されたテーブルに対するすべての更新をマスタ・サーバーで実行する必要があります。これを行わない場合は、ユーザーがマスタ上のテーブルに対して行った更新と、スレーブ上のテーブルに対して行った更新との間に矛盾が生じないように常に注意する必要があります。

レプリケーションにより、堅牢性、速度、およびシステム管理上の利点をもたらされます。

- マスタおよびスレーブ設定により、堅牢性が向上します。マスタで問題が発生した場合、バックアップとして機能しているスレーブに切り替えることができます。
- クライアント・クエリの処理負荷をマスタ・サーバーとスレーブ・サーバーに分散することにより、クライアントへの応答時間を短縮できます。SELECT クエリをスレーブに送信し、マスタのクエリ処理負荷を軽減することもできます。ただし、マスタとスレーブが同期から外れないようにするために、データを変更するステートメントについては、マスタに送信する必要があります。このロード・バランス戦略は、更新以外のクエリが中心となる場合に効果的ですが、実際はこれが標準的なケースです。
- レプリケーションのもう1つの利点は、マスタに負荷をかけることなく、スレーブ・サーバーを使用してデータベース・バックアップを実行できることです。バックアップ中も、マスタは継続して更新を処理します。

Plug-in for MySQL は、単一マスタ・レプリケーション環境のバックアップとリカバリをサポートしています。

レプリケーションのサポートの有効化

レプリケーションのサポートは、**[設定]** ダイアログ・ボックスを使って有効にします。このダイアログ・ボックスへのアクセスについては、「**プラグインの設定**」を参照してください。

- **[MySQL レプリケーションを可能にする]**：このインスタンスに対してネイティブの MySQL レプリケーションを有効化するには、このチェック・ボックスを選択します。
 - **[スレーブ・インスタンス]**：MySQL レプリケーションを有効化したインスタンスをスレーブ・インスタンスとして設定する場合、このオプションを選択します。
 - **[マスタ・インスタンス]**：MySQL レプリケーションを有効化したインスタンスをマスタ・インスタンスとして設定する場合、このオプションを選択します。

- [特定時点リカバリを可能にする]: 特定時点バックアップおよびリストアを有効化した場合、このチェック・ボックスを選択します。
- [バイナリ・ログ・インデックス・パス]: [特定時点リカバリを可能にする] チェック・ボックスを選択した場合、このフィールドを使用してバイナリ・ログ・インデックス・ファイルへのフル・パスを指定します。
- [リレー・ログ・インデックス・パス]: [スレーブ・インスタンス] を設定中の場合、このフィールドを利用してリレー・ログ・インデックス・ファイルへのフル・パスを入力し、バックアップに含めます。

レプリケーション・サーバーのバックアップ

MySQL レプリケーション環境のバックアップは、以下の制限付きでサポートされます。

- スレーブ・レプリケーション・サーバー: サポートされるバックアップ・タイプは以下のとおりです。
 - フル
 - 増分
 - 差分
 - 個々のデータベース/テーブル・コピーのみ
- マスター・レプリケーション・サーバー: サポートされるバックアップ・タイプは以下のとおりです。
 - 個々のデータベース/テーブル・コピーのみ

スレーブ・サーバーで増分および差分バックアップを実行するには、MySQL で `--log-slave-updates` オプションを有効にする必要があります。このオプションは、スレーブの SQL スレッドによって実行された更新をそれ自体のバイナリ・ログに記録するようスレーブに指示するものです。このオプションを動作させるには、バイナリ・ログを有効にする `--log-bin` オプションを使用してスレーブを起動する必要があります。通常、このオプションはレプリケーション・サーバーを連結するために使用されますが、バイナリ・ログ・バックアップに対して使用すると、マスター・サーバー上のバイナリ・ログをスレーブへの適用前にパージするといった面倒なオペレーションを行わなくても、レプリケーション環境で PIT リカバリが可能になります。

レプリケーション設定のバックアップ

[リレー・ログ・インデックス・パス] オプションを使用することにより、リレー・ログ・インデックス・ファイルへのフル・パスを指定し、バックアップに含めることができます。デフォルトでは、ステータス・ファイル `master.info` および `relay-log.info` は、同一の場所に配置されています。[リレー・ログ・インデックス・パス] オプションを使用し、デフォルトのファイル名および位置を指定した場合、これらのファイルすべてがバックアップされ、プラグインにより自動的にスレーブ・レプリケーション・サーバーにリストアされます。

レプリケーション・サーバーのリストア

MySQL レプリケーション・スレーブ・インスタンスのフル、増分および差分バックアップを使用して、MySQL レプリケーション・マスター・インスタンスの災害復旧を実行することができます。マスターのリストア後、同じバックアップ・セットを使用して各スレーブ・インスタンスをマスター・インスタンスと同レベルにリストアして、レプリケーションを再開したり、『MySQL Reference Guide』で説明されているその他の初期化方法でスレーブを再度初期化することができます。

マスターおよびスレーブ双方の個々のデータベース/テーブル・バックアップを使用して、個々のデータベース/テーブルをマスターにリストアすることができます。スレーブ上の個々のテーブルまたはデータベースを再同期する場合、Quest ではスレーブにリストアしてからスレーブとマスターを同期するよう配慮するのではなく、MySQL のレプリケーション・プロセスを利用して再同期することをお勧めします。

フェイルオーバー・クラスタ環境でのプラグインの使用

- MySQL サーバー・フェイルオーバー・クラスタリング：概要
- プラグインのインストールまたはアップグレード
- プラグインの設定
- データのバックアップ
- データのリストア

MySQL サーバー・フェイルオーバー・クラスタリング：概要

MySQL フェイルオーバー・クラスタリング（アクティブ/パッシブ）は、MySQL サーバー・インスタンス全体の高可用性を確保することを目的に設計されています。たとえば、フェイルオーバー・クラスタの1つのノードで、ハードウェア障害またはオペレーティング・システム障害が発生した場合、あるいは計画されたアップグレードを行う際に、クラスタ内の別のノードにフェイルオーバーするよう MySQL サーバー・インスタンスを設定できます。

フェイルオーバー・クラスタは、1つ以上のノード（ホスト）と1つ以上の共有ディスクで構成されます。IP アドレス、共有ストレージ、およびアプリケーション（この場合、MySQL）などのノードによってホスティングされるさまざまなリソースを組み合わせることで**クラスタ・サービス**と呼ばれるグループを構成します。ネットワーク上では、仮想サービスはアプリケーションを実行中の単一のコンピュータとして認識されますが、現在のノードが使用不可になった場合はノード間でのフェイルオーバーが可能です。

Plug-in for MySQL は、MySQL サーバー・フェイルオーバー・クラスタリングをサポートしています。プラグインはフェイルオーバー・クラスタ・ネットワーク名によって、MySQL サーバー・クラスタ・サービスを管理している現在のノードを特定し、それをバックアップ対象とすることができます。

このトピックでは、プラグインの設定と使用が、フェイルオーバー・クラスタ環境と従来の環境でどのように異なるかを説明します。このトピックは、**[MySQL Standard/Community]** オプションを説明するセクションを反映させるため、以下のような構成になっています。

重要な注意事項

- 以降のトピックで説明していない場合、本プラグインを使用したクラスタ・データのバックアップおよびリストア手順は、従来の MySQL サーバー・データのバックアップおよびリストア手順と同様です。
- 以降のトピックでは、フェイルオーバー・クラスタ環境で本プラグインを使用する場合に必要な、MySQL 固有の設定についてのみ説明しています。NetVault Backup の**アプリケーション・クラスタ・サポート**を使用して MySQL サーバー以外の関連データ/ファイルのバックアップ/リストアを管理する設定手順については説明していません。このプロセスは本プラグイン固有のものではありません。この手順について詳しくは、『Quest NetVault Backup アドミニストレータズ・ガイド』を参照してください。

- 次のセクションに進む前に、『Quest NetVault Backup アドミニストレーターズ・ガイド』のすべてのクラスタ関連情報を確認し、ここで説明する内容が MySQL サーバー・フェイルオーバー・クラスタ機能とどのように関連しているかを理解しておいてください。

プラグインのインストールまたはアップグレード

本プラグインをインストールするには、以下のトピックに従います。

インストールの前提条件

Plug-in for MySQL をクラスタ環境にインストールするには、以下の前提条件を満たす必要があります。

- **MySQL フェイルオーバー・クラスタ環境を展開する**：正しく構成された MySQL クラスタ環境が必要です。
 - i 重要**：この機能のサポートはRed Hat ClusteringおよびClustered Storage Suiteを使用するRed Hat Enterprise Linux (RHEL) v5.x上でテストされ、データベースのデータ・ファイルおよびログを含む共有ストレージとMySQL (v5.5) 2ノード・クラスタ構成を採用します。クラスタリング機能を異なる構成で使用する場合、実際の運用環境に配置する前に、バックアップおよびリストアをテストします。
- **NetVault Backup サーバー・マシンを別に用意する**：NetVault Backup サーバーとして使用するマシンが適切に設定されている必要があります。このマシンは、MySQL サーバー・クラスタの**外部**に設置し、クラスタ内のノード（ホスト）へのネットワーク接続を行う必要があります。

ソフトウェアのインストール

クラスタ環境へのプラグインのインストールは、従来のインストール方法と同じです。詳しくは、「[プラグインのインストールと削除](#)」を参照してください。

プラグインの設定

プライマリ・ノード上で以下の手順を実行します。

- 1 NetVault Backup サーバーの NetVault Backup WebUI の [ナビゲーション] パネルで、[バックアップ・ジョブ作成] をクリックして、次に [セクション] リストの隣りにある [新規作成] をクリックします。
- 2 セクション・ツリー内で、プライマリ・ノードを開きます。
- 3 **Plug-in for MySQL** を開きます。
- 4 [すべてのインスタンス] ノードをクリックし、コンテキスト・メニューから [設定] を選択します。
- 5 [設定] ダイアログ・ボックスで、利用可能な設定オプションを設定します。

使用可能なオプションは、「[プラグインの設定](#)」で説明するオプションと同じです。

- i 重要**：[設定] ダイアログ・ボックスの [インスタンス名] フィールドに各クラスタ・インスタンスを追加します。インスタンスを追加するには、MySQL クラスタ・サービス名を VIRTUAL SERVER NAME\INSTANCE NAME の形式で指定します。

- 6 その他のバックアップ・ジョブの作成や、セカンダリ・ノード上で既存バックアップ・ジョブの変更が見込まれる場合は、以下の手順を実行します。
 - a プライマリ・ノードをセカンダリ・ノードにフェイルオーバーします。
 - b **ステップ 1** から **ステップ 5** を繰り返します。
 - c プライマリ・ノードにフェイル・バックします。
- 7 設定を保存するには、**[OK]** をクリックします。

データのバックアップ

[NetVault Backup 選択] ページで Plug-in for MySQL ノードを開き、バックアップ対象にする MySQL サーバー仮想サーバー（またはそれに含まれるアイテム）を選択します。

実際には、このページに表示されるインスタンス名は、MySQL クラスタ化サービスです。このノードでほかの MySQL クラスタ化サービスが稼働している場合、これらのインスタンスも Plug-in for MySQL ノード内に表示されます。このような他のインスタンスのデータをバックアップ対象として**選択しない**よう注意してください。

- i** **メモ：**データのバックアップまたはリストアを実行する際は、プライマリ・ノードを使用してプロセスを実行します。ノードを展開して階層をドリル・ダウンしていくと、MySQL クラスタ化サービスが表示され、ノードがアクティブかどうかによって、さらにアイテムをドリル・ダウンして選択することができます。ログ情報を管理する目的で、システムがこのインスタンスを使用している可能性があるため、このレベルではいかなる処理も実行しないでください。

データのリストア

Plug-in for MySQL を使用したリストアに使用可能なすべてのオプションは、フェイルオーバー・クラスタ環境でも使用できます。また、データも同様の方法で選択します。唯一の違いは、**[リストア・ジョブ作成 — セーブセットの選択]** ページで、リストア可能なバックアップが、各バックアップ中に使用していたプライマリ・ノードの下に表示される点です。リストア・ジョブを開始すると、NetVault Backup はすべてのメンバー・クライアントと通信し、フェイルオーバー・クラスタを管理しているマシンを特定し、このマシンをリストア対象として指定します。

記載されているすべてのリストア実行手順は、フェイルオーバー・クラスタのリカバリでも使用できます。詳しくは、「**データのリストア**」トピックを参照してください。フェイルオーバー・クラスタをスタンドアロンの NetVault Backup クライアントにリストアするには、「**異なる MySQL サーバーへのリストア**」セクションで説明している手順に従います。

トラブルシューティング

このトピックでは一般的なエラーとその解決方法について説明します。この表に記載されていないエラーが発生した場合は、NetVault Backup ログから MySQL エラー番号を取得し、MySQL のドキュメンテーションで関連するトラブルの解決手段を参照してください。

表2. トラブルシューティング

エラー・メッセージ	説明
<ul style="list-style-type: none"> バックアップ・レコードの追加に失敗しました バックアップ・インデックスをデータベースに書き込むことができませんでした 	<p>これらのメッセージは、選択されたデータのバックアップは完了したものの、NetVault Backupによってジョブのインデックス情報がデータベースに適切に追加されなかったことを示します。このインデックス情報が追加されていないと、データは正しくリストアされません。</p> <p>方法1:</p> <p>NetVault Backup WebUIの[デバイス管理]ページに移動して、ジョブが対象とするメディアのスキャンを実行します。NetVault Backupでは、バックアップ・ジョブのインデックス情報はNetVaultデータベースとバックアップ対象メディアの双方に保存されます。このスキャンを実行することで、インデックス情報が、NetVaultデータベースに書き込まれます。情報が追加されたことを確認するには、[リストア・ジョブ作成 — セーブセットの選択]ページを開いて対象のジョブを見つけます。ジョブが参照でき、リストア・ジョブも設定できる場合には、スキャン処理によってこの問題は解決されました。</p> <p>方法2:</p> <p>方法1が失敗した場合は、バックアップ・ジョブを再実行します。</p>
バックアップがレプリケーションエラーを伴って失敗しました。	<p>「レプリケーション・スレーブ・サーバーの開始に失敗しました」などのメッセージが表示されてバックアップが失敗する場合は、[MySQLレプリケーションを可能にする]チェックボックスは選択したが、レプリケーションを設定していない可能性があります。この問題を修正するには、[設定]ダイアログ・ボックスの[MySQLレプリケーションを可能にする]チェック・ボックスをクリアするか、レプリケーションを設定してからバックアップ・ジョブを再実行する必要があります。設定の更新について詳しくは、「プラグインの設定」を、レプリケーションについて詳しくは、「MySQLレプリケーションの使用」を参照してください。</p>
LinuxまたはUNIX環境では、バックアップまたはリストア・ジョブが次のエラーで失敗します。 MySQLサーバーへの接続を確立できません。接続のオープンに失敗し、「Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2) (ソケット /tmp/mysql.sockを介してローカルMySQLサーバーに接続できません (2))」というエラーが表示されません。	<p>このジョブはMySQLサーバーのソケット・ファイルのデフォルトの場所 /tmp/mysql.sockにアクセスしようとしていますが、ファイルは別の場所にあります。このファイルは、/var/lib/mysql/mysql.sockまたは /opt/mysql/mysql.sockなどの場所にある場合があります。この問題に対処するには、次のコマンドを使用してシンボリック・リンクを作成し、ジョブがソケット・ファイルにアクセスできるようにします。</p> <pre>ln -s <existingFile> <symbolicLinkFile></pre> <p>パスおよびファイル名のアップデートの詳細については、「プラグインの設定」を参照してください。</p>

Quest は、急速に変化する企業 IT の世界にソフトウェア・ソリューションを提供します。データの急増、クラウドの拡張、ハイブリッド・データセンター、セキュリティの脅威、規制要件によって生じる課題を簡素化することができます。弊社は、Fortune 500 の 95% の企業および Global 1000 の 90% の企業など、100 か国におよぶ 130,000 社に対するグローバル・プロバイダーです。1987 年以來、データベース管理、データ保護、ID およびアクセス管理、Microsoft のプラットフォーム管理、統合エンドポイント管理などのソリューションのポートフォリオを構築してきました。Quest により、組織は IT 管理に費やす時間を短縮し、ビジネスの革新に費やす時間を増やすことができます。詳しくは、以下を参照してください。<http://www.bakbone.co.jp>

テクニカル・サポート用リソース

テクニカル・サポートは、Quest の有効な保守契約を締結している場合、または試用版を保有している場合にご利用いただけます。Quest サポート・ポータル (<https://support.quest.com/ja-jp>) にアクセスすることができます。

サポート・ポータルには、問題を自主的にすばやく解決するために使用できるセルフヘルプ・ツールがあり、24 時間 365 日ご利用いただけます。サポート・ポータルでは次のことを実行できます。

- サービス・リクエストの送信と管理。
- ナレッジベース記事の参照。
- 製品に関するお知らせへの登録。
- ソフトウェアと技術文書のダウンロード。
- 入門ビデオの閲覧。
- コミュニティ・ディスカッションへの参加。
- サポート・エンジニアとのオンライン・チャット。
- 製品に関する支援サービスの表示。