

MySQL 용 Quest® NetVault® Backup 플러그  
인- 12.2

사용 설명서



© 2019 Quest Software Inc.

**ALL RIGHTS RESERVED.**

본 안내서에는 저작권으로 보호되는 독점 정보가 수록되어 있습니다. 본 안내서에서 설명하는 소프트웨어는 소프트웨어 라이선스 또는 비공개 계약에 의거하여 제공됩니다. 이 소프트웨어는 해당 계약의 규정을 준수하는 방법으로만 사용 또는 복사할 수 있습니다. 본 안내서의 일부 또는 전부를 Quest Software Inc.의 서면 허가 없이 구매자의 개인적인 용도 이외의 다른 용도로 복제 또는 전송하는 행위는 금지되며, 여기에는 복사 및 녹화를 포함한 모든 전자 또는 기계적인 수단을 사용한 모든 형태의 복제 또는 전송이 포함됩니다.

본 문서에 수록된 정보는 Quest Software 제품과 관련하여 제공됩니다. 본 문서에 의해, 또는 Quest Software 제품의 판매와 관련하여서는 지적 재산권에 대한 금반언 원칙 등에 따른 명시적이거나 암묵적인 라이선스가 부여되지 않습니다. 본 제품에 대한 라이선스 계약에 규정된 약관에 명시된 경우를 제외하고, Quest Software는 자사 제품의 상업성, 특정 용도에 대한 적합성 또는 비침해에 대한 암묵적인 보증 등을 포함해 해당 제품에 관한 모든 명시적이거나 암묵적이거나 법적인 보증에 대해 어떤 책임도 지지 않고 부인합니다. Quest Software는 본 문서를 사용하거나 사용할 수 없음으로 인해 발생하는 (이익 손실, 사업 중단 또는 정보 손실을 제한 없이 포함하는) 모든 직접적, 간접적, 필연적, 징벌적, 특수 또는 부수적 피해에 대해 Quest Software가 해당 피해가 발생할 가능성에 대해 알고 있었을 경우를 포함한 어떤 경우에도 책임을 지지 않습니다. Quest Software는 본 문서의 정확성 또는 완전성에 관해 어떤 진술 또는 보증도 하지 않으며, 언제든지 예고 없이 사양 및 제품 설명을 변경할 권리를 보유합니다. Quest Software는 본 문서에 수록된 정보의 업데이트에 대해 어떠한 약속도 하지 않습니다.

본 자료의 잠재적인 사용에 대해 궁금한 사항이 있으면 다음 연락처로 문의하십시오.

Quest Software Inc.  
귀중: 법무 부서  
4 Polaris Way  
Aliso Viejo, CA 92656

지역 및 해외 사무소에 대한 정보는 당사 웹 사이트(<https://www.quest.com>)를 참조하십시오.

**특허권**

Quest Software는 당사의 고급 기술에 대해 자부심을 갖고 있습니다. 특허권과 출원 중인 특허권이 이 제품에 적용될 수 있습니다. 이 제품에 적용된 특허권에 대한 최신 정보는 당사의 웹사이트 <https://www.quest.com/legal>에서 확인할 수 있습니다.

**상표**

Quest, Quest 로고, NetVault는 Quest Software Inc.에 등록된 상표입니다. 모든 Quest 상표 목록은 <https://www.quest.com/legal/trademark-information.aspx>에서 확인할 수 있습니다. 기타 모든 상표와 등록된 상표는 해당 소유자의 자산입니다.

**법례**

■ **경고:** 경고 아이콘은 재산 손실, 신체적 상해 또는 사망을 초래할 수 있는 위험이 있음을 알려 줍니다.

! **주의:** 주의 아이콘은 지침을 준수하지 않으면 하드웨어 손상이나 데이터 손실의 위험이 있음을 알려 줍니다.

i **중요 참고, 참고, 팁, 휴대폰 또는 비디오:** 정보 아이콘은 지원 정보를 알려 줍니다.

MySQL용 NetVault Backup 플러그인- 사용 설명서  
업데이트됨 - 2 2019  
소프트웨어 버전 - 12.2  
MYG-101-12.2-KO-01

# 목차

<b>MySQL용 NetVault Backup 플러그인- 소개</b> .....	<b>5</b>
MySQL용 NetVault Backup 플러그인-: 개요 .....	5
주요 이점 .....	5
기능 요약 .....	6
대상 독자 .....	7
권장된 추가 문서 .....	7
<b>플러그인 설치 및 제거</b> .....	<b>8</b>
설치 필수 구성 요소 .....	8
MySQL Server에서 바이너리 로그 활성화(Standard/Community 옵션 전용) .....	9
권장 구성 검토 .....	10
플러그인 설치 또는 업그레이드 .....	10
플러그인 제거 .....	11
특정 MySQL 인스턴스 제거 .....	11
<b>플러그인 구성</b> .....	<b>12</b>
기본 설정 구성 .....	12
기존 인스턴스 구성 업데이트 .....	15
오류 조건에 대한 기본 작업 설정(선택 사항) .....	16
<b>데이터 백업</b> .....	<b>17</b>
데이터 백업: 개요 .....	17
백업 전략 정의 .....	20
백업 수행 .....	24
백업 데이터 선택 .....	24
백업 옵션 설정 .....	25
작업 마무리 및 제출 .....	28
<b>데이터 복원</b> .....	<b>29</b>
데이터 복원: 개요 .....	29
MySQL Standard/Community에 사용 가능한 복원 방법 검토 .....	29
MySQL Enterprise Backup에 사용 가능한 복원 옵션 검토 .....	30
MySQL에서 데이터 복원 .....	31
복원 데이터 선택 .....	31
복원 옵션 설정 .....	32
작업 마무리 및 제출 .....	37
MySQL Standard/Community 복원 시나리오 예시 .....	38
MySQL Enterprise Backup 복원 시나리오 예시 .....	60

MySQL Standard/Community에 대한 고급 복원 절차 사용 .....	61
복원 중에 데이터베이스 이름 바꾸기 .....	62
동일한 서버에서 다른 MySQL 인스턴스로 복원 .....	62
대체 MySQL Server로 복구 .....	64
<b>기본 MySQL 복제 작업 .....</b>	<b>67</b>
기본 환경에서 플러그인 사용: 개요 .....	67
복제 지원 활성화 .....	67
복제 서버 백업 .....	68
복제 구성 백업 .....	68
복제 서버 복원 .....	68
<b>장애 조치 클러스터 환경에서 플러그인 사용 .....</b>	<b>69</b>
MySQL Server 장애 조치 클러스터링: 개요 .....	69
플러그인 설치 또는 업그레이드 .....	70
설치 필수 구성 요소 .....	70
소프트웨어 설치 .....	70
플러그인 구성 .....	70
데이터 백업 .....	71
데이터 복원 .....	71
<b>문제 해결 .....</b>	<b>72</b>
기술 지원 리소스 .....	73

# MySQL 용 NetVault Backup 플러그인- 소개

- MySQL 용 NetVault Backup 플러그인-: 개요
- 주요 이점
- 기능 요약
- 대상 독자
- 권장된 추가 문서

## MySQL 용 NetVault Backup 플러그인-: 개요

MySQL 용 Quest® NetVault® Backup 플러그인- MySQL 용 플러그인-은 복잡한 스크립팅 없이 여러 MySQL 저장소 엔진의 백업 및 복구를 단일 작업으로 통합합니다. **mysql Enterprise Backup** 옵션(MEB 기반 방법)을 사용하는 경우, 플러그인이 백업 중에 InnoDB 테이블의 핫 백업을 지원합니다. **MySQL Standard/Community** 옵션 (mysqldump 기반 방법)을 사용하는 경우 플러그인은 모든 테이블의 워م 백업을 지원하며 데이터를 읽기 전용 액세스 권한이 있는 온라인 상태로 유지합니다. 또한 **MySQL Standard/Community** 옵션을 사용하면 플러그인은 보다 세분화된 복원을 수행할 수 있는 향상된 특정 시점(PIT) 기능을 제공하므로 정확한 시점으로 복원하고 데이터 손실을 줄일 수 있습니다.

## 주요 이점

- **MySQL 배포 중 신뢰도 향상 및 위험 감소:** MySQL 용 플러그인-은 복잡한 백업 스크립트를 생성할 필요가 없으며 많은 복구 시나리오를 처리할 수 있을 정도로 충분히 유연합니다. MySQL 내부를 이해하지 않고도 백업 중 커밋된 트랜잭션의 손실을 방지하는 백업 정책을 구현하고 바이너리 로그를 삭제할 시기를 파악할 수 있습니다. 이러한 정보는 플러그인에 내장되어 있습니다.

MySQL 용 플러그인-의 유연한 백업 기능에는 다음과 같은 사항도 포함됩니다.

- 데이터가 온라인 상태이며 액세스할 수 있는 동안 전체, 증분 및 차등 백업
- 여러 저장소 엔진에 공통된 사용자 인터페이스
- 테이블 및 보기 수준까지 보호
- 여러 저장소 엔진을 단일 작업으로 통합

플러그인을 사용하여 백업 정책을 구현하면 장애 발생시 필요한 내용을 복구하는 기능이 위험에 처하지 않고 더 중요한 작업에만 집중할 수 있습니다. 또한 MySQL Server 데이터가 보호되고 있다는 것을 인지함으로써 IT 관리자의 신뢰성이 향상됩니다.

- **복원 시간 단축으로 가동 중지 시간 감소:** 복원할 대상, 복원할 백업 집합, 해당하는 경우 복원할 시점 또는 위치, 복원을 자동으로 수행하는 플러그인을 선택합니다. 사용자 상호 작용이 최소화되어 복원 속도가 빨라지고 구문 오류가 발생할 가능성이 제거됩니다.

MySQL용 플러그인-의 추가 복원 기능에는 다음 사항이 포함됩니다.

- 전체, 증분 및 시간 기반/위치 기반 PIT 복원
- 전체 인스턴스, 개별 데이터베이스 또는 개별 테이블 및 보기 복원
- 복원 중에 데이터베이스 이름 바꾸기
- 대체 MySQL 인스턴스로 복원
- **비즈니스 지속성 보장:** 오프 사이트 백업은 업무상 중요한 응용 프로그램을 위한 데이터 보호의 중요한 부분입니다. 플러그인은 다양한 백업 장치와의 NetVault Backup 통합을 활용합니다. NetVault Backup에서 백업을 저장할 백업 장치를 선택할 수 있습니다. VTL(가상 테이프 라이브러리)에 백업을 온라인으로 저장할 수 있습니다. 또한 여러 MySQL 인스턴스, 기타 전용 데이터베이스 또는 일반 백업 파일을 통해 공유하는 물리적 테이프 라이브러리에 작업을 복제할 수도 있습니다.
- **고급 MySQL 복제 기술 지원 - MySQL 참조 안내서**에 설명된 대로 MySQL은 단방향 비동기 복제를 지원합니다. 이 경우 하나의 서버가 마스터 역할을 하는 반면 다른 서버는 슬레이브로 작동합니다.

단일 마스터 복제에서는 마스터 서버가 바이너리 로그에 업데이트를 작성하고 해당 파일의 인덱스를 유지하여 로그 회전을 추적합니다. 바이너리 로그는 임의의 슬레이브 서버에 전송될 업데이트 레코드 역할을 합니다. 슬레이브가 마스터에 접속하면 슬레이브가 마지막 업데이트 시 로그를 읽은 지점까지 마스터에 알립니다. 슬레이브는 그 시간 이후 발생한 모든 업데이트를 수신한 다음 마스터가 새로운 업데이트를 알릴 때까지 차단하고 대기합니다.

MySQL용 플러그인-에서는 MySQL 환경이 보호되고 재해 복구를 위해 오프사이트에 저장된다는 확신을 심어줍니다. 이와 동시에 관리자는 하루 24시간 연중 무휴로 자신의 업무에 전념할 수 있습니다. 경험이 부족한 직원이 복원을 시작할 수 있으므로 가동 중지 시간을 줄이고 비즈니스 연속성을 개선할 수 있습니다.

## 기능 요약

- 클러스터 지원 MySQL Server 5.6을 기반으로 하는 MySQL Cluster Network Database(NDB) 7.x를 지원합니다. 이 기능은 mysqldump 유틸리티를 백업 방법으로 사용합니다.
- **MySQL Standard/Community** 옵션을 사용하여 다음 지원을 제공합니다.
  - 전체 및 증분 백업
  - 차등 백업
  - 개별 데이터베이스/테이블 복사 전용 백업
  - InnoDB, MyISAM, 병합(MRG\_MyISAM이라고도 함), 메모리/힙, 페더레이션, Berkeley DB(BDB), 아카이브 및 CSV 저장소 엔진
  - 저장소 엔진에 공통된 사용자 인터페이스
  - 시간 및 위치 기반 PIT 복원
  - 데이터 손상 전 후의 PIT 복원
  - 개별 테이블이나 데이터베이스 또는 전체 인스턴스 복원
  - 복원 중에 데이터베이스 이름 바꾸기
  - 대체 인스턴스로 복원
  - 기본 MySQL 복원 슬레이브 및 마스터 인스턴스 백업

- **MySQL Enterprise Backup** 옵션을 사용하여 다음 지원을 제공합니다.
  - 전체 및 증분 백업
  - InnoDB, MyISAM, 병합(MRG\_MyISAM이라고도 함), 아카이브 및 CSV 저장소 엔진
  - InnoDB 테이블의 핫 백업
  - TTS(Transportable Tablespace) 백업
  - 저장소 엔진에 공통된 사용자 인터페이스
  - 개별 테이블이나 데이터베이스 또는 전체 인스턴스 복원
  - TTS 백업 복원 중 단일 테이블의 이름 바꾸기
- 포인트 앤 클릭 WebUI

## 대상 독자

고급 MySQL 데이터베이스 관리자(DBA) 기술은 일상적인 백업 작업을 생성하고 실행할 필요가 없으므로 효율적인 백업 및 복구 전략을 정의하고 고급 복구 시나리오를 수행하는 데 필요합니다.

## 권장된 추가 문서

이 플러그인을 설정하고 사용할 때 참조할 수 있도록 Quest에서는 다음 설명서를 준비해 두는 것을 권장합니다.

- **MariaDB 설명서:** <https://mariadb.com/kb/en/library/documentation/>
- **MySQL <X> 참조 안내서**(여기서 <X>는 MySQL Server에 설치된 MySQL의 버전을 나타냄):
  - **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 설명서:**
  - *Quest NetVault Backup 설치 안내서:* 이 안내서에서는 NetVault Backup 서버 및 클라이언트 소프트웨어를 설치하는 방법에 대한 세부 정보를 제공합니다.
  - *Quest NetVault Backup 관리자 안내서:* 이 안내서에서는 NetVault Backup을 사용하는 방법 및 모든 플러그인에 공통적으로 적용되는 기능에 대해 설명합니다.
  - *Quest NetVault Backup CLI 참조 안내서:* 이 안내서는 명령줄 유틸리티를 설명합니다.

<https://support.quest.com/technical-documents>에서 이러한 안내서를 다운로드할 수 있습니다.

## 플러그인 설치 및 제거

- 설치 필수 구성 요소
- 권장 구성 검토
- 플러그인 설치 또는 업그레이드
- 플러그인 제거
- 특정 MySQL 인스턴스 제거

### 설치 필수 구성 요소

MySQL 용 플러그인-을 설치하기 전에 MySQL Server로 사용할 시스템에 다음과 같은 소프트웨어가 설치되고 적절히 구성되었는지 확인합니다.

- **NetVault Backup 서버 및 클라이언트 소프트웨어:** 최소한 NetVault Backup 클라이언트가 MySQL Server로 구성된 시스템에 설치되어야 합니다.
- **MySQL 데이터베이스 소프트웨어**
- **MySQL Server에서 바이너리 로그 활성화(MySQL Standard/Community 옵션 전용)** 이 설정은 MySQL Server의 특정 시점(PIT) 백업 및 복원을 지원합니다. 자세한 내용은 [MySQL Server에서 바이너리 로그 활성화\(Standard/Community 옵션 전용\)](#) 항목을 참조하십시오.
- **MySQL 데이터베이스 클라이언트 패키지의 적절한 버전:** 플러그인은 MySQL 클라이언트 패키지와 함께 설치된 구성 요소와 상호 작용하여 플러그인을 통해 더 많은 기능에 액세스할 수 있도록 합니다. 이 패키지와 함께 설치된 구성 요소의 버전은 설치된 MySQL 버전과 호환되어야 합니다. 기본적으로 이 두 가지 MySQL 구성 요소를 설치하고 해당 버전을 확인해야 합니다.
  - **mysqldump:** 이 유틸리티를 사용하면 여러 유형의 MySQL 저장소 엔진에 대한 백업 및 복원을 수행할 수 있습니다. 이 구성 요소의 버전이 MySQL의 현재 버전과 호환되고 이전 버전의 MySQL 용 플러그인-과 함께 제공된 버전이 **아닌지** 확인하십시오.
  - **mysqlbinlog:** 이 유틸리티를 사용하면 PIT 백업 및 복원을 사용할 수 있습니다. 설치된 MySQL 버전에서 이 구성 요소의 적절한 버전을 사용할 수 있는지 확인합니다.
- **MySQL Enterprise Backup:** 독립 실행형(비 클러스터) 환경에서 **MySQL Enterprise Backup** 옵션(MEB 기반 방법)을 사용하려면 사용자 환경이 다음 요구 사항을 충족해야 합니다.
  - Windows, Linux 및 UNIX 환경의 경우 MySQL Server가 버전 5.6 또는 5.7을 사용해야 합니다.
  - MySQL Enterprise Backup 제품의 버전 3.12 또는 4.0이 설치되어야 합니다. MySQL Enterprise Backup은 MySQL Enterprise Edition 및 특정 상용 버전에서 사용할 수 있습니다. 설치 지침은 MySQL Enterprise Edition 제품에 해당하는 설명서를 참조하십시오.
    - Windows Server 2008 R2, Windows Server 2012 R2, Windows Server 2016 및 RHEL(Red Hat Enterprise Linux) 7.x에서 MySQL 5.6과 함께 MEB 버전 3.12 옵션을 사용할 수 있습니다.
    - Windows Server 2008 R2, Windows 2012 R2, Windows Server 2016, RHEL 6.x 및 7.x에서 MySQL 5.7과 함께 MEB 버전 4.0 옵션을 사용할 수 있습니다.

- 중요:** Windows Server 2008 R2 또는 Windows Server 2016을 사용하는 경우 Windows 컴퓨터 관리를 사용하여 로컬 계정 및 관리자 그룹 구성원을 시스템의 로컬 관리자 그룹에 추가할 수 있습니다.

RHEL 6.x를 사용하는 경우 계속하기 전에 라이브러리가 최신 상태인지 확인하십시오.

## MySQL Server에서 바이너리 로그 활성화 (Standard/Community 옵션 전용)

MySQL Standard/Community 옵션에서 PIT 백업 및 복원에 대한 지원을 구성하기 전에 MySQL 바이너리 로그를 활성화해야 합니다.

### Linux 또는 UNIX 기반 MySQL Server에서 로그 활성화

- MySQL 설치 디렉터리에 액세스하고, "**my.cnf**" 등의 MySQL 구성 파일을 찾습니다.  
파일의 이름과 위치는 MySQL 구성에 따라 다릅니다. 자세한 내용은 관련 MySQL 설명서를 참조하십시오.
- 텍스트 편집기를 사용하여 파일을 열고 "**[mysqld]**" 섹션을 찾습니다.
- 기본 MySQL 디렉터리를 사용하여 MySQL 바이너리 로그를 저장하려면 다음 항목을 추가하십시오.

log-bin

- 중요:** 원하는 경우 **my.cnf** 파일에 추가된 "**log-bin**" 항목을 다음 구문을 사용해 설정하고, 바이너리 로그를 저장할 다른 파일을 지정할 수 있습니다.

**log-bin=<NameOfDestinationFile>**

바이너리 로그의 대상 파일 이름을 지정할 때 파일 자체의 이름만 사용합니다. 전체 경로 정보 또는 파일 확장명을 포함하지 마십시오. *MySQL 참조 안내서*를 참조하여 바이너리 로그 활성화에 대한 자세한 내용을 확인한 후 플러그인 설치를 계속하십시오.

- 변경을 적용하려면 MySQL Server를 다시 시작합니다.

### Windows 기반 MySQL Server에서 로그 활성화

- MySQL 관리자** 응용 프로그램을 시작하고 자세한 내용은 관련 MySQL 설명서를 참조하십시오.
  - 중요:** MySQL 관리자가 설치되어 있지 않으면 Linux 또는 UNIX 시스템에서 구성 파일을 업데이트한 다음 MySQL 서비스를 중지했다가 다시 시작하여 바이너리 로그를 활성화합니다.
- MySQL 관리자** 창의 왼쪽 영역에서 **시작 변수**를 클릭합니다.
- 오른쪽 창에서 **로그 파일** 탭을 선택합니다.
- 바이너리 로그 파일 이름** 확인란을 선택하고 파일의 고유한 이름을 입력하거나, **log-bin**의 기본값이 사용되도록 필드를 비워 둡니다.

- 중요:** 바이너리 로그의 대상 파일 이름을 지정할 때 파일 자체의 이름만 사용합니다. 전체 경로 정보 또는 파일 확장명을 포함하지 마십시오. *MySQL 참조 안내서*를 참조하여 바이너리 로그 활성화에 대한 자세한 내용을 확인한 후 플러그인 설치를 계속하십시오.

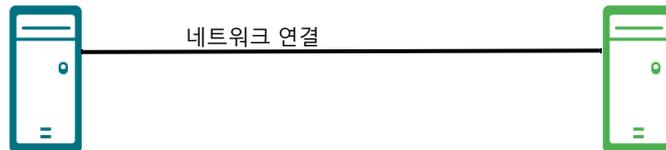
- MySQL 관리자** 응용 프로그램을 종료합니다.

6 변경을 적용하려면 MySQL Server를 다시 시작합니다.

## 권장 구성 검토

단일 시스템을 NetVault Backup Server 및 MySQL Server로 설정할 수 있습니다. 즉, 모든 소프트웨어 설치 및 구성 요구 사항이 단일 시스템에서 수행되므로 Quest에서는 이러한 두 엔터티가 **별도의** 시스템에 존재하도록 권장합니다.

표 1. 권장 구성



MySQL Server 시스템	NetVault Backup Server 시스템
<b>소프트웨어 설치 및 구성</b> <ul style="list-style-type: none"> <li>• MySQL 데이터베이스 소프트웨어(5.5 이상)</li> <li>• NetVault Backup Server 및 클라이언트 소프트웨어</li> <li>• MySQL용 플러그인-</li> <li>• 바이너리 로그 활성화됨(MySQL Standard/Community 옵션 전용)</li> <li>• mysqldump/mysqlbinlog 유틸리티—설치된 MySQL 버전과 호환 가능</li> <li>• mysqlbackup 유틸리티—MySQL Enterprise Backup 옵션 전용</li> </ul>	<b>소프트웨어 설치 및 구성</b> <ul style="list-style-type: none"> <li>• NetVault Backup Server 소프트웨어</li> <li>• MySQL Server가 NetVault Backup 클라이언트로 추가됨—NetVault Backup Server에 클라이언트 시스템을 추가하는 것에 대한 자세한 내용은 <i>Quest NetVault Backup 관리자 안내서</i>를 참조하십시오.</li> </ul>

**i** | **중요:** 이 안내서 전체의 샘플 이미지 및 절차는 사용자가 **2대의 시스템 환경**을 사용 중이며 구성 요구 사항이 충족되었다고 가정합니다.

## 플러그인 설치 또는 업그레이드

1 NetVault 구성 마법사 또는 클라이언트 관리 페이지에 액세스합니다.

**i** | **참고:** 선택한 클라이언트가 모두 동일한 유형인 경우 구성 마법사를 사용하여 동시에 여러 클라이언트에 플러그인을 설치할 수 있습니다. 여러 클라이언트를 선택하는 경우 플러그인 바이너리 파일이 대상 클라이언트의 OS 및 플랫폼과 호환되는지 확인하십시오. **클라이언트 관리** 페이지에서 플러그인 설치를 위해 하나의 클라이언트만 선택할 수 있습니다.

- NetVault 구성 마법사 페이지에 액세스하려면 다음을 수행하십시오.
  - a 탐색 창에서 **구성 안내**를 클릭합니다.
  - b NetVault 구성 마법사 페이지에서 **플러그인 설치**를 클릭합니다.
  - c 다음 페이지에서 해당 클라이언트를 선택합니다.

- **클라이언트 관리** 페이지에 액세스하려면 다음을 수행하십시오.
  - a 탐색 창에서 **클라이언트 관리**를 클릭합니다.
  - b **클라이언트 관리** 페이지에서 해당 시스템을 선택하고 **관리**를 클릭합니다.
  - c **클라이언트 보기** 페이지에서 **플러그인 설치** 단추(+)를 클릭합니다.
- 2 **플러그인 파일 선택**을 클릭하고 플러그인에 대한 ".npk" 설치 파일의 위치(예: 설치 CD 또는 웹 사이트에서 파일을 다운로드한 디렉터리)로 이동합니다.  
 사용 중인 운영 체제(OS)에 따라 이 소프트웨어의 경로는 설치 CD에서 다를 수 있습니다.
- 3 "mys-x-x-x-x.npk"라는 파일을 선택하고(여기서 xxxxx는 버전 번호 및 플랫폼을 나타냄) **열기**를 클릭합니다.
- 4 설치를 시작하려면 **플러그인 설치**를 클릭합니다.  
 플러그인이 설치되면 메시지가 표시됩니다.

## 플러그인 제거

- 1 탐색 창에서 **클라이언트 관리**를 클릭합니다.
- 2 **클라이언트 관리** 페이지에서 해당 클라이언트를 선택하고 **관리**를 클릭합니다.
- 3 **클라이언트 보기** 페이지의 **설치된 소프트웨어** 표에서 **MySQL용 플러그인**- 항목을 선택하고 **플러그인 제거** 단추(-)를 클릭합니다.
- 4 **확인** 대화 상자에서 **제거**를 클릭합니다.

## 특정 MySQL 인스턴스 제거

MySQL 인스턴스가 성공적으로 구성되어 플러그인에 추가된 후에 이 인스턴스를 제거할 수도 있습니다.

**i | 중요:** 주의하여 이 절차를 따르십시오. 그러나 **플러그인 구성**의 단계에 따라 인스턴스를 다시 추가할 수 있습니다.

- 1 탐색 창에서 **백업 작업 생성**을 클릭하고 **선택** 목록 옆에 있는 **새로 만들기**를 클릭합니다.
- 2 선택 트리에서 적용 가능한 클라이언트 노드를 엽니다.
- 3 **MySQL용 플러그인**-을 엽니다.
- 4 해당 인스턴스를 클릭하고 상황에 맞는 메뉴에서 **서버 제거**를 선택합니다.  
 이 명령을 사용한 후에는 확인 대화 상자가 표시되지 않습니다.

## 플러그인 구성

- 기본 설정 구성
- 오류 조건에 대한 기본 작업 설정(선택 사항)

### 기본 설정 구성

MySQL 용 플러그인-은 단일 MySQL Server에서 여러 MySQL 인스턴스를 지원하며, 각 개별 인스턴스는 사용할 수 있도록 구성되어야 합니다. 사용 가능한 구성 옵션은 MySQL Server에서 사용 중인 OS 및 **MySQL Standard/Community** 옵션을 사용하는지 아니면 **MySQL Enterprise Backup** 옵션을 사용하는지에 따라 달라집니다.

- 1 탐색 창에서 **백업 작업 생성**을 클릭하고 **선택** 목록 옆에 있는 **새로 만들기**를 클릭합니다.
- 2 선택 트리에서 적용 가능한 클라이언트 노드를 엽니다.
- 3 **MySQL 용 플러그인-**을 클릭하고 상황에 맞는 메뉴에서 **새 서버 추가**를 선택합니다.
- 4 **구성** 대화 상자에서 해당 필드를 완성합니다.
  - **MySQL 인스턴스 이름**(필수): MySQL 인스턴스에 해당하는 이름을 입력합니다. 기본적으로 로컬 호스트 이름이 사용됩니다. 이 값은 NetVault Backup WebUI에서 사용됩니다. Quest에서는 MySQL 인스턴스가 있는 시스템과 관련된 이름을 사용하지 않고 일반적인 이름을 MySQL 인스턴스에 사용할 것을 적극 권장합니다. 일반적인 이름을 사용하면 모든 영향을 받는 클라이언트에서 이식성 및 정책 관리가 개선됩니다.
 

예를 들어 로컬 호스트 이름이 `test_mysql_01_machine`이고, MySQL 인스턴스에 대해 제시된 이름이 동일한 경우 해당 인스턴스의 이름을 `local_mysql_server`와 같이 변경합니다. 그런 다음 각 클라이언트에 대해 인스턴스를 구성할 때 인스턴스 이름 `local_mysql_server`를 사용할 수 있습니다. 따라서 모든 클라이언트의 인스턴스 이름은 `local_mysql_server`입니다.
  - **MySQL 버전**(필수): 해당하는 옵션을 선택합니다. **MySQL Standard/Community**, **MySQL Enterprise Backup** 또는 **MariaDB 서버**. 사용하는 버전에 따라 이 대화 상자에서 수정할 수 있는 옵션이 결정됩니다.
  - **사용자 이름 및 암호**(필수): 사용자 이름과 암호를 입력합니다. 예를 들어 관리자 권한이 있는 계정과 같이 백업 및 복원 대상이 되는 MySQL 인스턴스 데이터베이스의 테이블을 읽고 쓸 수 있는 충분한 권한을 보유한 계정을 사용하십시오.
    - **중요:** NetVault Backup은 선택한 인스턴스 즉, 백업 및 복원 모두에 대해 MySQL 데이터베이스 액세스를 시도할 때마다 **사용자 이름** 및 **암호** 값을 참조합니다. MySQL 인스턴스에 대해 이러한 값 중 하나가 변경된 경우 해당 필드에서 **업데이트해야 합니다**. 그렇지 않으면 NetVault Backup이 인스턴스 및 작업 요청에 액세스할 수 없습니다.

- **MySQL 기본 디렉터리(필수):** MySQL 프로그램 파일이 상주하는 기본 디렉터리의 전체 경로를 입력합니다.
  - **Windows 기반 서버:** Windows에서 디렉터를 찾으려면 Windows 레지스트리에서 "위치" 값을 쿼리합니다.
  - **Linux 또는 UNIX 기반 서버:** MySQL의 기본 설치 시 다음과 같은 디렉터리가 있습니다.  
"/var/lib/mysql"
- **MySQL Bin 디렉터리:** MySQL Server에 MySQL 실행 파일이 들어 있는 디렉터리의 전체 경로를 입력합니다. 기본적으로 디렉터리는 다음과 같습니다.
  - **Linux 또는 UNIX 기반 서버:**  
"<MySQLbaseDirectory>/bin"
  - **Windows 기반 서버:**  
"<MySQLbaseDirectory>\bin"
- **참고:** **mysqlbackup** 유틸리티의 기본 경로는 사용자 환경 및 해당 위치를 사용자 지정했는지 여부에 따라 달라집니다. 예를 들어 Ubuntu 14 deb 패키지를 사용하는 경우 이 유틸리티는 기본적으로 /usr/bin/mysqlbackup에 설치됩니다.
- **mysqldump 경로:** 백업 및 복원 프로세스 중에 사용되는 **mysqldump** 유틸리티의 전체 경로와 파일 이름을 입력합니다. 대상 인스턴스에 대해 MySQL의 기본 설치가 수행된 경우 MySQL Server에서 사용 중인 OS에 따라 기본값이 표시될 수 있습니다.
  - **Linux 또는 UNIX 기반 서버:**  
"<MySQLbaseDirectory>/bin/mysqldump"
  - **Windows 기반 서버:**  
"<MySQLbaseDirectory>\bin\mysqldump.exe"
- **중요:** MySQL Bin 디렉터리나 mysqldump 경로가 앞에 나와 있는 기본값으로 설정되어 있는 경우 이 필드는 비워 둘 수 있습니다.
- **TCP 포트(Windows 기반 서버에만 해당):** MySQL의 각 인스턴스는 적절한 액세스를 위해 자체 포트 값을 필요로 합니다. MySQL의 기본 설치에는 포트 **3306**를 사용하며, 이 값은 기본적으로 이 필드에 표시됩니다. 선택한 인스턴스에 대해 다른 포트가 설정되어 있으면 올바른 값을 입력합니다.
  - **중요:** 단일 MySQL Server에 여러 인스턴스가 있을 경우 각 인스턴스에 자체 포트 값이 할당되며 이 값은 **포트 번호** 필드에 입력해야 합니다. 이 값은 각 인스턴스의 "my.ini" 파일에 있는 "port=" 값에 대해 표시되는 값과 동일합니다.
- **소켓 파일 경로(Linux 및 UNIX 기반 서버에만 해당):** MySQL 소켓 파일의 경로와 파일 이름을 입력합니다. MySQL 기본 설치가 수행된 경우 이 필드에 기본값이 표시됩니다. 기본적으로 소켓 파일은 다음 디렉터리에 있습니다.  
"/tmp/mysql.sock"
  - **중요:** MySQL Server에서 MySQL 표준 설치를 수행하고 기본 디렉터를 사용한 경우 **소켓 파일** 경로는 기본 설정으로 유지되어야 합니다. 그러나 설치 중에 다른 디렉터를 선택한 경우 이 필드에 올바른 위치를 입력합니다. 이 경로를 확인하려면 MySQL Server의 터미널 세션 프롬프트에서 다음 명령을 실행합니다.  
"**show variables like 'socket'**"  
올바른 소켓 파일 변수를 입력하지 않으면 플러그인이 백업 및 복원을 수행하지 않습니다.

- **기본 문자 집합:** 기본 문자 집합은 latin1입니다. 인코딩에 다른 문자 집합(예: UTF-8)을 사용하려는 경우 목록에서 선택하십시오.

## 5 MySQL Standard/Community를 사용하는 경우 다음 필드를 완성하십시오.

- **MyISAM 백업 방법:** 환경에서 MyISAM 저장소 엔진 또는 테이블 유형을 사용하는 경우 이 확인란을 선택한 다음 해당 하위 옵션을 선택합니다.

- **테이블 파일 잠금 및 복사(기본 선택):** 테이블 파일을 잠금, 플러시 및 복사하기 위한 표준 방법을 사용하려면 이 옵션을 선택합니다.

**i** **팁:** 플러그인은 InnoDB 테이블에 대해 MySQL "--single-transaction" 옵션을 사용하지만 MyISAM은 이 옵션을 지원하지 않습니다. 이 옵션은 InnoDB와 같은 트랜잭션 테이블에 사용할 때 유용하며 MyISAM과 같은 다른 유형의 테이블에는 유용하지 않습니다. 플러그인이 전체 백업을 완료하면 대상 MySQL 인스턴스의 모든 테이블 유형에 대해 일관된 상태가 필요합니다. 또한 백업 작업은 백업되는 각 테이블에 대해 **mysqldump**를 실행하므로 모든 테이블이 동시에 백업되지 않습니다. 이 백업 방법을 사용하면 전체 MySQL 인스턴스를 복원하는 대신, 필요 시 복원할 특정 데이터베이스 개체를 선택할 수 있습니다.

업데이트를 방지하기 위해 테이블을 잠그지 않는 온라인 접근 방식을 사용하려면 MySQL 복제 환경을 사용해 보십시오. 이 환경에서는 데이터베이스 사용자가 마스터 MySQL Server와 상호 작용하며, 이 경우 어떠한 테이블도 잠기지 않습니다. 슬레이브 MySQL Server에서 백업이 수행됩니다. 백업 프로세스가 진행되는 동안 백업이 완료될 때까지 마스터 서버에서 복제 업데이트가 일시 중지됩니다. 복제 설정과 함께 플러그인 사용에 대한 자세한 내용은 [기본 MySQL 복제 작업](#)을 참조하십시오.

- **Mysqldump:** 테이블이 과도하게 사용되거나 로드되는 경우 테이블을 복사하는 대신 **mysqldump** 유틸리티를 사용하려면 이 옵션을 선택합니다. 이 설정은 성능에 영향을 줄 수 있습니다.

- **MySQL 복제 활성화:** 이 인스턴스에 기본 MySQL 복제가 활성화되어 있으면 이 확인란을 선택합니다. 자세한 내용은 [기본 MySQL 복제 작업](#) 항목을 참조하십시오.

- **슬레이브 인스턴스:** 복제 가능 인스턴스의 경우, 이 인스턴스가 슬레이브로 구성되어 있으면 이 옵션을 선택합니다.
- **마스터 인스턴스:** 복제 가능 인스턴스의 경우, 이 인스턴스가 마스터로 구성되어 있으면 이 옵션을 선택합니다.

**i** **중요:** 복제를 구성하지 않으려면 이 옵션을 *선택하지 마십시오*. 그렇지 않으면 백업이 실패합니다.

- **특정 시점 복구 활성화:** PIT 백업 및 복원을 사용하려면 이 확인란을 선택합니다. 이 설정을 사용하면 데이터 손상이 발생하기 전후의 특정 시점으로 복구를 수행할 수 있습니다. 이 기능을 사용하려면 [MySQL Server에서 바이너리 로그 활성화\(Standard/Community 옵션 전용\)](#)에 설명된 대로 MySQL 바이너리 로그를 활성화하십시오.

- **바이너리 로그 인덱스 경로:** 특정 시점 복구 활성화 확인란을 선택한 경우 이 필드를 사용하여 바이너리 로그 인덱스 파일의 전체 경로를 지정합니다. 기본적으로 경로와 파일은 다음과 같습니다.

- **Linux 또는 UNIX 기반 서버:**

```
<MySQLbaseDirectory>/data/<instanceName>-bin.index
```

- **Windows 기반 서버:**

```
<MySQLbaseDirectory>\data\<instanceName>-bin.index
```

**i** **중요:** 이 플러그인은 구성 중에 바이너리 로그 인덱스 경로에 지정된 파일이 있는지 확인할 수 있습니다. 그러나 백업 작업이 제출될 때까지 지정된 파일이 실제로 바이너리 로그 인덱스인지 확인할 수는 없습니다. 지정된 파일이 잘못되었다고 확인하면 작업이 실패합니다.

- **릴레이 로그 인덱스 경로:** 슬레이브 인스턴스를 구성하는 경우, 릴레이 로그 인덱스 파일에 대한 전체 경로를 입력하여 백업에 포함시킵니다.

- **Linux 또는 UNIX 기반 서버:**

```
<MySQLbaseDirectory>/data/<instanceName>-relay-bin.index
```

- **Windows 기반 서버:**

```
<MySQLbaseDirectory>\data\<instanceName>-relay-bin.index
```

- 6 **MySQL Enterprise Backup**을 사용하는 경우 **mysqlbackup** 유틸리티가 있는 디렉터리의 전체 경로를 입력하여 필요한 **Mysqlbackup 경로** 필드를 완성합니다.

Linux 환경의 경우, Quest에서는 **mysqlbackup** 유틸리티를 실행하는 대신 NetVault Backup 스크립트, .sh 파일을 사용할 것을 권장합니다.

예를 들어 Linux를 사용하는 경우 .sh 파일의 기본 경로는 다음과 같습니다.

- MySQL Enterprise Backup 3.12의 경우: /usr/netvault/plugins/mysql/mysqlbackup-3.12.sh
- MySQL Enterprise Backup 4.0의 경우: /usr/netvault/plugins/mysql/mysqlbackup-4.0.sh

**i** | **참고:** .sh 파일 대신 유틸리티를 사용하도록 선택하는 경우 **mysqlbackup** 유틸리티의 기본 경로는 사용자 환경 및 해당 위치를 사용자 지정했는지 여부에 따라 달라집니다. 예를 들어 Ubuntu 14 deb 패키지를 사용하는 경우 이 유틸리티는 기본적으로 /usr/bin/mysqlbackup에 설치됩니다.

Windows 환경의 경우 **mysqlbackup** 유틸리티를 사용할 수 있습니다. 설정을 저장하려면 **확인**을 클릭하십시오.

## 기존 인스턴스 구성 업데이트

MySQL의 인스턴스가 성공적으로 구성되고 사용을 위해 플러그인에 추가된 후에는 다음 단계를 수행하여 구성 옵션을 편집할 수 있습니다.

- 1 탐색 창에서 **백업 작업 생성**을 클릭하고 **선택** 목록 옆에 있는 **새로 만들기**를 클릭합니다.
- 2 선택 트리에서 적용 가능한 클라이언트 노드를 엽니다.
- 3 **MySQL 용** 플러그인- 노드를 엽니다.
- 4 해당 인스턴스를 클릭하고 상황에 맞는 메뉴에서 **구성**을 선택합니다.

**구성** 대화 상자는 이전의 모든 설정과 함께 표시되어 필요한 수정을 할 수 있습니다.

**i** | **중요:** 편집 버전에서 **MySQL 인스턴스 이름** 필드가 비활성화됩니다. 이 필드는 정보 제공용이며 선택한 인스턴스의 이름을 표시합니다.

# 오류 조건에 대한 기본 작업 설정(선택 사항)

MySQL 백업 작업에는 일반적으로 여러 저장소 엔진, 데이터베이스 및 테이블이 포함됩니다. 백업 작업을 실행하는 동안 지원되지 않는 저장소 엔진이 발견되거나 데이터베이스 또는 테이블에 액세스할 수 없는 경우가 있습니다. 이런 상황이 발생하면 하나 이상의 항목이 제대로 백업되지 않지만 백업 작업에서 선택된 나머지 항목은 **백업됩니다**. 이러한 조건이 발생할 경우 MySQL DBA는 취해야 할 조치를 결정해야 합니다.

- 백업 작업을 경고와 함께 완료하거나 경고 없이 완료해야 합니까 아니면 실패해야 합니까?
- 완료된 항목의 백업을 보존해야 합니까 아니면 삭제해야 합니까?

플러그인을 사용하면 백업 및 복구 작업의 기본 옵션을 설정할 수 있습니다. 이러한 옵션은 작업별로 재정의할 수 있습니다.

- 1 탐색 창에서 **설정 변경**을 클릭합니다.
- 2 구성 페이지에서 필요에 따라 **서버 설정** 또는 **클라이언트 설정**을 클릭합니다.
- 3 **클라이언트 설정**을 선택한 경우에는 해당하는 클라이언트를 선택하고 **다음**을 클릭합니다.
- 4 **NetVault 서버 설정** 또는 **클라이언트 설정** 페이지에서 **플러그인 옵션**을 클릭합니다.

**MySQL 용** 플러그인- 섹션에 다음 항목이 나열됩니다.

- **잠긴 테이블:** 이 문제는 백업에 포함하도록 선택한 테이블이 플러그인 이외의 클라이언트 세션에서 잠긴 경우에 발생합니다.
  - **수동으로 선택한 테이블 사용 불가:** 이 문제는 백업 작업이 정의된 이후 삭제되는 등의 이유로 인해 개별 테이블을 백업할 수 없는 경우에 발생합니다.
  - **수동으로 선택한 데이터베이스 사용 불가:** 이 문제는 백업 작업이 정의된 이후 삭제되는 등의 이유로 인해 개별 데이터베이스를 백업할 수 없는 경우에 발생합니다.
  - **지원되지 않는 저장소 엔진:** 이 문제는 플러그인이 지원하지 않는 저장소 엔진 유형이 있는 백업 중에 테이블이 발견될 때 발생합니다.
- 5 이러한 각 조건에 대해 다음 설정 중 하나를 선택하십시오.
    - **경고와 함께 완료 - 저장 집합이 유지됨:** 작업에서 "백업이 경고와 함께 완료" 상태를 반환하고 성공적으로 백업된 항목을 포함하는 백업 저장 집합이 만들어집니다.
    - **경고 없이 완료 - 저장 집합이 유지됨:** 작업이 완료되고 "백업 완료" 상태를 반환합니다. 오류는 NetVault Backup 바이너리 로그에 기록되고 **작업 상태** 페이지에서는 무시됩니다. 백업된 항목을 포함하는 백업 저장 집합이 만들어집니다.
    - **실패 - 저장 집합이 유지됨:** 작업에서 "백업 실패" 상태를 반환합니다. 하지만 성공적으로 백업된 항목을 포함하는 백업 저장 집합이 생성됩니다.
    - **실패 - 저장 집합이 유지되지 않음:** 작업에서 "백업 실패" 상태를 반환하고 백업된 개체의 저장 집합이 유지되지 않습니다. 즉, 개체의 일부가 성공적으로 백업된 경우에도 저장 집합이 삭제됩니다.

**i | 중요:** 개별 백업 작업 수준에서 선택한 기본 작업을 재정의할 수 있습니다.

- 6 설정을 저장하려면 **적용**을 클릭합니다.

## 데이터 백업

- 데이터 백업: 개요
- 백업 수행

### 데이터 백업: 개요

백업을 완료하기 전에 다음 항목의 정보를 검토하십시오.

- [MySQL Standard/Community에 대한 중요 참고 사항](#)
- [MySQL Enterprise Backup에 대한 중요 참고 사항](#)
- 백업 전략 정의

**i** **중요:** Quest에서는 데이터베이스 이름에서 특수 문자를 생략할 것을 적극 권장합니다. 데이터베이스 이름에 다음 문자가 포함되어 있으면 플러그인이 복원할 수 없습니다. \$ ^ = @ # % +

백업 데이터는 MySQL Server에서 NetVault Backup Media Manager로 직접 스트리밍되므로, 위의 문자를 사용할 경우 백업 데이터의 무결성에 영향을 주는 시퀀스 명령으로 해석될 수 있습니다.

### MySQL Standard/Community에 대한 중요 참고 사항

MySQL Standard/Community 옵션을 사용하려면 다음 지침 및 정보를 검토하십시오.

- 영숫자 및 밑줄을 제외한 모든 문자는 특수 문자로 간주됩니다.
- MIXED 바이너리 로깅 형식 사용

#### 영숫자 및 밑줄을 제외한 모든 문자는 특수 문자로 간주됩니다.

사용자 환경에서 이름에 하이픈과 같은 특수 문자가 포함된 데이터베이스를 사용하는 경우 다음과 같은 제한 사항을 고려해야 합니다.

- 데이터베이스 이름에 하이픈이 포함되어 있더라도 MyISAM을 제외한 모든 테이블 유형이 백업됩니다. 이러한 백업은 해당 테이블 유형에 항상 mysqldump 명령이 사용되기 때문에 발생합니다.
- 데이터베이스 이름에 하이픈이 포함되어 있으면 **MyISAM 백업 방법**이 버전 4.2에 도입된 **Mysqldump** 옵션으로 설정되어 있는 경우 MyISAM 테이블이 백업됩니다. 그에 따라 백업 및 복원 성능에 부정적인 영향을 미칠 수 있습니다.
- **MyISAM 백업 방법**이 기본 **테이블 파일 잠금 및 복사** 옵션을 사용하도록 설정되어 있고 데이터베이스 이름에 하이픈이 포함되어 있는 경우 MyISAM 테이블이 백업되지 *않습니다*. 플러그인이 MySQL 명령을 무시하고 테이블 파일을 직접 복제하려고 시도하므로 백업이 생성되지 않습니다. 이 플러그인은 테이블 파일을 찾을 수 없다는 오류 메시지를 기록한 다음 저장 집합을 생성하지 않고 백업 작업이 실패합니다.

이전 버전에서는 플러그인이 데이터베이스 디렉터리의 존재 여부를 확인하려고 시도하고 실패 시 경고 메시지를 기록한 후 다음 데이터베이스의 백업을 계속했습니다. 백업이 경고와 함께 완료되고 다른 모든 데이터베이스가 포함된 저장 집합을 생성했습니다.

**MySQldump** 옵션을 사용할 때 최적화되지 않은 성능 효과 등의 이유로 원래 동작을 유지하면서 **테이블 파일 잠금 및 복사** 옵션을 사용할 수 있습니다. 이렇게 하려면 다음과 같이 플러그인 구성 파일 **"Nvmysql.cfg"**에서 **ValidateDatabaseDirectory** 매개 변수를 수동으로 **TRUE**로 설정합니다.

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

그런 다음 새 동작을 대신 사용하려는 경우 매개 변수를 **FALSE**로 변경하거나 **"Nvmysql.cfg"** 파일에서 매개 변수를 제거할 수 있습니다.

- 데이터베이스 또는 데이터베이스 개체가 있는 경우 테이블, 뷰 등과 같은 데이터베이스 또는 데이터베이스 개체를 생성할 수 있는 SQL 문이 포함된 증분 또는 차등 백업 복원에 실패합니다. 이 문제를 방지하려면 하나 이상의 데이터베이스 또는 데이터베이스 개체를 생성하거나 삭제한 후에 전체 백업을 실행합니다. 이 단계를 수행하면 이후의 증분 또는 차등 백업에 **CREATE** 또는 **DROP SQL** 문이 포함되지 않게 됩니다.

## MIXED 바이너리 로깅 형식 사용

MySQL에서는 **MIXED** 바이너리 로깅 형식을 사용할 때 **USE** 문의 사용을 강제하지 않습니다. 그러므로 Quest의 권장 사항에 따라 모든 데이터베이스 사용자와 프로그램에서 수정된 테이블이 **USE**가 선택한 데이터베이스에 있도록 합니다. 또한 데이터베이스 간 업데이트가 실행되지 않도록 하는 것이 좋습니다. 이 지침이 사용자 환경에 적합하지 않을 경우 Quest에서는 **MIXED** 바이너리 로깅 형식을 사용하지 않는 것을 권장합니다.

**i | 중요:** **MIXED** 바이너리 로깅 형식이 **used**인 경우 증분 및 차등 백업 작업이 경고와 함께 완료됩니다.

**MIXED** 바이너리 로깅 형식을 사용하는 환경에서는 **PIT** 복구 중에 바이너리 로그 항목이 재생되는 것을 방해할 수 있습니다. 복구 중에 플러그인은 **"--database"** 옵션과 함께 **mysqlbinlog**를 사용하여 복원 작업을 위해 선택한 데이터베이스와 관련된 항목만 재생할 수 있습니다. **"--Database"**를 사용하지 않으면 모든 항목이 재생되고 모든 데이터베이스에 영향을 미칩니다. **MIXED** 바이너리 로깅 형식을 사용하는 경우, **"--database"** 옵션을 사용하는 **mysqlbinlog**가 전체 또는 일부 항목을 재생하지 못하도록 하는 방식으로 항목이 작성됩니다. 자세한 내용은 [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) 페이지를 참조하십시오.

**MIXED** 바이너리 로깅 형식이 **"--database"** 옵션과 함께 올바르게 작동하려면 데이터베이스에 대한 특정 업데이트의 트랜잭션이 모두 데이터베이스를 선택하는 **USE** 문에서 실행되어야 합니다.

증분 또는 차등 백업이 복원되지 않고 **mysqlbinlog**가 MySQL Server에서 현재 바이너리 로그를 적용하는 경우에도 이와 동일한 상황이 발생합니다. 이 상황은 바이너리 로그가 백업에 저장되는 방식 때문이 아니라 바이너리 로그가 기록되는 방식 때문에 발생합니다.

**i | 중요:** 수정하는 테이블이 **USE** 문에 지정된 데이터베이스에 속하도록 하면 MySQL 명령 프롬프트를 통해 생성되는 트랜잭션에 적용됩니다. 또한 MySQL Server 데이터베이스와 상호 작용하는 스크립트, 프로그램 및 기타 응용 프로그램에 의해 생성되는 트랜잭션에도 적용됩니다.

다음 예제에서는 **MIXED**가 복원 동작에 영향을 미치는 다양한 방법을 보여 줍니다.

- **예 1:** 이 예에서는 데이터 행이 **my\_database**의 **my\_table**에 삽입됩니다. **USE** 문이 없으므로, **mysql** 데이터베이스와 같이 사용 중인 데이터베이스가 기본 데이터베이스입니다. **binlog\_format**이 **MIXED**로 설정되어 있으면 **mysqlbinlog**가 **"--database my\_database"** 옵션을 바이너리 로그에 적용할 경우 다음 트랜잭션이 재생되지 않습니다.

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

- **예 2:** 이 예에서는 데이터 행이 **my\_database**의 **my\_table**에 삽입됩니다. **USE** 문이 있지만 다른 데이터베이스를 지정합니다. 즉, **my\_database**가 **USE** 문에서 선택되지 않습니다. **binlog\_format**이 **MIXED**로 설정되어 있으면 **mysqlbinlog**가 **"--database my\_database"** 옵션을 바이너리 로그에 적용할 경우 다음 트랜잭션이 재생되지 않습니다.

```
-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)
```

- **예 3:** 이 예에서는 데이터 행이 **my\_database**의 **my\_table**에 삽입되고 **USE** 문에서 **my\_database**가 선택됩니다. **binlog\_format**이 **MIXED**로 설정되어 있으면 **mysqlbinlog**가 "**--database my\_database**" 옵션을 바이너리 로그에 적용할 경우 다음 트랜잭션이 재생됩니다.

```
-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)
```

- **예 4:** 이 예에는 두 개의 **insert** 쿼리가 있습니다. 첫 번째 **insert**는 **USE** 문에서 선택한 데이터베이스와 다른 **my\_database**에 대해 수행됩니다. 두 번째 **insert**는 **my\_database**를 선택하는 **USE** 문 범위에서 수행됩니다. **binlog\_format**이 **MIXED**로 설정되어 있으면 **my\_database**가 **USE** 문에 지정되어 있지 않으므로 첫 번째 **insert**가 재생되지 않지만, 두 번째 **insert**는 **my\_database**가 **USE** 문에 지정되어 있으므로 재생됩니다.

```
-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 Backup에 대한 중요 참고 사항

MySQL Enterprise Backup 옵션을 사용하려면 다음 지침 및 정보를 검토하십시오.

- MySQL에서는 백업 프로세스가 더 빠르고 안정성 및 확장성 기능이 중요하므로 중요 데이터에 InnoDB 테이블을 사용할 것을 권장합니다. MySQL Enterprise Backup을 사용하면 다양한 종류의 MySQL 테이블을 백업할 수 있으며 이는 InnoDB 테이블 백업에 최적화되어 있습니다. 이 옵션은 모든 InnoDB 테이블에 대해 핫 백업을 수행합니다. 데이터베이스가 실행되는 동안 핫 백업이 수행되기 때문에 백업으로 인해 진행 중인 데이터베이스 작업이 중단되지 않습니다. 또한 백업 프로세스 중에 변경된 모든 데이터베이스가 포함됩니다. 이러한 동작은 데이터베이스가 온라인 상태를 유지하면서 확장을 지원해야 하는 환경에서 중요하며, 이는 백업 완료에 필요한 시간에 영향을 미칩니다.
- 이 옵션을 사용하면 MyISAM 테이블과 기타 InnoDB 외 테이블이 워م 백업을 사용하여 마지막으로 백업됩니다. 워م 백업에서는 데이터베이스가 계속 실행되지만 백업이 완료되는 동안 테이블이 읽기 전용 액세스로 설정됩니다.
- 핫 백업 단계 중에 대량 데이터를 백업하려면 InnoDB를 새 테이블의 기본 저장소 엔진으로 만들고 기존 테이블을 InnoDB 저장소 엔진을 사용하도록 변환하는 것이 좋습니다. MySQL Server 5.5 이상에서는 InnoDB가 기본적으로 사용됩니다.
- 증분 백업은 주로 읽기 전용이거나 드물게 업데이트되는 InnoDB 테이블과 InnoDB 외 테이블을 대상으로 합니다. InnoDB 외 파일의 경우 마지막 백업 이후 변경된 경우 전체 파일에 포함됩니다.
- 플러그인을 사용할 때 다음 조건 중 하나가 충족될 경우 MySQL 인스턴스에 있는 모든 InnoDB 테이블이 백업됩니다.
  - 백업할 테이블만 명시적으로 선택되고 테이블이 저장소 엔진 InnoDB나 해당 유형이 아닙니다.

**예:** 두 개의 데이터베이스(DB1 및 DB2)가 포함된 MySQL 인스턴스가 있습니다. 각 데이터베이스에는 두 개의 테이블이 있습니다. DB1은 T1\_InnoDB와 T1\_MyISAM을 포함하고 DB2는 T2\_InnoDB와 T2\_MyISAM을 포함합니다. T1\_MyISAM과 T2\_MyISAM을 백업하는 경우 T1\_InnoDB와 T2\_InnoDB 또한 백업됩니다. InnoDB 테이블 중 하나를 포함하면 해당 InnoDB 테이블만 백업됩니다. 데이터베이스 중 하나를 선택하면 이 데이터베이스의 테이블만 백업됩니다.

- 일부 또는 모든 데이터베이스가 백업용으로 선택되고 모든 관련 InnoDB 테이블이 백업에서 제외됩니다.

**예:** 두 개의 데이터베이스(DB1 및 DB2)가 포함된 MySQL 인스턴스가 있습니다. 각 데이터베이스에는 두 개의 테이블이 있습니다. DB1은 T1\_InnoDB와 T1\_MyISAM을 포함하고 DB2는 T2\_InnoDB와 T2\_MyISAM을 포함합니다. DB1과 DB2를 백업하고 T1\_InnoDB와 T2\_InnoDB를 제외하는 경우 T1\_InnoDB와 T2\_InnoDB 또한 백업됩니다. 두 개의 InnoDB 테이블 중 하나만 제외하면 다른 InnoDB 테이블만 백업됩니다.

이 설명은 **mysqlbackup** 유틸리티인 MySQL Enterprise Backup의 현재 동작을 반영하며, 이 동작은 향후 릴리스인 MySQL 3.12 이후에 변경될 수 있습니다.

- MySQL 5.6 이상에서는 **innodb\_file\_per\_table** 구성 옵션이 기본적으로 **활성화**됩니다. **innodb\_file\_per\_table** 옵션이 비활성화된 상태로 생성된 InnoDB 테이블은 InnoDB 시스템 테이블스페이스에 저장되며 백업에서 생략될 수 없습니다. 테이블스페이스 외부에 InnoDB 테이블을 배치해야 하는 경우 **innodb\_file\_per\_table** 옵션이 MySQL에서 활성화되어 있는 동안 생성합니다. 각각의 .ibd 파일에는 테이블이 1개인 데이터 및 인덱스만 포함되어 있습니다.

## 백업 전략 정의

MySQL 백업 전략을 정의할 때 다음 질문에 답하십시오.

- **MySQL Standard/Community** 또는 **MySQL Enterprise Backup** 옵션을 사용하기를 원하십니까? 두 가지 버전 모두 사용자 환경에 구축되어 있는 경우에도 플러그인과 함께 하나의 전략만 사용할 수 있습니다. MEB 기반 방법 또는 mysqldump 기반 방법을 사용하되 이 두 가지 조합을 함께 사용할 수는 없습니다.

MEB 기반 옵션을 사용하면 백업을 위해 선택한 모든 데이터베이스 개체에 대해 **mysqlbackup** 유틸리티 또는 해당 NetVault Backup 스크립트가 한 번 실행되고 **mysqlbackup** 출력 로그가 작업 로그에 포함됩니다. 데이터 백업은 두 단계로 수행됩니다. 첫 번째 단계에서는 모든 InnoDB 테이블이 복사됩니다. 두 번째 단계에서는 다른 모든 유형의 테이블이 복사됩니다. InnoDB 테이블의 핫 백업 지원 외에도 MEB 기반 옵션은 향상된 백업 성능을 지원합니다.

mysqldump 기반 옵션을 사용할 경우 각 테이블, 트리거 및 저장된 절차에 대해 명령을 한 번만 실행합니다. 핫 백업은 지원되지 않습니다.

- 전체 백업 중에 전체 인스턴스에 대한 읽기 전용 액세스가 필요한 경우 전체 백업을 얼마나 자주 수행해야 합니까?
- 빠른 백업과 빠른 복원 중 무엇이 더 중요합니까?
- 허용 가능한 최대 데이터 손실량은 어느 정도입니까?

이러한 질문에 답변하면 이행해야 하는 백업의 유형 및 빈도를 정의하는 데 도움이 됩니다.

- [MySQL Standard/Community 백업 유형 검토](#)
- [MySQL Enterprise Backup의 백업 유형 검토](#)
- [MySQL Standard/Community 백업 시퀀스의 예](#)

## MySQL Standard/Community 백업 유형 검토

MySQL Standard/Community 옵션을 사용하는 경우 플러그인은 **mysqldump**를 사용하여 다음 유형의 백업을 제공합니다.

- 전체 백업
- 증분 백업
- 차등 백업

- 개별 데이터베이스/테이블 복사 전용 백업
- 전체 데이터베이스 복사본 백업

각 MySQL 인스턴스에 대한 데이터 보호 요구 사항에 부합하는 적절한 백업 시퀀스를 선택하기 위한 첫 번째 단계는 바로 이러한 백업이 어떻게 다른지 이해하는 것입니다.

## 전체 백업

**MySQL Standard/Community** 옵션의 전체 백업에서 플러그인은 **mysqldump** 유틸리티를 사용하여 *인스턴스에 포함된 모든 데이터베이스를 백업*합니다. 전체 백업은 거의 모든 복원 시나리오에 대한 시작점을 제공하므로 모든 백업 전략의 기본이 됩니다. 플러그인을 사용하여 생성된 전체 백업은 전체 인스턴스, 개별 또는 여러 데이터베이스, 개별 또는 여러 테이블을 복원하는 데 사용할 수 있습니다.

**전체 또는 증분 백업 이후에 바이너리 로그 삭제** 옵션을 사용하면 전체 또는 증분 백업 이후에 바이너리 로그가 삭제됩니다. 이 옵션은 표준 MySQL Server 구성에서 플러그인을 사용할 때, **MySQL 복제 활성화**가 비활성화되어 있을 때, **특정 시점 복구 활성화**가 활성화되어 있을 때 기본적으로 사용됩니다. 플러그인이 클러스터에 연결되면 비활성화됩니다. 플러그인 외부에서 바이너리 로그 삭제를 관리해야 합니다.

**i | 중요:** NetVault Backup Server가 클러스터링된 MySQL Server와 표준 MySQL Server를 모두 관리하는 혼합 환경에서는 MySQL 기반 클러스터에 대해 표준 MySQL Server용으로 생성된 백업 옵션 집합을 재사용하지 마십시오.

**바이너리 로그 삭제...** 옵션을 선택하지 않으면 플러그인이 해당 구성 파일에서 **마지막 백업 로그**를 추적하므로 사용자는 재량에 따라 수동으로 바이너리 로그를 삭제할 수 있습니다. 예를 들어 바이너리 로그를 슬레이브 인스턴스에 복제할 때까지 마스터 인스턴스에서 삭제하지 않으려는 MySQL 복제 환경을 사용하는 경우 바이너리 로그를 수동으로 제거해야 합니다.

## 증분 백업:

증분 백업에서는 마지막 전체 백업 또는 증분 백업 이후 생성된 트랜잭션 로그를 백업한 다음 바이너리 로그를 삭제합니다. 바이너리 로그는 인스턴스 기반이므로 모든 데이터베이스에 대한 트랜잭션 로그가 백업되고 하나의 단위로 삭제됩니다.

증분 백업은 미디어 오류 또는 데이터 손상이 발생한 후에 데이터 손실을 최소화하는 데 필수적입니다. 증분 백업을 사용하여 잘못된 업데이트 또는 삭제된 테이블과 같은 데이터 손상 전후의 시점으로 복원할 수 있습니다. 전체 백업과 달리 증분 백업은 백업 중에 읽기 전용 액세스가 필요하지 않습니다.

MySQL 증분 백업의 경우 바이너리 로그를 활성화하는 **"-log-bin"** 옵션을 사용하여 MySQL 인스턴스를 시작해야 합니다. 이 절차는 **MySQL Server에서 바이너리 로그 활성화(Standard/Community 옵션 전용)**에 설명되어 있습니다. 자세한 내용은 *MySQL 참조 안내서*의 바이너리 로그 섹션을 참조하십시오.

앞에서 설명한 대로, **전체 또는 증분 백업 이후에 바이너리 로그 삭제** 옵션을 사용하면 전체 또는 증분 백업 이후에 바이너리 로그가 삭제됩니다. 이 옵션을 사용하지 않으면 플러그인이 해당 구성 파일에서 **마지막 백업 로그**를 추적하므로 사용자는 재량에 따라 수동으로 바이너리 로그를 삭제할 수 있습니다.

## 차등 백업

차등 백업은 마지막 전체 백업 또는 증분 백업 이후 생성된 트랜잭션 로그를 백업합니다. 그러나 백업 완료 시 바이너리 로그는 **삭제되지 않습니다**. 따라서 이후의 차등 백업은 크기와 시간이 늘어납니다. 이 유형의 각 백업에는 이전 차등 백업에 포함된 바이너리 로그와 이전 차등 백업 이후 생성된 바이너리 로그가 포함되므로 크기와 시간이 늘어납니다. 예를 들어 월요일부터 토요일까지 예약된 차등 백업과 함께 전체 백업을 일요일에 수행하는 경우 월요일의 차등 백업에는 일요일의 전체 백업 이후에 생성된 바이너리 로그가 포함됩니다. 또한 화요일의 차등 백업에는 월요일에 생성된 바이너리 로그와 화요일에 생성된 바이너리 로그가 포함됩니다. 수요일의 차등 백업에는 월요일, 화요일, 수요일의 바이너리 로그가 포함되는 식으로 작업이 진행됩니다.

증분 백업과 마찬가지로 차등 백업을 사용하여 미디어 오류 또는 데이터 손상 이후 데이터 손실을 줄이고 오류 또는 손상 전후의 시점으로 복원할 수도 있습니다. 전체 백업과 달리 차등 백업은 백업 중에 읽기 전용 액세스가 **필요하지 않습니다**.

차등 백업의 경우 바이너리 로그를 활성화하는 **"-log-bin"** 옵션을 사용하여 MySQL 인스턴스를 시작해야 합니다. 이 절차는 [MySQL Server에서 바이너리 로그 활성화\(Standard/Community 옵션 전용\)](#)에 설명되어 있습니다. 자세한 내용은 [MySQL 참조 안내서](#)의 바이너리 로그 섹션을 참조하십시오.

## 증분 백업 및 차등 백업

증분 백업은 백업 후에 바이너리 로그를 삭제하므로 마지막 증분 백업 이후 생성된 바이너리 로그만 백업되어 후속 증분 백업이 더 빨라집니다. 하지만 증분 백업을 사용하는 복원 시퀀스에서는 전체 백업과 실패한 시점 사이에 수행한 모든 증분 백업을 연속으로 복원해야 합니다. 이 프로세스를 사용하면 다수의 복원 작업을 시작하는 데 필요한 DBA 개입이 증가하므로 복원 시간이 더 오래 걸릴 수 있습니다.

차등 백업은 백업된 후에 바이너리 로그를 삭제하지 않으므로 마지막 전체 백업 이후의 모든 바이너리 로그가 백업에 포함되어 후속적인 각 차등 백업이 더 오래 걸립니다. 그렇지만 차등 백업을 사용하는 복원 시퀀스에서는 전체 백업을 복원한 후에 단 하나의 차등 백업만 복원하면 됩니다. 이 프로세스는 복원 프로세스 중에 필요한 DBA 개입이 줄어들기 때문에 복원 속도가 빨라집니다.

## 개별 데이터베이스/테이블 복사 전용 백업

특수한 용도로 백업을 수행해야 할 때도 있는데 이 경우 전체 데이터베이스에 대한 전반적인 백업 및 복원 절차에 영향을 주지 않아야 합니다. 예를 들어 백업이 테스트 환경의 소스일 수도 있고 복제 슬레이브 인스턴스에 대한 초기 동기화로 사용될 수도 있습니다. 개별 데이터베이스/테이블 복사 전용 백업은 이러한 특수 용도로 설계되었기 때문에 MySQL 환경을 "복사"할 수 있습니다. **"복사 전용"** 백업은 설정된 백업 시퀀스와 무관하며 전체, 증분 또는 차등 백업의 복구 성능에 영향을 주지 않습니다. 하지만 전체 백업의 대체 백업으로 **사용할 수는 없습니다**.

## 전체 데이터베이스 복사본 백업

개별 데이터베이스/테이블 복사 전용 백업에 대해 설명된 것처럼, 전체 데이터베이스 복사본 백업 옵션은 선택한 MySQL 데이터베이스의 복사본(선택된 데이터베이스의 모든 해당 InnoDB 테이블 포함)을 생성하기 때문에 특수 용도로만 사용됩니다. **"복사본"** 백업은 설정된 백업 시퀀스와 무관하며 전체, 증분 또는 차등 백업의 복구 성능에 영향을 주지 않습니다. 하지만 전체 백업의 대체 백업으로 **사용할 수는 없습니다**.

**i | 중요:** 선택한 데이터베이스에 대한 모든 테이블이 InnoDB 테이블인 경우에만 이 옵션을 사용할 수 있습니다.

## 개별 데이터베이스/테이블 복사 전용 백업과 전체 데이터베이스 복사본 백업

선택한 각 데이터베이스에 대해 데이터베이스의 테이블이 하나만 선택되어 있더라도 전체 데이터베이스 복사본 백업 옵션은 전체 데이터베이스를 백업합니다. 이 옵션을 사용하면 백업할 개별 데이터베이스를 선택할 수 있지만 개별 테이블은 선택할 수 없습니다. 또한 이 옵션은 InnoDB 테이블 백업만 지원합니다.

개별 데이터베이스/테이블 복사 전용 백업 옵션을 사용하면 개별 데이터베이스 및 개별 테이블을 선택할 수 있으며 백업에 InnoDB 및 MyISAM 테이블을 포함할 수 있습니다. 그러나 일반적으로 개별 데이터베이스/테이블 복사 전용 백업 옵션보다 전체 데이터베이스 복사본 백업 옵션을 사용할 때 백업이 더 빨리 완료됩니다.

## MySQL Enterprise Backup의 백업 유형 검토

**MySQL Enterprise Backup** 옵션의 경우, 플러그인이 선택된 모든 데이터베이스 개체에 대해 **mysqlbackup** 명령을 한 번 실행하여 전체, 증분 및 TTS 유형의 백업을 수행할 수 있습니다.

## 전체 백업

**MySQL Enterprise Backup** 옵션의 전체 백업에서 플러그인은 **mysqlbackup** 유틸리티 또는 해당 NetVault Backup 스크립트를 사용하여 **인스턴스에 포함된 모든 선택된 데이터베이스 개체**를 백업합니다. 전체 백업은 거의 모든 복원 시나리오에 대한 시작점을 제공하므로 모든 백업 전략의 기본이 됩니다. 플러그인을 사용하여 생성된 전체 백업은 전체 인스턴스, 개별 또는 여러 데이터베이스, 개별 또는 여러 테이블을 복원하는 데 사용할 수 있습니다.

## 증분 백업

InnoDB 테이블의 경우 마지막 전체 백업 또는 증분 백업 이후 변경된 데이터만 백업됩니다. InnoDB 외 테이블의 경우 마지막 전체 백업 또는 증분 백업 이후 테이블에서 변경된 사항이 있으면 전체 테이블이 백업됩니다.

## TTS(Transportable Tablespace) 백업

TTS 백업을 수행하는 경우 플러그인이 전체 백업을 수행하고 "--use-tts" MySQL 옵션을 추가합니다.

- 중요:** Quest에서는 백업 계획과 별도로 독립 실행형 백업으로 TTS 백업만 생성할 것을 권장합니다. TTS 백업은 부분 백업이므로 전체 또는 증분 백업 전략을 대체하거나 보완할 수 없으며 재해 복구 작업에서 사용할 수 없습니다.

TTS 백업을 생성하려는 경우 다음과 같은 제한 사항을 고려해야 합니다.

- TTS 백업은 MySQL Server 5.6 이상에서만 지원됩니다.
- InnoDB 테이블만 백업에 포함됩니다.
- `innodb_file_per_table` 옵션을 활성화한 상태에서 생성된 테이블만 백업에 포함됩니다.
- 파티션이 공유 테이블스페이스에 생성된 경우 분할된 테이블의 백업에 실패합니다.
- 백업 시 바이너리 또는 릴레이 로그가 제외됩니다.

"--use-tts" 옵션 사용 시 제한 사항에 대한 자세한 내용은 <https://dev.mysql.com/doc/mysql-enterprise-backup/4.0/en/backup-partial-options.html> 페이지를 참조하십시오.

## MySQL Standard/Community 백업 시퀀스의 예

다음은 다양한 시퀀스의 예입니다.

- 전체 백업만 수행:** 요구 사항에 따라 전 날까지 데이터 보호를 보장해야 하는 경우 매일 읽기 전용 액세스가 허용된다면(예: 정규 업무 시간 이후) 매일 전체 백업만 수행해도 충분합니다. DBA는 마지막 전체 백업 시점까지만 데이터베이스를 복구할 수 있도록 보장하지만 MySQL Server에 현재 존재하는 바이너리 로그를 사용하여 PIT 복구를 수행할 수 있습니다.
- 전체 및 증분 백업:** 요구 사항에 따라 전 날까지 데이터 보호를 보장해야 하지만 대상 MySQL 인스턴스에 대한 읽기 전용 액세스만 간헐적으로 허용되는 경우(예: 정규 업무 시간 이후, 주 단위 또는 격주 단위로만) **최대한 빠른 속도로 백업을 수행해야 하며** 증분 백업과 결합된 전체 백업이 최적의 조합입니다. 예를 들어 매 주 일요일 밤 11:00에 전체 백업이 수행되지만, 증분 백업은 월요일~토요일 밤 11:00에 수행됩니다. 일요일 밤 전체 백업이든 아니면 증분 백업 중 하나이든 상관없이 각 증분 백업에는 전날 밤의 백업 이후에 생성된 바이너리 로그가 포함됩니다.

이 유형의 백업 시퀀스를 복원하는 데에는 더 많은 시간이 걸립니다. 예를 들어 화요일에 복구를 수행하는 경우 일요일의 전체 백업과 월요일의 증분 백업만 복원해야 합니다. 반면, 목요일에 복구를 수행하는 경우 일요일의 전체 백업 이후에 월요일, 화요일, 수요일의 증분 백업을 복원해야 합니다. 백업이 더 빠르긴 하지만 여러 개의 복원 작업을 실행하는 데 필요한 추가적인 개입 때문에 복원 시간이 더 오래 걸릴 수 있습니다.

- 전체 및 차등 백업:** 요구 사항에 따라 전 날까지 데이터 보호를 보장해야 하지만 대상 MySQL 인스턴스에 대한 읽기 전용 액세스만 간헐적으로 허용되는 경우(예: 정규 업무 시간 이후, 주 단위 또는 격주 단위로만) **최대한 빠른 속도로 복원을 수행해야 하며** 차등 백업과 결합된 전체 백업이 최적의 조합입니다. 예를 들어 매 주 일요일 밤 11:00에 전체 백업이 수행되지만, 차등 백업은 월요일~토요일 밤 11:00에 수행됩니다. 각 차등 백업에는 마지막 전체 백업 이후에 생성된 바이너리 로그가 포함됩니다. 앞에서 설명한 것처럼 이 프로세스에서는 전체 백업 시간이 더 오래 걸립니다.

복구가 필요한 시간과 관계없이 동일한 수의 복원 작업이 필요합니다. 예를 들어 화요일에 복구를 수행하는 경우 일요일의 전체 백업과 월요일의 차등 백업만 복원해야 합니다. 반면, 목요일에 복원을 수행하는 경우 일요일의 전체 백업 이후에 수요일의 차등 백업을 복원해야 합니다. 이후의 차등 백업에서는 크기는 물론 기간도 증가하지만 실행해야 하는 복원 작업의 수가 줄어들기 때문에 복원 속도가 빨라집니다.

# 백업 수행

MySQL용 플러그인-을 사용하는 백업에는 다음과 같은 항목에 설명된 단계가 포함됩니다.

- 백업 데이터 선택
- 백업 옵션 설정
- 작업 마무리 및 제출

## 백업 데이터 선택

백업 작업을 생성하려면 백업 선택 집합, 백업 옵션 집합, 일정 집합, 대상 집합 및 고급 옵션 집합을 사용해야 합니다.

백업 선택 집합은 증분 및 차등 백업에 필수적입니다. 전체 백업 중에 백업 선택 집합을 만들어서 전체, 증분 및 차등 백업에 사용합니다. 증분 또는 차등 백업에 선택 집합을 사용하지 않으면 백업 작업에서 오류가 보고됩니다. 자세한 내용은 *Quest NetVault Backup 관리자 안내서*를 참조하십시오.

**i | 팁:** 기존 집합을 사용하려면 **백업 작업 생성**을 클릭하고 **선택** 목록에서 집합을 선택합니다.

1 탐색 창에서 **백업 작업 생성**을 클릭합니다.

구성 안내 링크에서 마법사를 시작할 수도 있습니다. 탐색 창에서 **구성 안내**를 클릭합니다. **NetVault 구성 마법사** 페이지에서 **백업 작업 생성**을 클릭합니다.

2 **작업 이름**에 작업의 이름을 지정합니다.

진행률을 모니터링하거나 데이터를 복원할 때 작업을 쉽게 식별할 수 있도록 설명 이름을 지정합니다. 작업 이름에는 영숫자 및 영숫자가 아닌 문자가 포함될 수 있지만 라틴 문자가 아닌 문자는 포함될 수 없습니다. Linux에서 이름의 길이는 최대 200자입니다. Windows에서는 길이 제한이 없습니다. 그러나 모든 플랫폼에서 최대 40자를 권장합니다.

3 **선택** 목록 옆에 있는 **새로 만들기**를 클릭합니다.

4 플러그인 목록에서 **MySQL용 플러그인-**을 열어 MySQL Server를 표시합니다.

5 해당 데이터를 선택합니다.

- 백업 작업에서 선택한 인스턴스에 모든 MySQL 데이터베이스를 포함하려면 **모든 데이터베이스** 노드를 선택합니다.
- 자세한 선택 항목을 확인하려면 **모든 데이터베이스** 노드를 열어 개별 데이터베이스를 표시합니다. 또한 각 개별 데이터베이스를 열어 해당하는 개별 테이블을 표시할 수 있으며, 필요에 따라 이러한 개별 테이블을 백업 작업에 포함하도록 선택할 수 있습니다.
- 백업에서 항목을 명시적으로 생략하려면 상위 수준 항목을 선택하고 해당 하위 수준 항목을 클릭하여 녹색 확인 표시를 빨간색의 "X"(십자 모양)로 바꾸십시오. 그러면 해당 항목이 생략된 것으로 표시됩니다.

**i** **중요:** MySQL Standard/Community 옵션을 사용하여 백업에 대한 자세한 데이터 집합을 선택한 경우 백업 옵션 탭에서 개별 데이터베이스/테이블 복사 전용을 백업 유형으로 선택합니다. 다른 형태의 백업, 즉 전체, 증분 또는 차등 백업이 선택된 경우 상세한 선택 항목은 무시되고 전체 데이터베이스가 백업됩니다. MySQL 5.5 이상에서는 저장된 절차, 기능 및 트리거가 MySQL Standard/Community 옵션의 전체 및 개별 데이터베이스/테이블 복사 전용 백업으로 자동 백업됩니다.

MySQL 5.5 이상의 경우 "information\_schema" 데이터베이스가 선택 트리에 표시되지만 선택할 수는 없습니다. 이 문제는 이 데이터베이스에 포함된 모든 데이터가 동적으로 생성되고 영구적인 의미로 존재하지 않기 때문에 발생합니다. 따라서 플러그인이 모든 백업에서 information\_schema 데이터베이스를 자동으로 제외합니다.

6 저장을 클릭하고 새 집합 만들기 대화 상자에 이름을 입력한 다음 저장을 클릭합니다.

이름에는 영숫자 및 영숫자가 아닌 문자가 포함될 수 있지만 라틴 문자가 아닌 문자는 포함될 수 없습니다. Linux에서 이름의 길이는 최대 200자입니다. Windows에서는 길이 제한이 없습니다. 그러나 모든 플랫폼에서 최대 40자를 권장합니다.

## 백업 옵션 설정

다음 단계는 백업 옵션 집합을 생성하거나 기존 옵션을 선택하는 것입니다. 백업 옵션 탭에서 사용 가능한 설정은 MySQL Standard/Community 또는 MySQL Enterprise Backup 옵션의 사용 여부에 따라 달라집니다.

## MySQL Standard/Community에 대한 백업 옵션 설정

백업할 항목을 선택한 상태에서 수행하려는 백업 유형을 선택하고 실패하는 경우 다른 동작을 선택할 수 있습니다.

**i** **팁:** 기존 집합을 사용하려면 플러그인 옵션 목록에서 사용할 집합을 선택합니다.

- 1 플러그인 옵션 목록 옆의 새로 만들기를 클릭합니다.
- 2 해당하는 옵션을 선택합니다.

**i** **중요:** MySQL의 대상 인스턴스를 "복제 마스터 인스턴스"로 지정한 경우, 즉, 이 인스턴스에 대해 구성 대화 상자에서 MySQL 복제 활성화 및 마스터 인스턴스를 선택한 경우, 백업의 전체, 증분 및 차등 형태를 선택할 수 없습니다. 자세한 내용은 기본 MySQL 복제 작업 항목을 참조하십시오.

- **모든 데이터베이스에 대한 전체 백업**(기본 선택): 현재 MySQL 인스턴스에 포함된 모든 데이터베이스 및 모든 테이블의 전체 백업을 수행하려면 이 옵션을 선택합니다. 이 옵션을 선택하면 백업 선택 집합 대해 선택한 데이터에 관계없이 모든 항목이 백업됩니다.
- **증분 백업:** 마지막 전체 백업 또는 증분 백업 이후 생성된 트랜잭션 로그만 백업하려면 이 옵션을 선택합니다.
- **차등 백업:** 마지막 전체 백업 또는 증분 백업 이후 생성된 모든 트랜잭션 로그를 백업하려면 이 옵션을 선택합니다. 후속 차등 백업을 수행할 때마다 원래 전체 백업이 수행된 이후에 생성된 모든 바이너리 로그가 포함됩니다. 백업이 완료되면 바이너리 로그가 MySQL 인스턴스에 **보관됩니다**.

**i** **중요:** MIXED 바이너리 로깅 형식을 사용하는 경우 증분 및 차등 백업 작업이 경고와 함께 완료됩니다. 자세한 내용은 MIXED 바이너리 로깅 형식 사용 항목을 참조하십시오.

**!** **주의:** MIXED 바이너리 로깅 형식을 사용하는 경우 PIT 복구 중에 바이너리 로그 항목이 재생되지 않을 수 있으므로 복구를 위해 선택한 데이터베이스가 선택한 지점으로 롤백되지 않을 수 있습니다. 자세한 내용은 MySQL Standard/Community에 대한 중요 참고 사항 및 MIXED 바이너리 로깅 형식 사용을 참조하십시오.

- **개별 데이터베이스/테이블 복사 전용:** 데이터베이스의 전체 백업 및 복원 절차(예: 테스트 환경 생성)에 영향을 미치지 않는 특수 목적의 MySQL 환경 복사본을 생성하려면 이 옵션을 선택하십시오. 이 옵션을 사용할 경우 복사본 백업은 전체, 증분 또는 차등 MySQL 백업 시나리오에 의해 설정된 시퀀스에 영향을 미치지 않습니다. 즉, 이러한 백업은 바이너리 로그에 영향을 주지 않습니다. 이 백업 형태는 전체, 증분 또는 차등 MySQL 백업 시나리오를 사용하여 일반적인 백업 시퀀스로 설정해야 하는 것과는 무관합니다. 또한 복사본 백업을 전체 백업 **대신 사용할 수는 없습니다.**
- **전체 데이터베이스 복사본 백업:** 특수한 용도로 MySQL 환경의 복사본을 생성하여 선택한 데이터베이스의 모든 내용을 백업하려면 이 옵션을 선택하십시오. 예를 들어 테스트 환경을 구축하려면 이 옵션을 선택합니다. 이 옵션은 개별 데이터베이스/테이블 복사 전용 옵션과 유사하지만, 이 옵션은 선택한 모든 데이터베이스와 해당 테이블을 백업합니다. 선택한 데이터베이스에 대한 모든 테이블이 InnoDB 테이블인 경우에만 이 옵션을 사용할 수 있습니다. 이 옵션을 사용할 경우 복사본 백업은 전체, 증분 또는 차등 MySQL 백업 시나리오에 의해 설정된 시퀀스에 영향을 미치지 않습니다. 즉, 이러한 백업은 바이너리 로그에 영향을 주지 않습니다. 또한 복사본 백업을 전체 백업 **대신 사용할 수는 없습니다.**

**i | 중요:** 선택한 데이터베이스에 대한 모든 테이블이 InnoDB 테이블인 경우에만 이 옵션을 사용할 수 있습니다. 이 옵션을 선택하면 선택한 모든 데이터베이스 및 모든 해당 테이블이 백업됩니다. 백업하려는 특정 테이블을 선택하는 경우에도 이런 결과가 발생합니다. 특정 테이블을 백업하려면 개별 데이터베이스/테이블 복사 전용 옵션을 사용합니다.

- **읽기 액세스 권한이 있는 모든 테이블 잠금...: 전체 백업**을 선택했으며 현재 읽기 전용 액세스 권한이 있는 인스턴스의 모든 데이터베이스를 잠금 처리하여 트랜잭션의 손실을 방지하려면 이 확인란을 선택합니다. 이 옵션을 선택하면 전체 백업이 진행 중인 동안 사용자가 전체 인스턴스에서 데이터를 삽입, 업데이트 또는 삭제할 수 없습니다. 이 옵션을 선택 취소하면 플러그인은 테이블이 **백업될 때** 백업 프로세스 중에 각 테이블을 잠급니다. 따라서 인스턴스에 관련된 테이블이 포함되어 있는 경우 Quest에서는 이 옵션을 선택하여 프로세스 중에 모든 테이블이 잠겨 있는지 확인할 것을 권장합니다.
- **바이너리 로그 삭제...** 이 옵션은 표준 MySQL Server 구성에서 플러그인을 사용할 때, **MySQL 복제 활성화**가 비활성화되어 있을 때, **특정 시점 복구 활성화**가 활성화되어 있을 때 기본적으로 선택되어 있습니다. 플러그인이 클러스터에 연결되면 비활성화됩니다. 플러그인 외부에서 바이너리 로그 삭제를 관리해야 합니다. Quest에서는 이 옵션을 사용하도록 권장하지만 바이너리 로그에 대해 원하는 제어 수준을 결정할 수 있습니다.

**i | 중요:** NetVault Backup Server가 클러스터링된 MySQL Server와 표준 MySQL Server를 모두 관리하는 혼합 환경에서는 MySQL 기반 클러스터에 대해 표준 MySQL Server용으로 생성된 백업 옵션 집합을 재사용하지 **마십시오.**

3 각 조건에 대해 해당하는 작업을 선택합니다. 자세한 내용은 **오류 조건에 대한 기본 작업 설정(선택 사항)**을 참조하십시오.

이러한 각각의 조건에서 기본 작업을 다르게 설정하더라도 이 작업에 대해 수행할 작업을 선택할 수 있습니다.

- **경고와 함께 완료 - 저장 집합이 유지됨:** 작업에서 **"백업이 경고와 함께 완료"** 상태를 반환하고 성공적으로 백업된 항목을 포함하는 백업 저장 집합이 만들어집니다.
- **경고 없이 완료 - 저장 집합이 유지됨:** 작업이 완료되고 **"백업 완료"** 상태를 반환합니다. 오류는 NetVault Backup 바이너리 로그에 기록되고 **작업 상태** 페이지에서는 무시됩니다. 백업된 항목을 포함하는 백업 저장 집합이 만들어집니다.
- **실패 - 저장 집합이 유지됨:** 작업에서 **"백업 실패"** 상태를 반환합니다. 하지만 성공적으로 백업된 항목을 포함하는 백업 저장 집합이 생성됩니다.
- **실패 - 저장 집합이 유지되지 않음:** 작업에서 **"백업 실패"** 상태를 반환하고 백업된 개체의 저장 집합이 유지되지 않습니다. 즉, 개체의 일부가 성공적으로 백업된 경우에도 저장 집합이 삭제됩니다.

#### 4 **mysqldump 옵션** 상자에 **mysqldump** 유틸리티가 작업에 사용할 명령 옵션을 나열합니다.

옵션은 대시 또는 이중 대시로 시작해야 하며, 잘못된 문자(; | < >)를 포함할 수 없습니다.

이러한 옵션들은 먼저 **mysqldump** 명령에 추가된 이후 플러그인이 내부에서 생성하는 옵션에 추가됩니다. 이 순서로 인해 여기에 입력한 옵션 중 내부에서 생성된 옵션과 모순되는 옵션은 플러그인에서 생성된 옵션으로 재정의됩니다.

작업 실패의 원인이 되며 **mysqldump**에서 감지된 오류는 작업 로그의 오류 로그 메시지에 포함됩니다.

이 작업을 수행하기 위해 이전에 MySQL 옵션 파일을 설정한 경우 여기에 입력하는 옵션은 옵션 파일에 지정된 옵션에 추가됩니다. 플러그인이 기존 MySQL 옵션 파일을 무시하도록 하려면 이 상자에 첫 번째 옵션으로 **--no-defaults**를 입력합니다.

**mysqldump** 버전에서 지원하는 옵션에 대한 자세한 내용은 해당 MySQL 설명서를 참조하십시오.

**주의:** 이 기능과 함께 **--routines (-R)** 또는 **--triggers** 옵션을 **사용하지 마십시오.** 이러한 옵션을 사용하면 데이터베이스 테이블 백업에 방해가 되고 백업이 성공하더라도 복원에 실패할 수 있습니다. 데이터베이스에 대해 백업해야 하는 저장된 절차 및 트리거가 있는 경우 플러그인은 **--routines** 및 **--triggers** 옵션을 사용하여 내부에서 **mysqldump** 명령을 생성합니다.

#### 5 **저장**을 클릭하여 집합을 저장합니다.

#### 6 **새 집합 만들기** 대화 상자에서 집합에 대한 이름을 입력하고 **저장**을 클릭합니다.

이름에는 영숫자 및 영숫자가 아닌 문자가 포함될 수 있지만 라틴 문자가 아닌 문자는 포함될 수 없습니다. Linux에서 이름의 길이는 최대 200자입니다. Windows에서는 길이 제한이 없습니다. 그러나 모든 플랫폼에서 최대 40자를 권장합니다.

## MySQL Enterprise Backup에 대한 백업 옵션 설정

백업할 항목을 선택한 상태에서 수행하려는 백업 유형을 선택하고 실패하는 경우 다른 동작을 선택할 수 있습니다.

**팁:** 기존 집합을 사용하려면 **플러그인 옵션** 목록에서 사용할 집합을 선택합니다.

#### 1 **플러그인 옵션** 목록 옆의 **새로 만들기**를 클릭합니다.

#### 2 해당하는 옵션을 선택합니다.

- **전체 백업**(기본 선택): 현재 MySQL 인스턴스에 선택된 모든 데이터베이스 및 테이블을 백업하려면 이 옵션을 선택합니다.
- **증분 백업**: 마지막 전체 백업 또는 증분 백업 이후 변경된 InnoDB 테이블, 전체 테이블 또는 InnoDB 외 테이블에 대해서만 데이터를 백업하려면 이 옵션을 선택하십시오.
- **TTS(Transportable Tablespace) 백업**: MySQL TTS 기능을 활용하는 부분 백업을 생성하려면 이 옵션을 선택하십시오.

#### 3 **저장**을 클릭하여 집합을 저장합니다.

#### 4 **새 집합 만들기** 대화 상자에서 집합에 대한 이름을 입력하고 **저장**을 클릭합니다.

이름에는 영숫자 및 영숫자가 아닌 문자가 포함될 수 있지만 라틴 문자가 아닌 문자는 포함될 수 없습니다. Linux에서 이름의 길이는 최대 200자입니다. Windows에서는 길이 제한이 없습니다. 그러나 모든 플랫폼에서 최대 40자를 권장합니다.

## 작업 마무리 및 제출

- 1 일정, 대상 저장소 및 고급 옵션 목록을 사용하여 필요한 추가 옵션을 구성합니다.
- 2 해당되는 경우 저장 또는 저장 및 제출을 클릭합니다.

**i | 팁:** 이미 생성하여 저장한 작업을 실행하려면 탐색 창에서 **작업 정의 관리**를 선택하고 해당 작업을 선택한 다음 **지금 실행**을 클릭합니다.

**작업 상태** 페이지에서 진행률을 모니터링하고 **로그 보기** 페이지에서 로그를 볼 수 있습니다. 자세한 내용은 *Quest NetVault Backup 관리자 안내서*를 참조하십시오.

# 데이터 복원

- 데이터 복원: 개요
- MySQL에서 데이터 복원
- MySQL Standard/Community에 대한 고급 복원 절차 사용

## 데이터 복원: 개요

이 항목에서는 플러그인 복원 프로세스에 대해 간략히 소개하고 사용할 수 있는 모든 기능을 설명합니다. 또한 [MySQL Standard/Community 복원 시나리오 예시](#) 및 [MySQL Enterprise Backup 복원 시나리오 예시](#)에서는 다양한 복원 유형의 예를 제공합니다. Quest에서는 이러한 항목을 검토하여 사용 가능한 기능 및 다양한 복원 유형에 적용되는 방법을 숙지할 것을 권장합니다.

- [MySQL Standard/Community에 사용 가능한 복원 방법 검토](#)
- [MySQL Enterprise Backup에 사용 가능한 복원 옵션 검토](#)

## MySQL Standard/Community에 사용 가능한 복원 방법 검토

성공적인 복원을 수행하려면 사용할 수 있는 복원 유형을 이해해야 합니다.

### 전체 또는 개별 데이터베이스/테이블 복사 전용 복원

플러그인이 전체 또는 개별 데이터베이스/테이블 복사 전용 복원을 수행하는 경우 MySQL의 **mysqldump** 유틸리티를 사용하여 테이블을 생성하고 채우는 데 사용되는 SQL 문을 백업 미디어에 직접 스트리밍합니다. 플러그인이 이러한 백업 형태 중 하나를 복원하면 백업 미디어에서 SQL 문을 직접 읽고 자동으로 실행합니다.

### 증분 또는 차등 복원

플러그인이 증분 또는 차등 백업을 수행하는 경우 MySQL 바이너리 로그 인덱스는 백업 미디어에 복사해야 하는 바이너리 로그를 결정하는 데 사용됩니다. 이러한 백업이 복원되면 바이너리 로그가 임시 디렉터리 **""NETVAULT\_HOME/tmp/MySQL"**로 복원됩니다. **mysqlbinlog** 유틸리티는 바이너리 로그에 기록된 각 트랜잭션에 대한 SQL 문을 생성하는 데 사용됩니다. 이러한 문은 자동으로 실행됩니다. 이 프로세스를 "바이너리 로그 적용"이라고 합니다.

증분 및 차등 복원 중에는 바이너리 로그에 기록된 모든 트랜잭션이 적용될 수 있으며 특정 시점(PIT 복구)에 적용될 수 있습니다. PIT 복구는 실수로 테이블을 삭제하거나 잘못된 업데이트를 실행하는 경우와 같이 데이터 손상 직전 시점까지 복구하려고 할 때 유용합니다.

## 시간 기반 특정 시점(PIT) 복구

중분 또는 차등 복원 중에 복원할 바이너리 로그에서 PIT 복구를 수행할 수 있습니다. 시간 기준 PIT 복구는 데이터 손상이 발생한 시간을 알고 있을 때 유용합니다. 예를 들어 개발자가 오전 6시에 테이블을 삭제한 경우, PIT 복구는 오전 5시 55분의 중지 시간으로 수행될 수 있습니다.

시간 기반 PIT 복구는 일반적으로 1단계 프로세스입니다. **옵션** 탭에서 **바이너리 로그 복원 및 적용** 옵션을 선택하고 중지 시간을 원하지 않는 트랜잭션 **직전**으로 지정하여 중분 또는 차등 백업으로부터 바이너리 로그를 복원합니다.

## 위치 기반 특정 시점(PIT) 복구

데이터 손상의 실제 시간을 알 수 없거나 보다 정확한 복구가 필요한 경우 위치 기반 PIT 복구를 사용해야 합니다. 예를 들어 개발자가 데이터베이스에서 테이블을 삭제했지만 테이블을 삭제한 정확한 시간을 알 수 없는 경우 위치 기반 PIT 복구를 사용해야 합니다.

위치 기반 PIT 복구는 3단계 프로세스입니다.

- 1 **옵션** 탭에서 **시간 또는 위치를 식별하기 위해 바이너리 로그를 임시 디렉터리로 복원** 옵션을 선택하여 중분 또는 차등 백업의 바이너리 로그를 MySQL Server의 임시 디렉터리로 복원합니다.
- 2 MySQL의 **mysqlbinlog** 유틸리티를 사용하여 원치 않는 트랜잭션의 특정 위치를 식별하십시오. 자세한 내용은 *MySQL 참조 안내서*의 특정 시점(PIT) 복구 섹션을 참조하십시오.
- 3 동일한 중분 또는 차등 백업을 다시 복원하십시오. 그러나 **임시 디렉터리에서 바이너리 로그 적용** 복원 옵션을 선택하고 원치 않는 트랜잭션 바로 앞에 위치하는 중지 위치를 지정하십시오.

# MySQL Enterprise Backup에 사용 가능한 복원 옵션 검토

MEB 기반 방법을 사용하여 전체 또는 중분 복원을 수행할 수 있습니다. TTS 백업 옵션을 사용한 경우 TTS 복원을 완료할 수도 있습니다. TTS 복원 프로세스 내에서 특정 테이블을 복원(부분 복원이라고 함)하고 지정된 테이블 중 하나의 이름을 바꿀 수 있는 추가 옵션을 사용할 수 있습니다.

**i | 중요:** TTS 백업 또는 복원 기능을 제한적으로 사용할 수 있으므로 Quest에서는 TTS 옵션을 신중하게 사용할 것을 권장합니다.

TTS 백업을 복원하기 위한 다음과 같은 제한 사항에 유의하십시오.

- MySQL Server와의 연결이 설정되어 있어야 하므로 대상 MySQL Server가 실행 중인지 확인합니다.
- 복원하려는 테이블이 대상 서버에 존재하지 않는지 확인합니다.
- 대상 서버에서 백업이 수행된 원래 MySQL Server에 사용된 것과 같은 페이지 크기를 사용하는지 확인합니다.
- 대상 서버에서 **innodb\_file\_per\_table** 옵션이 활성화되어 있는지 확인합니다.
- 복원할 InnoDB 파일(.ibd 파일)이 대상 서버의 **innodb\_file\_format** 변수 값과 일치하지 않으면 복원이 실패합니다.

자세한 내용은 <https://dev.mysql.com/doc/mysql-enterprise-backup/4.0/en/restore-use-tts.html> 페이지를 참조하십시오.

# MySQL에서 데이터 복원

MySQL용 플러그인-을 사용한 표준 복원에는 다음과 같은 항목에 설명된 단계가 포함됩니다.

- 복원 데이터 선택
- 복원 옵션 설정
- 작업 마무리 및 제출
- MySQL Standard/Community 복원 시나리오 예시
- MySQL Enterprise Backup 복원 시나리오 예시

## 복원 데이터 선택

- 1 탐색 창에서 **복원 작업 생성**을 클릭합니다.
- 2 **복원 작업 생성 - 저장 집합 선택** 페이지의 **플러그인 유형** 목록에서 **MySQL용** 플러그인-을 선택합니다.
- 3 저장 집합 테이블에 표시된 항목을 추가로 필터링하려면 **클라이언트**, **날짜** 및 **작업 ID** 목록을 사용하십시오. 테이블에는 저장 집합 이름(작업 이름 및 저장 집합 ID), 만든 날짜 및 시간, 크기가 표시됩니다. 기본적으로 목록은 만든 날짜별로 정렬됩니다.
- 4 저장 집합 테이블에서 해당 항목을 선택합니다.  
저장 집합을 선택하면 다음 세부 정보가 **저장 집합 정보** 영역에 표시됩니다. 작업 ID, 작업 제목, 서버 이름, 클라이언트 이름, 플러그인 이름, 저장 집합 날짜 및 시간, 만료 설정, 증분 백업 여부, 아카이브 여부, 저장 집합 크기, 스냅샷 기반 백업 여부.
- 5 **다음**을 클릭합니다.
- 6 **선택 집합 만들기** 페이지에서 복원할 데이터를 선택합니다.

복원에 포함할 첫 번째 선택 가능한 노드는 복구할 백업 유형에 따라 다릅니다:

- **전체 또는 개별 데이터베이스/테이블 복사 전용 백업:** 루트 노드는 실제 데이터베이스 및 테이블 데이터가 백업에 포함되었기 때문에 **"모든 데이터베이스"**로 나열됩니다.

**i** **중요:** 루트 노드에 **"모든 데이터베이스"**라는 이름이 지정되어 있지만 대상 MySQL 인스턴스에 대해 현재 존재하는 모든 데이터베이스를 고려하지는 않습니다. 이 옵션을 선택하면 백업 작업을 위해 선택된 모든 데이터 항목만 복원됩니다. 즉, 복원을 위해 이 노드를 선택하면 현재 MySQL 인스턴스에 있는 모든 데이터베이스를 복원하지 않고 백업에 포함된 데이터베이스만 복원합니다.

- **증분 또는 차등 백업:** 루트 노드는 **"바이너리 로그"**로 나열됩니다. 이전 백업이 수행된 이후에 발생한 트랜잭션(바이너리 로그)이 이러한 백업 형태로 포함되어 있기 때문입니다.

- 7 좀 더 세부적인 복원이 필요한 경우 루트 노드를 두 번 클릭하여 열고 백업에 포함된 개별 데이터베이스를 표시합니다.

또한 개별 데이터베이스를 열어 선택할 테이블을 열 수 있습니다.

**i** **중요:** MySQL은 여러 파일 형식을 사용하여 데이터베이스 정보를 저장합니다. **.frm** 파일을 복원 프로세스에 포함시켜 복원된 데이터베이스가 제대로 작동하는지 확인하십시오.

# 복원 옵션 설정

옵션 탭에 표시되는 옵션은 **MySQL Standard/Community** 옵션 또는 **MySQL Enterprise Backup** 옵션의 사용 여부에 따라 달라집니다.

- [MySQL Standard/Community에 대한 복원 옵션 설정](#)
- [MySQL Enterprise Backup에 대한 복원 옵션 설정](#)

## MySQL Standard/Community에 대한 복원 옵션 설정

선택 집합 만들기 페이지에서 **플러그인 옵션 편집**을 클릭하고 **특정 시점(PIT) 복구** 및 **복원 대상** 탭에서 해당 매개변수를 구성합니다. 표시되는 옵션은 복원할 백업 유형에 따라 달라집니다.

- [전체 또는 개별 데이터베이스 복원 옵션](#)
- [증분 또는 차등 데이터베이스 복원 옵션](#)

### 전체 또는 개별 데이터베이스 복원 옵션

전체 백업이나 개별 데이터베이스/테이블 복사 전용 백업만 복원하려면 다음 단계를 수행하십시오.

1 다음 지침에 따라 **특정 시점 복구** 탭에서 해당하는 옵션을 선택합니다.

- **현재 바이너리 로그에서 PIT 복구 수행:** MySQL Server의 MySQL 바이너리 로그 디렉터리에 현재 있는 바이너리 로그를 사용하여 선택한 데이터 개체의 **특정 시점** 복원을 수행하려면 이 옵션을 선택합니다. 이 옵션을 활성화하면 이 탭의 나머지 모든 옵션을 사용할 수 있습니다.
- **특정 시점 유형:** 이 섹션에서 해당 PIT 복구 형태를 선택합니다:
  - **시간 기반 PIT(기본 선택):** 선택한 데이터를 **시간 기반 특정 시점(PIT) 복구**에 설명된 **특정 시점**으로 복원하려면 이 옵션을 선택합니다. 이 옵션을 선택하면 **시간 기반 PIT 세부 정보** 섹션을 사용할 수 있습니다.
  - **위치 기반 PIT:** 선택한 데이터를 **위치 기반 특정 시점(PIT) 복구**에 설명된 **원치 않는 트랜잭션 바로 앞에 위치하는 특정 중지 위치**로 복원하려면 이 옵션을 선택합니다. 이 옵션을 선택하면 **위치 기반 PIT 세부 정보** 섹션을 사용할 수 있습니다.
- **시간 기반 PIT 세부 정보:** 시간 기반 PIT를 선택한 경우 해당 옵션을 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 원치 않는 트랜잭션 **이전**에 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택합니다. 이 옵션만 선택하면 여기서 지정한 시간 **이후**에 발생한 모든 트랜잭션이 손실됩니다. 24시간 형식을 사용하여 관련된 **중지 날짜/시간** 필드에 해당 날짜 및 시간을 지정합니다.
  - **오류/잘못된 SQL 문 이후로 복구 활성화:** 원치 않는 트랜잭션 **이후**에 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택합니다. 이 옵션만 선택하면 여기서 지정한 시간 **이전**에 발생한 모든 트랜잭션이 손실됩니다. 24시간 형식을 사용하여 관련된 **시작 날짜/시간** 필드에 해당 날짜 및 시간을 지정합니다. 특정 시작 날짜와 시간을 선택한 경우 트랜잭션에 대한 중지 날짜 및 시간도 설정할 수 있습니다.
    - **없음(기본 선택):** 지정된 날짜 및 시간 이후에 발생한 모든 트랜잭션을 복구하려면 이 옵션을 선택한 상태로 둡니다.
    - **특정 날짜:** 특정 기간 동안 발생한 트랜잭션만 포함하려면 이 옵션을 선택하고 24시간 형식을 사용하여 관련 필드에 해당 중지 시간을 입력하십시오.

**i** **중요:** 복원된 바이너리 로그 및 현재 바이너리 로그 모두에서 PIT 복구가 설정되어 있으면 복원된 바이너리 로그 또는 현재 바이너리 로그에 중지 시간이 있는지 확인할 필요가 없습니다. MySQL은 지정된 시간에 자동으로 중지 및 시작되며 지정된 중지 시간 이후의 모든 바이너리 로그를 무시합니다.

특히 원치 않는 트랜잭션이 발생한 특정 시간 범위가 있는 경우 이 두 옵션을 모두 사용할 수 있습니다. 예를 들어 2007년 1월 29일 오전 11시에서 오전 11시 15분 사이에 수집된 데이터를 원하지 않는 경우 ... **이전으로 복구 활성화** 옵션을 선택하고 "11:00" - "29 Jan 2007"을 **중지 날짜/시간**으로 입력합니다. 또한 ... **이후로 복구 활성화** 옵션을 활성화하고 "11:15" - "29 Jan 2007"을 **시작 날짜/시간**으로 입력합니다. 그 결과 2007년 1월 29일 오전 11시에 서 11시 15분 사이에 발생한 모든 트랜잭션이 복원에서 제외됩니다.

- **위치 기반 PIT 세부 정보:** 위치 기반 PIT를 선택한 경우 해당 옵션을 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 원치 않는 트랜잭션 **이전**에 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택합니다. 이 옵션만 선택하면 여기에 지정된 위치 **이후**에 발생한 모든 트랜잭션이 손실됩니다. 이 옵션은 다음과 같은 관련 옵션을 제공합니다.
    - **중지 위치:** 바이너리 로그에서 원치 않는 트랜잭션 **이전**의 위치를 입력하십시오. 예를 들어 원치 않는 트랜잭션의 위치가 805인 경우 804를 입력합니다.
    - **중지 위치가 포함된 바이너리 로그:** 이 목록을 사용하여 **중지 위치**에 지정된 중지 위치가 포함된 특정 바이너리 로그를 선택합니다. 다른 파일 또는 해당 파일이 목록에 없으면 **기타**를 선택하고 텍스트 상자에 해당 파일 이름을 입력합니다.
  - **오류/잘못된 SQL 문 이후로 복구 활성화:** 원치 않는 트랜잭션 **이후**에 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택합니다. 이 옵션만 선택하면 여기에 지정된 위치 **이전**에 발생한 모든 트랜잭션이 손실됩니다. 또한 이 옵션은 다음과 같은 관련 옵션을 제공합니다.
    - **시작 위치:** 바이너리 로그에서 원치 않는 트랜잭션 **이후**의 위치를 입력하십시오. 예를 들어 원치 않는 트랜잭션의 위치가 805인 경우 806을 입력합니다.
    - **시작 위치가 포함된 바이너리 로그:** 이 목록을 사용하여 **시작 위치**에 지정된 시작 위치가 포함된 특정 바이너리 로그를 선택합니다. 다른 파일 또는 해당 파일이 목록에 없으면 **기타**를 선택하고 텍스트 상자에 해당 파일 이름을 입력합니다.
    - **중지 위치: 없음** (기본 선택) - 지정된 **시작 위치** 다음에 발생한 모든 트랜잭션을 복원하려는 경우 이 옵션을 선택한 상태로 둡니다.
    - **중지 위치: 특정 위치:** 바이너리 로그 위치의 특정 범위 내에 발생한 트랜잭션만 포함하려면 이 옵션을 선택합니다. 해당 중지 위치를 입력하고 **중지 위치가 포함된 바이너리 로그** 목록에서 해당 바이너리 로그를 선택합니다. 다른 파일이 사용된 경우 **기타**를 선택한 뒤 파일 이름을 입력합니다. **시작 위치** 및 **특정 위치** 필드에 지정된 위치 사이에 발생한 트랜잭션만 복원에 포함됩니다.

**i** **중요:** 특히 원치 않는 트랜잭션이 발생한 특정 위치 범위가 있는 경우 이 두 옵션을 모두 사용할 수 있습니다. 예를 들어 위치 805와 810 사이에 수집된 데이터에 원치 않는 트랜잭션이 포함되어 있는 경우 ... **이전으로 복구 활성화** 옵션을 선택하고 "805"를 **중지 위치**로 입력한 다음 관련 옵션을 구성하여 바이너리 로그를 호출할 수 있습니다. 또한, ... **이후로 복구 활성화** 옵션을 선택하고 "810"을 **시작 위치**로 입력한 다음 관련 옵션을 구성하여 바이너리 로그를 호출할 수 있습니다. 그 결과, 805와 810 사이에 지정된 바이너리 로그에 기록된 모든 트랜잭션이 복원에서 생략됩니다. 또한 중지 및 시작 위치는 원치 않는 트랜잭션의 위치보다 큰 임의의 숫자가 아니라 바이너리 로그에 나열된 **실제 위치**여야 합니다.

2 다음 지침에 따라 **복원 대상** 탭에서 해당 옵션을 선택합니다.

- **동일한 MySQL 인스턴스로 복원:** 복원이 원래 백업했던 것과 동일한 인스턴스를 대상으로 하는 경우 이 필드를 비워 둡니다. NetVault Backup은 구성 대화 상자에 설정된 값을 사용합니다. 자세한 내용은 **플러그인 구성** 항목을 참조하십시오.
- **다른 MySQL 인스턴스로 복원:** 선택한 데이터의 복원을 다른 인스턴스로 재배포하려는 경우 새 인스턴스에 대한 액세스를 허용하는 **사용자 이름** 및 **암호** 필드에 해당 정보를 입력합니다. 또한 **인스턴스 이름** 필드에서 새 인스턴스에 대해 설정된 NetVault Backup 이름을 입력합니다. 이 이름은 구성 대화 상자에서 **MySQL 인스턴스 이름**으로 설정된 이름입니다. 자세한 내용은 **플러그인 구성**을 참조하십시오.

**i | 중요:** 다른 MySQL 인스턴스에 대한 재배포 복원을 시도하기 전에 **대체 MySQL Server로 복구**를 검토해야 합니다.

## 증분 또는 차등 데이터베이스 복원 옵션

증분 또는 차등 백업을 복원하려면 다음 단계를 수행하십시오.

1 다음 지침에 따라 **특정 시점 복구** 탭에서 해당하는 옵션을 선택합니다.

- **PIT 복구 수행:** 선택한 데이터 항목의 **특정 시점** 복원을 수행하려면 이 옵션을 선택합니다. 이 옵션을 활성화하면 이 탭의 나머지 모든 옵션을 사용할 수 있습니다.

증분 및 차등 복원에서는 바이너리 로그를 사용하여 복원을 완료합니다. 따라서 이 형태의 백업을 복원할 때는 선택한 데이터베이스와 관련된 바이너리 로그를 복구하는 방법을 결정합니다. 다음 방법 중 하나를 선택합니다.

- **바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용):** 손상이 발생한 시간이나 위치를 알고 있는 경우 이 옵션을 선택하여 백업 장치에서 바이너리 로그를 복원하고, 기록된 트랜잭션을 하나의 복원 작업에 적용합니다. MySQL 바이너리 로그 디렉터리에 현재 상주하는 바이너리 로그에서 PIT 복구를 수행하려면 **현재 바이너리 로그 포함** 확인란을 선택합니다. 이 프로세스는 증분 또는 차등 백업에 저장된 모든 바이너리 로그 트랜잭션이 복원되고 적용된 **이후에** 수행됩니다.
- **시간 또는 위치를 식별하기 위해 로그를 임시 디렉터리로 복원:** 선택한 증분 또는 차등 백업과 관련된 바이너리 로그만 MySQL Server의 **임시 디렉터리**, 즉 **"NETVAULT\_HOME/tmp/MySQL/"**로 복원하려면 이 옵션을 선택합니다. 이 옵션을 사용하면 **mysqlbinlog** 유틸리티를 사용하여 복구된 로그를 검토하고 데이터 손상의 시간과 위치를 확인할 수 있습니다.
- **임시 디렉터리의 바이너리 로그 적용:** 이전에 **시간 또는 위치를 식별하기 위해 로그를 임시 디렉터리로 복원** 옵션을 사용하고 **mysqlbinlog** 유틸리티를 사용하여 복원에서 생략할 손상된 데이터를 식별하려면 이 옵션을 선택합니다. 이 프로세스는 임시 디렉터리로 복원된 바이너리 로그를 적용합니다. MySQL 바이너리 로그 디렉터리에 현재 상주하는 바이너리 로그에서 PIT 복구를 수행하려면 **현재 바이너리 로그 포함** 확인란을 선택합니다. 이 프로세스는 임시 디렉터리에 존재하는 바이너리 로그 트랜잭션이 복원되고 적용된 **후에** 수행됩니다.
- **특정 시점 유형: PIT 복구 수행** 옵션을 활성화하고 해당 PIT 복구를 선택합니다.
  - **시간 기반 PIT(기본 선택):** 선택한 데이터를 **시간 기반 특정 시점(PIT) 복구**에 설명된 **특정 시점으로** 복원하려면 이 옵션을 선택합니다. 이 옵션을 선택하면 **시간 기반 PIT 세부 정보** 섹션을 사용할 수 있습니다.
  - **위치 기반 PIT:** 선택한 데이터를 **위치 기반 특정 시점(PIT) 복구**에 설명된 **위치 않는 트랜잭션 바로 앞에 위치하는 특정 중지 위치**로 복원하려면 이 옵션을 선택합니다. 이 옵션을 선택하면 **위치 기반 PIT 세부 정보** 섹션을 사용할 수 있습니다.

- **시간 기반 PIT 세부 정보:** 시간 기반 PIT를 선택한 경우 해당 옵션을 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 원치 않는 트랜잭션 *이전에* 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택합니다. 이 옵션만 선택하면 여기서 지정한 시간 *이후*에 발생한 모든 트랜잭션이 손실됩니다. 24시간 형식을 사용하여 관련된 **중지 날짜/시간** 필드에 해당 날짜 및 시간을 지정합니다.
  - **오류/잘못된 SQL 문 이후로 복구 활성화:** 원치 않는 트랜잭션 *이후*에 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택합니다. 이 옵션만 선택하면 여기서 지정한 시간 *이전에* 발생한 모든 트랜잭션이 손실됩니다. 24시간 형식을 사용하여 관련된 **시작 날짜/시간** 필드에 해당 날짜 및 시간을 지정합니다. 특정 시작 날짜와 시간을 선택한 경우 트랜잭션에 대한 중지 날짜 및 시간도 설정할 수 있습니다.
    - **없음**(기본 선택): 지정된 날짜 및 시간 이후에 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택한 상태로 둡니다.
    - **특정 날짜:** 특정 시간 범위 내에 발생한 트랜잭션만 포함하려면 이 옵션을 선택합니다. 24시간 형식을 사용하여 관련 필드에 해당 중지 시간을 입력합니다.
- i** **중요:** 특히 원치 않는 트랜잭션이 발생한 특정 시간 범위가 있는 경우 이 두 옵션을 모두 사용할 수 있습니다. 예를 들어 2007년 1월 29일 오전 11시에서 오전 11시 15분 사이에 수집된 데이터를 원하지 않는 경우 ... **이전으로 복구 활성화** 옵션을 선택하고 "**11:00**" - "**29 Jan 2007**"을 **중지 날짜/시간**으로 입력합니다. 또한 ... **이후로 복구 활성화** 옵션을 선택하고 "**11:15**" - "**29 Jan 2007**"을 **시작 날짜/시간**으로 입력합니다. 그 결과 2007년 1월 29일 오전 11시에서 11시 15분 사이에 발생한 모든 트랜잭션이 복원에서 제외됩니다.
- **위치 기반 PIT 세부 정보:** 위치 기반 PIT를 선택한 경우 해당 옵션을 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 원치 않는 트랜잭션 *이전에* 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택합니다. 이 옵션만 선택하면 여기에 지정한 위치 *이후*에 발생한 모든 트랜잭션이 손실됩니다. 이 옵션은 다음과 같은 관련 옵션을 제공합니다.
    - **중지 위치:** 바이너리 로그에서 원치 않는 트랜잭션 *이전의* 위치를 입력하십시오. 예를 들어 원치 않는 트랜잭션의 위치가 805인 경우 804를 입력합니다.
    - **중지 위치가 포함된 바이너리 로그:** 이 목록을 사용하여 **중지 위치**에 지정된 중지 위치가 포함된 특정 바이너리 로그를 선택합니다. 다른 파일 또는 해당 파일이 목록에 없으면 **기타**를 선택하고 텍스트 상자에 해당 파일 이름을 입력합니다.
  - **오류/잘못된 SQL 문 이후로 복구 활성화:** 원치 않는 트랜잭션 *이후*에 발생한 모든 트랜잭션을 복원하려면 이 옵션을 선택합니다. 이 옵션만 선택하면 여기에 지정한 위치 *이전에* 발생한 모든 트랜잭션이 손실됩니다. 또한 이 옵션은 다음과 같은 관련 옵션을 제공합니다.
    - **시작 위치:** 바이너리 로그에서 원치 않는 트랜잭션 *이후*의 위치를 입력하십시오. 예를 들어 원치 않는 트랜잭션의 위치가 805인 경우 806을 입력합니다.
    - **시작 위치가 포함된 바이너리 로그:** 이 목록을 사용하여 **시작 위치**에 지정된 시작 위치가 포함된 특정 바이너리 로그를 선택합니다. 다른 파일 또는 해당 파일이 목록에 없으면 **기타**를 선택하고 텍스트 상자에 해당 파일 이름을 입력합니다.
    - **중지 위치:** **없음**(기본 선택): 지정된 **시작 위치** 이후에 발생한 **모든** 트랜잭션을 복원하려는 경우 이 옵션을 선택한 상태로 둡니다.
    - **중지 위치: 특정 위치:** 바이너리 로그 위치의 특정 범위 내에 발생한 트랜잭션만 포함하려면 이 옵션을 선택합니다. 해당 중지 위치를 입력하고 **중지 위치가 포함된 바이너리 로그** 목록에서 해당 바이너리 로그를 선택합니다. 다른 파일이 사용된 경우 **기타**를 선택한 뒤 파일 이름을 입력합니다. **시작 위치** 및 **특정 위치** 필드에 지정된 위치 사이에 발생한 트랜잭션만 복원에 포함됩니다.

**i | 중요:** 특히 원치 않는 트랜잭션이 발생한 특정 위치 범위가 있는 경우 이 두 옵션을 모두 사용할 수 있습니다. 예를 들어 위치 805와 810 사이에 수집된 데이터에 원치 않는 트랜잭션이 포함되어 있는 경우 ... **이전으로 복구 활성화** 옵션을 선택하고 **"805"를 중지 위치**로 입력한 다음 관련 옵션을 구성하여 바이너리 로그를 호출할 수 있습니다. 또한, ... **이후로 복구 활성화** 옵션을 선택하고 **"810"을 시작 위치**로 입력한 다음 관련 옵션을 구성하여 바이너리 로그를 호출할 수 있습니다. 그 결과, 805와 810 사이에 지정된 바이너리 로그에 기록된 모든 트랜잭션이 복원에서 생략됩니다. 또한 중지 및 시작 위치는 원치 않는 트랜잭션의 위치보다 큰 임의의 숫자가 아니라 바이너리 로그에 나열된 **실제 위치**여야 합니다.

2 다음 지침에 따라 **복원 대상** 탭에서 해당 옵션을 선택합니다.

이 탭에는 **복원 대상 세부 정보** 섹션이 포함되어 있습니다. 이 섹션의 필드를 사용하여 MySQL의 대상 인스턴스에 대한 복원 액세스를 허용하려면 계정 정보를 입력하십시오. 원하는 복원 유형에 따라 다음과 같은 옵션을 사용하십시오.

- **동일한 MySQL 인스턴스로 복원:** 복원이 원래 백업했던 것과 동일한 인스턴스를 대상으로 하는 경우 이 필드를 비워 둡니다. NetVault Backup은 구성 대화 상자에 설정된 값을 사용합니다. 자세한 내용은 **플러그인 구성** 항목을 참조하십시오.
- **다른 MySQL 인스턴스로 복원:** 선택한 데이터의 복원을 다른 인스턴스로 재배포하려는 경우 새 인스턴스에 대한 액세스를 허용하는 **사용자 이름** 및 **암호** 필드에 해당 정보를 입력합니다. 또한 **인스턴스 이름** 필드에서 새 인스턴스에 대해 설정된 NetVault Backup 이름을 입력합니다. 이 이름은 구성 대화 상자에서 **MySQL 인스턴스 이름**으로 설정된 이름입니다. 자세한 내용은 **플러그인 구성**을 참조하십시오.

**i | 중요:** 다른 MySQL 인스턴스에 대한 재배포 복원을 시도하기 전에 **대체 MySQL Server로 복구**를 검토해야 합니다.

## MySQL Enterprise Backup에 대한 복원 옵션 설정

**선택 집합 만들기** 페이지에서 **플러그인 옵션 편집**을 클릭하고 **옵션** 탭에서 해당 매개 변수를 구성합니다.

**i | 중요:** 복원을 수행하기 전에 기본 NetVault Backup **임시 디렉터리**에 **MySQL Enterprise Backup** 옵션을 사용하여 생성된 전체 백업에 포함된 모든 데이터를 임시로 저장하기에 충분한 공간이 있는지 확인합니다. **일반** 옵션을 사용하여 기본 설정을 충분한 공간을 제공하는 위치로 변경하여 매핑된 드라이브, NFS(Network File System) 또는 SMB 마운트를 사용할 수도 있습니다. 탐색 창에서 **설정 변경**을 클릭하고 **클라이언트 설정**을 클릭한 다음 **시스템 및 보안** 섹션에서 **일반**을 클릭합니다.

- **전체 복원:** 해당 옵션을 선택합니다.
  - **복원, 원시 전체 백업 추출...(기본 선택):** MySQL Server 데이터 리포지토리 디렉터리 계층 구조를 미러링하는 임시 위치에 전체 백업을 복원하려면 이 옵션을 선택합니다. 이 옵션은 사용자가 어떤 백업을 복원할지 알고 있다고 가정합니다. 그렇지 않은 경우 다음 두 옵션을 사용할 수 있습니다.
  - **전체 백업 이미지를 임시 파일로 복원:** 다음 옵션을 실행하는 데 필요한 백업을 결정하기 위해 백업 내용을 나열해야 하는 경우 이 옵션을 선택하십시오.
  - **임시 파일에서 원시 전체 백업 추출:** 위의 옵션을 사용하여 복원해야 하는 백업을 결정한 후 이 옵션을 선택합니다. 이 옵션은 MySQL Server 데이터 리포지토리 디렉터리 계층 구조를 미러링하는 임시 위치에 전체 백업을 복원합니다.
  - **MySQL Server 종료 및 다시 복사...(표준 전체 복원에 사용할 수 있는 옵션):** MySQL Server를 종료하고 복원된 내용을 임시 위치에서 원래 위치로 복사할 준비가 되었다면 이 옵션을 선택합니다.
  - **MySQL Server 리포지토리에 준비된 전체 백업 다시 복사(TTS 전체 복원에 사용할 수 있는 옵션):** 복원된 내용을 임시 위치에서 원래 위치로 복사할 준비가 되었다면 이 옵션을 선택합니다. 다음과 같은 두 가지 추가 옵션을 사용할 수 있습니다.

- **테이블 포함:** 부분 복원을 수행하려면 이 필드에 정규식을 입력하여 복원에 포함할 테이블의 명명 패턴을 설명하십시오. 이 필드를 완성하면 플러그인에서 "--include-tables" MySQL 명령을 실행합니다.
- **테이블 이름 바꾸기:** 복원할 테이블을 지정하기 위해 **테이블 포함** 필드를 완성한 경우, 이 필드를 사용하여 지정된 테이블 중 하나의 이름을 변경할 수 있습니다. 테이블의 이름을 바꾸려면 `original_name to new_name` 식을 사용합니다. 이 필드를 완성하면 플러그인에서 "--rename" MySQL 명령을 실행합니다.

**i | 중요: MySQL Server 종료 및 다시 복사...** 옵션을 사용하여 복원하려는 TTS 백업을 선택하면 플러그인이 **복원 선택** 대화 상자에서 선택한 항목을 무시합니다. 이 플러그인은 **테이블 포함** 필드에 지정된 테이블과 해당 데이터베이스만 복원합니다.

- **백업 이미지 유효성 검사:** 추출된 데이터에 대해 유효성 검사 명령을 실행하도록 플러그인에 지시하려면 이 확인란을 선택합니다.
- **백업 이미지 나열:** 출력 로그에 백업 내용을 나열하려면 이 확인란을 선택합니다.
- **증분 복원:** 해당 옵션을 선택합니다.
  - **복원, 증분 백업 추출...(기본 선택):** 증분 백업을 복원하려면 이 옵션을 선택합니다. 이 옵션은 사용자가 어떤 백업을 복원할지 알고 있다고 가정합니다. 그렇지 않은 경우 다음 두 옵션을 사용할 수 있습니다.
  - **증분 백업 이미지를 임시 파일로 복원:** 다음 옵션을 실행하는 데 필요한 백업을 결정하기 위해 백업 내용을 나열해야 하는 경우 이 옵션을 선택하십시오.
  - **임시 파일에서 증분 백업 추출...:** 위의 옵션을 사용하여 복원해야 하는 백업을 결정한 후 이 옵션을 선택합니다.
  - **MySQL Server 종료 및 다시 복사...:** MySQL Server를 종료하고 복원된 내용을 임시 위치에서 원래 위치로 복사할 준비가 되었다면 이 옵션을 선택합니다.
  - **백업 이미지 유효성 검사:** 추출된 데이터에 대해 유효성 검사 명령을 실행하도록 플러그인에 지시하려면 이 확인란을 선택합니다.
  - **백업 이미지 나열:** 출력 로그에 백업 내용을 나열하려면 이 확인란을 선택합니다.

## 직업 마무리 및 제출

마지막 단계에는 일정, 소스 옵션 및 고급 옵션 페이지에 대한 추가 옵션 설정, 작업 제출 및 작업 상태와 로그 보기 페이지를 통한 진행 상태 모니터링이 포함됩니다. 이러한 페이지 및 옵션은 모든 NetVault Backup 플러그인에 공통입니다. 자세한 내용은 *Quest NetVault Backup 관리자 안내서*를 참조하십시오.

- 1 설정을 저장하려면 **확인**을 클릭 한 후 **다음**을 클릭합니다.
- 2 기본 설정을 사용하지 않으려는 경우 **작업 이름**에 작업의 이름을 지정합니다.

진행 상황을 모니터링할 때 작업을 쉽게 식별할 수 있도록 설명 이름을 지정합니다. 작업 이름에는 영숫자 및 영숫자가 아닌 문자가 포함될 수 있지만 라틴 문자가 아닌 문자는 포함될 수 없습니다. Linux에서 이름의 길이는 최대 200자입니다. Windows에서는 길이 제한이 없습니다. 그러나 모든 플랫폼에서 최대 40자를 권장합니다.

**i | 중요:** 대상 OS의 파일 이름에 지원되지 않는 특수 문자를 사용하지 마십시오. 예를 들어 /, \, \*, @ 문자는 Windows에서 사용할 수 없습니다. 이 요구 사항은 *MySQL 용* 플러그인-이 데이터를 임시로 복원하기 위해 작업 제목과 동일한 이름의 폴더를 생성하려고 하기 때문입니다.

- 3 **대상 클라이언트** 목록에서 데이터를 복원할 시스템을 선택합니다.

**i | 팁:** 선택을 클릭한 다음 **대상 클라이언트 선택** 대화 상자에서 해당 클라이언트를 찾아 선택합니다.

- 4 **일정, 소스 옵션 및 고급 옵션** 목록을 사용하여 필요한 추가 옵션을 구성합니다.
- 5 해당되는 경우 **저장** 또는 **저장 및 제출**을 클릭합니다.

**작업 상태** 페이지에서 진행률을 모니터링하고 **로그 보기** 페이지에서 로그를 볼 수 있습니다. 자세한 내용은 *Quest NetVault Backup 관리자 안내서*를 참조하십시오.

**i** **중요:** Linux 또는 UNIX 환경에서 MySQL Enterprise Backup을 사용하는 경우 복원된 데이터에 대한 파일 소유권 및 권한 정보가 데이터 백업 전의 정보와 일치하는지 확인하십시오. **mysqlbackup** 유틸리티에서 백업 프로세스 중에 이 정보를 기록하지 않으므로 복원이 완료된 후에 해당 정보가 다를 수 있습니다. 자세한 내용은 [https://docs.oracle.com/cd/E17952\\_01/mysql-enterprise-backup-3.11-en/bugs.backup.html](https://docs.oracle.com/cd/E17952_01/mysql-enterprise-backup-3.11-en/bugs.backup.html) 페이지를 참조하십시오.

## MySQL Standard/Community 복원 시나리오 예시

실패 또는 데이터 손상에서 성공적으로 복구하려면 복원을 위해 선택한 데이터 및 **옵션** 탭에서 사용 가능한 옵션과 관련하여 작업을 설정할 때 다양한 설정을 수행해야 합니다. 다음 항목에서는 다양한 복구 유형의 예시를 제공하고 필요한 특정 옵션을 다룹니다.

- 전체 백업 전용 복원 시나리오
- 전체 및 증분 백업 복원 시나리오
- 전체 및 차등 백업 복원 시나리오
- MIXED 바이너리 로깅 형식이 사용되고 데이터베이스 간 업데이트가 실행되는 경우의 PIT 복원

### 전체 백업 전용 복원 시나리오

다음 예에서 MySQL DBA는 매일 오후 11시에 전체 백업을 수행하는 백업 정책을 설정했습니다.

#### 전체 백업 복원 및 시간 기반 특정 시점 복구

DBA는 월요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 자신이 작업장에 도착하기 전인 **월요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

##### 방법 1: 오류 문 이전으로 복구

DBA는 Drop Table 명령이 실행되기 **직전**의 시점까지 복구하기로 결정합니다. 이 결정은 DBA가 일요일의 전체 백업을 복원하고 현재 바이너리 로그에서 PIT 복구를 수행해야 함을 의미합니다.

- 1 **일요일 밤에 수행된 전체 복원 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **현재 바이너리 로그에서 PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **시간 기반 PIT:** 유형으로 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 날짜/시간**을 "5:59" 및 "8 Jan 2007"로 설정합니다. 즉, 월요일 오전 6시의 1분 전입니다.
- 3 작업을 제출합니다.

## 방법 2: 오류 문 이전 및 이후로 복원

DBA는 Drop Table 명령이 실행되기 **직전**의 시점까지 복구하기로 결정합니다. 또한 DBA는 오류 문이 실행된 시점 **이후**부터 현재의 바이너리 로그가 끝날 때까지 나머지 테이블에 발생한 트랜잭션을 복구하려고 합니다. 이러한 결정을 통해 삭제된 테이블을 복구하는 것 외에도 가능한 한 많은 트랜잭션을 복구할 수 있습니다.

- 1 **일요일 밤에 수행된 전체 복원 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **현재 바이너리 로그에서 PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **시간 기반 PIT:** 유형으로 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 날짜/시간을 "5:59" 및 "8 Jan 2007"**로 설정합니다. 즉, 월요일 오전 6시의 1분 전입니다.
  - **오류/잘못된 SQL 문 이후로 복구 활성화:** 주문 테이블이 삭제된 이후에 발생한 트랜잭션을 복구하도록 선택하고 **이후 시간과 날짜를 시작 날짜/시간에** 입력했습니다. 마지막으로, 명명된 바이너리 로그 마지막까지 복구를 수행하기 때문에 **중지 날짜/시간에 대해 없음** 옵션이 선택되었습니다.

## 전체 복원 및 위치 기반 특정 시점 복구

DBA는 월요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 자신이 작업장에 도착하기 전인 **월요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

### 방법 1: 오류 문 이전으로 복구

DBA는 Drop Table 명령이 실행되기 **직전**의 시점까지 복구하기로 결정합니다. 또한 DBA는 개발자가 테이블을 삭제한 예측 시간보다 더 정확한 복구를 원하므로 위치 기반 복구를 사용하도록 선택합니다. 이 프로세스를 실행하려면 DBA가 일요일의 전체 백업을 복원하고 현재 바이너리 로그에서 PIT 복구를 수행해야 함을 의미합니다.

- 1 **현재 바이너리 로그에 mysqlbinlog 유틸리티 사용:** 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 **위치**를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 *MySQL 참조 안내서*를 참조하십시오.) 이 프로세스에서 DBA는 Drop Table 명령을 "**MYSQLSVR-bin.000009**" 바이너리 로그의 로그 위치 "**805**"로 식별했습니다.
- 2 **일요일 밤에 수행된 전체 복원 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 3 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **현재 바이너리 로그에서 PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **위치 기반 PIT:** 유형으로 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 위치를, mysqlbinlog를** 사용하여 식별된 위치 **앞**에 있는 "**804**"로 설정하십시오. **중지 위치가 포함된 바이너리 로그를 다른 파일로** 설정하고 텍스트 상자에 대상 바이너리 파일 이름(예: "**MYSQLSVR-bin.000009**")을 입력합니다.
    - i | 중요:** 중지 및 시작 위치는 원치 않는 트랜잭션의 위치보다 큰 임의의 숫자가 아니라 바이너리 로그에 나열된 **실제 위치**여야 합니다.
- 4 **작업을 제출합니다.**

## 방법 2: 오류 문 이전 및 이후로 복원

DBA는 Drop Table 명령이 실행되기 직전의 시점까지 복구하기로 결정합니다. 또한 DBA는 주문 테이블이 삭제된 시점 이후부터 현재의 바이너리 로그가 끝날 때까지 나머지 테이블에 발생한 트랜잭션을 복구하려고 합니다. 이러한 결정을 통해 삭제된 테이블을 복구하는 것 외에도 가능한 한 많은 트랜잭션을 복구할 수 있습니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 이 프로세스를 실행하려면 DBA가 일요일의 전체 백업을 복원하고 현재 바이너리 로그에서 PIT 복구를 수행해야 함을 의미합니다.

- 1 현재 바이너리 로그에 **mysqlbinlog** 유틸리티 사용: 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 위치를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 *MySQL 참조 안내서*를 참조하십시오.) 이 프로세스에서 DBA는 Drop Table 명령을 "MYSQLSVR-PM-bin.000009" 바이너리 로그의 로그 위치 "805"로 식별했습니다.
- 2 일요일 밤에 수행된 전체 복원 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 3 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - 현재 바이너리 로그에서 PIT 복구 수행: 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - 위치 기반 PIT: 유형으로 선택합니다.
  - 오류/잘못된 SQL 문 이전으로 복구 활성화: 이 옵션을 선택하고 중지 위치를, **mysqlbinlog**를 사용하여 식별된 위치 옆에 있는 "804"로 설정하십시오. 중지 위치가 포함된 바이너리 로그를 다른 파일로 설정하고 텍스트 상자에 대상 바이너리 파일 이름(예: "MYSQLSVR-PM-bin.000009")을 입력합니다.
  - 오류/잘못된 SQL 문 이후로 복구 활성화: 이 옵션을 선택하고 시작 위치를, **mysqlbinlog**를 사용하여 식별된 위치 옆에 있는 "806"로 설정하십시오. 시작 위치가 포함된 바이너리 로그를 다른 파일로 설정하고 텍스트 상자에 대상 바이너리 파일 이름(예: "MYSQLSVR-bin.000009")을 입력합니다. 마지막으로, 명명된 바이너리 로그 마지막까지 복구를 수행하기 때문에 중지 위치에 대해 없음 옵션이 선택되었습니다.
    - i | **중요:** 중지 및 시작 위치는 원치 않는 트랜잭션의 위치보다 큰 임의의 숫자가 아니라 바이너리 로그에 나열된 실제 위치여야 합니다.
- 4 작업을 제출합니다.

## 전체 및 증분 백업 복원 시나리오

DBA는 매주 일요일 오후 11시에 전체 백업을, 월요일부터 토요일까지 오후 11시에 증분 백업을 수행하는 백업 정책을 설정했습니다. DBA는 증분 백업을 수행하고 있으므로 각 증분 백업 이후에 바이너리 로그가 삭제됩니다. 이 프로세스는 전체 백업의 속도를 향상시키지만 복원을 수행할 때는 더 많은 시간과 단계가 필요합니다.

### 전체 및 증분 복원 전용

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 자신이 작업장에 도착하기 전인 목요일 오전 이른 시간에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 마지막 증분 백업, 즉 수요일 밤에 수행된 백업 시점에서의 전체 복구를 수행하기로 결정합니다.

#### 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 2단계: 월요일의 증분 복원

- 1 월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 3단계: 화요일의 증분 복원

- 1 화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 4단계: 수요일의 증분 복원

- 1 수요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출합니다.

## 전체 복원 및 시간 기반 특정 시점 복구

다음 예에서는 전체 백업 및 증분 백업 시나리오가 제시되며 DBA는 데이터를 특정 시점으로 복구하려고 합니다.

### 방법 1: 복원된 바이너리 로그만 사용하여 오류 문 이전으로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **수요일 오후 8시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 개발자가 테이블을 삭제한 **수요일 오후 8시 직전** 시점까지 데이터베이스를 복원하는 복구를 수행해야 합니다. 따라서 다음과 같은 단계를 수행합니다.

## 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 2단계: 월요일의 증분 복원

- 1 월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 3단계: 화요일의 증분 복원

- 1 화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.

- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 4단계: 수요일의 시간 기반 PIT 복원

- 1 수요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - PIT 복구 수행: PIT 복구를 지정하고 모든 관련 옵션을 설정합니다.
  - 바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용): 사용할 백업에 포함된 바이너리 로그를 지정합니다.
  - 시간 기반 PIT: 유형으로 선택합니다.
  - 오류/잘못된 SQL 문 이전으로 복구 활성화: 이 옵션을 선택하고 중지 날짜/시간을 "19:59" 및 "10 Jan 2007"로 설정합니다. 즉, 수요일 오후 8시의 1분 전입니다.
- 3 작업을 제출합니다.

#### 방법 2: 복원된 바이너리 로그만 사용하여 오류 문 이전 및 이후로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 수요일 오후 8시에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행된 오후 8시 직전 시점으로 복구하기로 결정했습니다. 또한 DBA는 주문 테이블이 삭제된 이후 시점부터 백업 바이너리 로그 마지막까지 남아 있는 테이블에 발생한 트랜잭션도 복구하려고 합니다. 이러한 결정을 통해 삭제된 테이블을 복구하는 것 외에도 가능한 한 많은 트랜잭션을 복구할 수 있습니다. 따라서 다음과 같은 단계를 수행합니다.

#### 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 2단계: 월요일의 증분 복원

- 1 월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 3단계: 화요일의 증분 복원

- 1 화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 4단계: 수요일의 시간 기반 PIT 복원

- 1 수요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - PIT 복구 수행: PIT 복구를 지정하고 모든 관련 옵션을 설정합니다.
  - 바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용): 사용할 백업에 포함된 바이너리 로그를 지정합니다.
  - 시간 기반 PIT: 유형으로 선택합니다.
  - 오류/잘못된 SQL 문 이전으로 복구 활성화: 이 옵션을 선택하고 중지 날짜/시간을 "19:59" 및 "10 Jan 2007"로 설정합니다. 즉, 수요일 오후 8시의 1분 전입니다.
  - 오류/잘못된 SQL 문 이후로 복구 활성화: 주문 테이블이 삭제된 이후에 발생한 트랜잭션을 복구하도록 선택하고 이후 시간과 날짜를 시작 날짜/시간에 입력했습니다. 마지막으로, 백업에 포함된 바이너리 로그 마지막까지 복구를 수행하기 때문에 중지 날짜/시간에 대해 없음 옵션이 선택되었습니다.
- 3 작업을 제출합니다.

#### 방법 3: 복원된 바이너리 로그 및 현재 바이너리 로그를 사용하여 오류 문 이전으로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **목요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 개발자가 테이블을 삭제한 **목요일 오전 6시 직전** 시점까지 데이터베이스를 복원하는 복구를 수행해야 합니다.

#### 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 2단계: 월요일의 증분 복원

- 1 월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 3단계: 화요일의 증분 복원

- 1 화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 4단계: 수요일의 시간 기반 PIT 복원

- 1 수요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - PIT 복구 수행: PIT 복구를 지정하고 모든 관련 옵션을 설정합니다.
  - 바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용): 사용할 백업에 포함된 바이너리 로그를 지정하도록 선택합니다.
  - 현재 바이너리 로그 포함: 이 옵션을 선택하면 현재 바이너리 로그를 사용하여 수요일의 백업 완료 시간과 Drop Table 명령이 실행된 시간 사이에 발생한 항목을 적용할 수 있습니다.
  - 시간 기반 PIT: 유형으로 선택합니다.
  - 오류/잘못된 SQL 문 이전으로 복구 활성화: 이 옵션을 선택하고 중지 날짜/시간을 "5:59" 및 "11 Jan 2007"로 설정합니다. 즉, 목요일 오전 6시의 1분 전입니다.
- 3 작업을 제출합니다.

#### 방법 4: 복원된 바이너리 로그 및 현재 바이너리 로그를 사용하여 오류 문 이전 및 이후로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **목요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행되기 직전의 시점까지 복구하기로 결정합니다. 또한 DBA는 주문 테이블이 삭제된 시점 이후부터 현재의 바이너리 로그가 끝날 때까지 나머지 테이블에 발생한 트랜잭션을 복구하려고 합니다. 이러한 결정을 통해 삭제된 테이블을 복구하는 것 외에도 가능한 한 많은 트랜잭션을 복구할 수 있습니다. 따라서 다음과 같은 단계를 수행합니다.

#### 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 2단계: 월요일의 증분 복원

- 1 월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 3단계: 화요일의 증분 복원

- 1 화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 4단계: 수요일의 시간 기반 PIT 복원

- 1 수요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - PIT 복구 수행: PIT 복구를 지정하고 모든 관련 옵션을 설정합니다.
  - 바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용): 사용할 백업에 포함된 바이너리 로그를 지정하도록 선택합니다.
  - 현재 바이너리 로그 포함: 이 옵션을 선택하면 현재 바이너리 로그를 사용하여 수요일의 백업 완료 시간과 Drop Table 명령이 실행된 시간 사이에 발생한 항목을 적용할 수 있습니다.
  - 시간 기반 PIT: 유형으로 선택합니다.
  - 오류/잘못된 SQL 문 이전으로 복구 활성화: 이 옵션을 선택하고 중지 날짜/시간을 "5:59" 및 "11 Jan 2007"로 설정합니다. 즉, 목요일 오전 6시의 1분 전입니다.
  - 오류/잘못된 SQL 문 이후로 복구 활성화: 주문 테이블이 삭제된 이후에 발생한 트랜잭션을 복구하도록 선택하고 이후 시간과 날짜를 시작 날짜/시간에 입력했습니다. 마지막으로, 현재 바이너리 로그 마지막까지 복구를 수행하기 때문에 중지 날짜/시간에 대해 없음 옵션이 선택되었습니다.
- 3 작업을 제출합니다.

#### 전체 복원 및 위치 기반 특정 시점 복구

다음 예에서는 전체 백업 및 증분 백업 시나리오가 제시되며 DBA는 데이터를 특정 시점으로 복구하려고 합니다. 하지만 보다 명확한 방법으로 시간을 정의해야 합니다. 이 복구는 MySQL 바이너리 로그에 존재하는 식별된 "위치 값"을 사용하여 수행됩니다.

##### 방법 1: 복원된 바이너리 로그만 사용하여 오류 문 이전으로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 수요일 오후 8시에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행되기 직전의 시점까지 복구하기로 결정합니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 이 프로세스를 수행하려면 DBA가 일요일의 전체 백업을 복원해야 하고, 이어서 월요일과 화요일에 수행된 증분 백업을 실시한 다음, 수요일 증분 백업을 사용하여 위치 기반 PIT 복구를 수행해야 합니다. 다음 단계에서는 이 프로세스를 설명합니다.

#### 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 2단계: 월요일의 증분 복원

- 1 월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 3단계: 화요일의 증분 복원

- 1 화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 4단계: 백업된 바이너리 로그를 복원하여 오류 문의 위치를 파악합니다.

이 단계에서는 수요일 밤의 증분 백업에 기록된 바이너리 로그만 임시 위치에 복원됩니다. 이 프로세스를 통해 DBA는 주문 테이블이 삭제된 시점을 표시하는 로그의 특정 위치를 찾을 수 있습니다.

- 1 수요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - PIT 복구 수행: 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - 시간 또는 위치를 식별하기 위해 로그를 임시 디렉터리로 복원: 수요일 밤의 증분 백업에 포함된 바이너리 로그만 복원하도록 선택합니다.
  - 시간 기반 PIT: 유형으로 선택했지만 시간 기반 PIT 세부 정보 섹션에 있는 모든 옵션이 **삭제**되었습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 5단계: 복원된 바이너리 로그에서 Drop Table 명령의 위치 식별

복원된 바이너리 로그에 **mysqlbinlog** 유틸리티 사용: 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 **위치**를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 *MySQL 참조 안내서*를 참조하십시오.) 이 프로세스에서 DBA는 MySQL Server의 임시 위치로 복원된 **"MYSQLSVR-bin.000009"** 바이너리 로그의 로그 위치 **"805"**로 Drop Table 명령을 식별했으며 두 값이 모두 기록되었습니다.

### 6단계: 위치 기반 PIT 복원 수행

복원된 바이너리 로그에서 위치를 식별하고 수요일의 증분 백업을 사용하여 PIT 복원을 수행합니다.

- 1 수요일 밤에 수행된 증분 백업 선택: DBA가 복원 작업 생성 - 저장 집합 선택 페이지 페이지에서 수요일의 증분 백업에 해당하는 백업 저장 집합을 다시 선택합니다.
- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - PIT 복구 수행: 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - 임시 디렉터리의 바이너리 로그 적용: 이 절차의 마지막 단계에서 임시 위치로 복원된 바이너리 로그를 대상으로 하도록 선택합니다. 복원된 바이너리 로그를 사용하여 Drop Table 명령이 사용된 특정 위치를 식별했기 때문에 플러그인에 동일한 바이너리 로그를 사용하도록 지시하기 위해 이 옵션을 선택합니다.
  - 오류/잘못된 SQL 문 이전으로 복구 활성화: 이 옵션을 선택하고 중지 위치를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 **값**에 있는 바이너리 로그의 위치인 **"804"**로 설정하십시오. **중지 위치가 포함된 바이너리 로그** 옵션은 임시 디렉터리에 복원된 바이너리 로그, **"MYSQLSVR-bin.000009"**를 선택하는 데 사용되었습니다.
- 3 작업을 제출합니다.

## 방법 2: 복원된 바이너리 로그만 사용하여 오류 문 이전 및 이후로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **수요일 오후 8시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행되기 **직전**의 시점까지 복구하기로 결정합니다. 또한 DBA는 주문 테이블이 삭제된 시점 **이후**부터 백업 바이너리 로그가 끝날 때까지 나머지 테이블에 발생한 트랜잭션을 복구하려고 합니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 이 프로세스를 수행하려면 DBA가 일요일의 전체 백업을 복원해야 하고, 이어서 월요일과 화요일에 수행된 증분 백업을 실시한 다음, 수요일 증분 백업을 사용하여 위치 기반 PIT 복구를 수행해야 합니다. 다음 단계에서는 이 프로세스를 설명합니다.

### 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 2단계: 월요일의 증분 복원

- 1 **월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 3단계: 화요일의 증분 복원

- 1 **화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 4단계: 백업된 바이너리 로그를 복원하여 오류 문의 위치를 파악합니다.

이 단계에서는 수요일 밤의 증분 백업에 기록된 바이너리 로그만 임시 위치에 복원됩니다. 이 단계를 통해 DBA는 주문 테이블이 삭제된 시점을 표시하는 로그의 특정 위치를 찾을 수 있습니다.

- 1 **수요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 수요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **시간 또는 위치를 식별하기 위해 로그를 임시 디렉터리로 복원:** 수요일 밤의 증분 백업에 포함된 바이너리 로그만 복원하도록 선택합니다.
  - **시간 기반 PIT:** 유형으로 선택했지만 **시간 기반 PIT 세부 정보** 섹션에 있는 모든 옵션이 **삭제**되었습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 5단계: 복원된 바이너리 로그에서 Drop Table 명령의 위치 식별

**복원된 바이너리 로그에 mysqlbinlog 유틸리티 사용:** 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 **위치**를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 *MySQL 참조 안내서*를 참조하십시오.) 이 프로세스에서 DBA는 MySQL Server의 임시 위치로 복원된 **"MYSQLSVR-bin.000009"** 바이너리 로그의 로그 위치 **"805"**로 Drop Table 명령을 식별했으며 두 값이 모두 기록되었습니다.

## 6단계: 위치 기반 PIT 복원 수행

복원된 바이너리 로그에서 위치를 식별하고 수요일의 증분 백업을 사용하여 PIT 복원을 수행합니다.

- 1 **수요일 밤에 수행된 증분 백업 선택:** DBA가 **복원 작업 생성 - 저장 집합 선택 페이지** 페이지에서 수요일의 증분 백업에 해당하는 백업 저장 집합을 다시 선택합니다.
- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **임시 디렉터리의 바이너리 로그 적용:** 이 절차의 마지막 단계에서 임시 위치로 복원된 바이너리 로그를 대상으로 하도록 선택합니다. 복원된 바이너리 로그를 사용하여 Drop Table 명령이 사용된 특정 위치를 식별했기 때문에 플러그인에 동일한 바이너리 로그를 사용하도록 지시하기 위해 이 옵션을 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 위치**를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 **앞**에 있는 바이너리 로그의 위치인 **"804"**로 설정하십시오. **중지 위치가 포함된 바이너리 로그** 옵션은 임시 디렉터리에 복원된 바이너리 로그, **"MYSQLSVR-bin.000009"**를 선택하는 데 사용되었습니다.
  - **오류/잘못된 SQL 문 이후로 복구 활성화:** 이 옵션을 선택하고 **시작 위치**를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 **뒤**에 있는 바이너리 로그의 위치인 **"806"**으로 설정하십시오. **중지 위치가 포함된 바이너리 로그** 옵션은 임시 디렉터리에 복원된 바이너리 로그, **"MYSQLSVR-bin.000009"**를 선택하는 데 사용되었습니다. 마지막으로, 명명된 바이너리 로그 마지막까지 복구를 수행하기 때문에 **중지 날짜/시간**에 대해 **없음** 옵션이 선택되었습니다.

**i | 중요:** 중지 및 시작 위치는 원치 않는 트랜잭션의 위치보다 큰 임의의 숫자가 아니라 바이너리 로그에 나열된 **실제 위치**여야 합니다.

- 3 **작업을 제출합니다.**

## 방법 3: 복원된 바이너리 로그 및 현재 바이너리 로그를 사용하여 오류 문 이전으로 복구

DBA는 목요일 오전 9시에 주문 테이블에 **"테이블을 찾을 수 없음"** 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **목요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 개발자가 테이블을 삭제한 **목요일 오전 6시 직전** 시점까지 데이터베이스를 복원하는 복구를 수행해야 합니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 이 프로세스를 수행하려면 DBA가 일요일의 전체 백업을 복원해야 하고, 이어서 월요일과 화요일에 수행된 증분 백업을 실시한 다음, 수요일 증분 백업을 사용하여 위치 기반 PIT 복구를 수행해야 합니다. 다음 단계에서는 이 프로세스를 설명합니다.

## 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택:** **복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 **작업을 제출하고 완료될 때까지 기다립니다.**

## 2단계: 월요일의 증분 복원

- 1 월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 3단계: 화요일의 증분 복원

- 1 화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 4단계: 현재 바이너리 로그에서 Drop Table 명령의 위치 식별

현재 바이너리 로그에 **mysqlbinlog** 유틸리티 사용: 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 **위치**를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 *MySQL 참조 안내서*를 참조하십시오.) 이 프로세스에서 DBA는 Drop Table 명령을 현재 바이너리 로그 **"MYSQLSVR-bin.000009"**의 로그 위치 **"805"**로 식별했습니다.

## 5단계: 위치 기반 PIT 복원 수행

복원된 바이너리 로그에서 위치를 식별하고 수요일의 증분 백업을 사용하여 PIT 복원을 수행합니다.

- 1 수요일 밤에 수행된 증분 백업 선택: DBA가 복원 작업 생성 - 저장 집합 선택 페이지 페이지에서 수요일의 증분 백업에 해당하는 백업 저장 집합을 다시 선택합니다.
- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용):** 백업에 포함된 바이너리 로그를 사용하도록 플러그인에 지시하기 위해 선택합니다.
  - **현재 바이너리 로그 포함:** 현재 바이너리 로그를 사용하여 수요일 밤의 증분 백업 **이후**에 발생한 모든 데이터베이스 트랜잭션을 적용하도록 NetVault Backup에 지시하기 위해 선택합니다. 이 단계는 수요일 밤의 증분 백업 완료 시간과 Drop Table 명령이 실행된 시간 사이에 발생한 모든 트랜잭션을 복구합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고, **중지 위치**를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 **앞**에 있는 현재 바이너리 로그의 위치인 **"804"**로 설정하십시오. **중지 위치가 포함된 바이너리 로그를 다른 파일로 설정**하고 텍스트 상자에 현재 바이너리 파일 이름 (예: **"MYSQLSVR-bin.000009"**)을 입력합니다.

## 방법 4: 복원된 바이너리 로그 및 현재 바이너리 로그를 사용하여 오류 문 **이전** 및 **이후**로 복구

DBA는 목요일 오전 9시에 주문 테이블에 **"테이블을 찾을 수 없음"** 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **목요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 개발자가 테이블을 삭제한 **목요일 오전 6시 직전** 시점까지 데이터베이스를 복원하는 복구를 수행해야 합니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 이 프로세스를 수행하려면 DBA가 일요일의 전체 백업을 복원해야 하고, 이어서 월요일과 화요일에 수행된 증분 백업을 실시한 다음, 수요일 증분 백업을 사용하여 위치 기반 PIT 복구를 수행해야 합니다. 다음 단계에서는 이 프로세스를 설명합니다.

### 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 2단계: 월요일의 증분 복원

- 1 월요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 월요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 3단계: 화요일의 증분 복원

- 1 화요일 밤에 수행된 증분 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 화요일의 증분 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 4단계: 현재 바이너리 로그에서 Drop Table 명령의 위치 식별

현재 바이너리 로그에 `mysqlbinlog` 유틸리티 사용: 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 위치를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 MySQL 참조 안내서를 참조하십시오.) 이 프로세스에서 DBA는 Drop Table 명령을 현재 바이너리 로그 "MYSQSVR-bin.000009"의 로그 위치 "805"로 식별했습니다.

### 5단계: 위치 기반 PIT 복원 수행

복원된 바이너리 로그에서 위치를 식별하고 수요일의 증분 백업을 사용하여 PIT 복원을 수행합니다.

- 1 수요일 밤에 수행된 증분 백업 선택: DBA가 복원 작업 생성 - 저장 집합 선택 페이지 페이지에서 수요일의 증분 백업에 해당하는 백업 저장 집합을 다시 선택합니다.
- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - PIT 복구 수행: 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - 바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용): 백업에 포함된 바이너리 로그를 사용하도록 플러그인에 지시하기 위해 선택합니다.
  - 현재 바이너리 로그 포함: 현재 바이너리 로그를 사용하여 수요일 밤의 증분 백업 이후에 발생한 모든 데이터베이스 트랜잭션을 적용하도록 NetVault Backup에 지시하기 위해 선택합니다. 이 단계는 수요일 밤의 증분 백업 완료 시간과 Drop Table 명령이 실행된 시간 사이에 발생한 모든 트랜잭션을 복구합니다.
  - 오류/잘못된 SQL 문 이전으로 복구 활성화: 이 옵션을 선택하고, 중지 위치를, `mysqlbinlog`를 사용하여 식별된 Drop Table 명령 위치 *앞*에 있는 현재 바이너리 로그의 위치인 "804"로 설정하십시오. 중지 위치가 포함된 바이너리 로그를 다른 파일로 설정하고 텍스트 상자에 현재 바이너리 파일 이름(예: "MYSQSVR-bin.000009")을 입력합니다.
  - 오류/잘못된 SQL 문 이후로 복구 활성화: 이 옵션을 선택하고, 시작 위치를, `mysqlbinlog`를 사용하여 식별된 Drop Table 명령 위치 *뒤*에 있는 현재 바이너리 로그의 위치인 "806"으로 설정하십시오. 중지 위치가 포함된 바이너리 로그를 다른 파일로 설정하고 텍스트 상자에 현재 바이너리 파일 이름(예: "MYSQSVR-bin.000009")을 입력합니다. 마지막으로, 현재 바이너리 로그 마지막까지 복구를 수행하기 때문에 중지 위치에 대해 **없음** 옵션이 선택되었습니다.

**i | 중요:** 중지 및 시작 위치는 원치 않는 트랜잭션의 위치보다 큰 임의의 숫자가 아니라 바이너리 로그에 나열된 *실제 위치*여야 합니다.

## 전체 및 차등 백업 복원 시나리오

DBA는 **매주 일요일 오후 11시에 전체 백업**을, **월요일부터 토요일까지 오후 11시에 차등 백업**을 수행하는 백업 정책을 설정했습니다. DBA가 차등 백업을 수행하고 있으므로 각 차등 백업 이후에 바이너리 로그가 유지되므로 백업 시간은 더 길어지지만 전반적으로 복원 속도가 빨라집니다.

### 전체 및 차등 복원 전용

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 자신이 작업장에 도착하기 전인 목요일 오전 이른 시간에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 마지막 차등 백업, 즉 **수요일** 밤에 수행된 백업 시점에서의 전체 복구를 수행하기로 결정합니다.

#### 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택:** 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 2단계: 수요일의 증분 복원

- 1 **수요일 밤에 수행된 차등 백업 선택:** 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 차등 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 옵션 탭에서 사용할 수 있는 옵션이 **사용되지 않습니다**.
- 3 작업을 제출합니다.

**i | 중요:** DBA가 **월요일**과 **화요일** 밤의 차등 백업을 복원하지 *않아도* 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 *그리고* 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.

## 전체 복원 및 시간 기반 특정 시점 복구

다음 예에서는 전체 백업 및 차등 백업 시나리오가 제시되며 DBA는 데이터를 특정 시점으로 복구하려고 합니다.

### 방법 1: 복원된 바이너리 로그만 사용하여 오류 문 이전으로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **수요일 오후 8시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 개발자가 테이블을 삭제한 **수요일 오후 8시 직전** 시점까지 데이터베이스를 복원하는 복구를 수행해야 합니다. 따라서 다음과 같은 단계를 수행합니다.

#### 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택:** 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.

- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 2단계: 수요일의 시간 기반 PIT 복원

- 1 수요일 밤에 수행된 차등 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 차등 백업에 해당하는 백업 저장 집합을 선택합니다.

**i** **중요:** DBA가 월요일과 화요일 밤의 차등 백업을 복원하지 *않아도* 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 *그리고* 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.

- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.
  - PIT 복구 수행: PIT 복구를 지정하고 모든 관련 옵션을 설정합니다.
  - 바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용): 사용할 백업에 포함된 바이너리 로그를 지정합니다.
  - 시간 기반 PIT: 유형으로 선택합니다.
  - 오류/잘못된 SQL 문 이전으로 복구 활성화: 이 옵션을 선택하고 중지 날짜/시간을 "19:59" 및 "10 Jan 2007"로 설정합니다. 즉, 수요일 오후 8시의 1분 전입니다.

- 3 작업을 제출합니다.

## 방법 2: 복원된 바이너리 로그만 사용하여 오류 문 이전 및 이후로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **수요일 오후 8시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행된 오후 8시 **직전** 시점으로 복구하기로 결정했습니다. 또한 DBA는 주문 테이블이 삭제된 **이후** 시점부터 백업 바이너리 로그 마지막까지 남아 있는 테이블에 발생한 트랜잭션도 복구하려고 합니다. 이러한 결정을 통해 삭제된 테이블을 복구하는 것 외에도 가능한 한 많은 트랜잭션을 복구할 수 있습니다. 따라서 다음과 같은 단계를 수행합니다.

### 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

## 2단계: 수요일의 시간 기반 PIT 복원

- 1 수요일 밤에 수행된 차등 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 차등 백업에 해당하는 백업 저장 집합을 선택합니다.

**i** **중요:** DBA가 월요일과 화요일 밤의 차등 백업을 복원하지 *않아도* 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 *그리고* 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.

2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.

- **PIT 복구 수행:** PIT 복구를 지정하고 모든 관련 옵션을 설정합니다.
- **바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용):** 사용할 백업에 포함된 바이너리 로그를 지정합니다.
- **시간 기반 PIT:** 유형으로 선택합니다.
- **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 날짜/시간**을 "19:59" 및 "10 Jan 2007"로 설정합니다. 즉, 수요일 오후 8시의 1분 전입니다.
- **오류/잘못된 SQL 문 이후로 복구 활성화:** **주문** 테이블이 삭제된 **이후**에 발생한 트랜잭션을 복구하도록 선택하고 **이후** 시간과 날짜를 **시작 날짜/시간**에 입력했습니다. 마지막으로, 복원된 바이너리 로그 마지막까지 복구를 수행하기 때문에 **중지 날짜/시간**에 대해 **없음** 옵션이 선택되었습니다.

3 작업을 제출합니다.

### 방법 3: 복원된 바이너리 로그 및 현재 바이너리 로그를 사용하여 오류 문 이전으로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **목요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 개발자가 테이블을 삭제한 **목요일 오전 6시 직전** 시점까지 데이터베이스를 복원하는 복구를 수행해야 합니다.

#### 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 2단계: 수요일의 시간 기반 PIT 복원

- 1 **수요일 밤에 수행된 차등 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 수요일의 차등 백업에 해당하는 백업 저장 집합을 선택합니다.

**i** **중요:** DBA가 **월요일**과 **화요일** 밤의 차등 백업을 복원하지 *않아*도 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 *그리고* 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.

2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.

- **PIT 복구 수행:** PIT 복구를 지정하고 모든 관련 옵션을 설정합니다.
- **바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용):** 사용할 백업에 포함된 바이너리 로그를 지정하도록 선택합니다.
- **현재 바이너리 로그 포함:** 이 옵션을 선택하면 현재 바이너리 로그를 사용하여 수요일의 백업 완료 시간과 Drop Table 명령이 실행된 시간 사이에 발생한 항목을 적용할 수 있습니다.
- **시간 기반 PIT:** 유형으로 선택합니다.
- **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 날짜/시간**을 "5:59" 및 "11 Jan 2007"로 설정합니다. 즉, 목요일 오전 6시의 1분 전입니다.

3 작업을 제출합니다.

#### 방법 4: 복원된 바이너리 로그 및 현재 바이너리 로그를 사용하여 오류 문 이전 및 이후로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **목요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행되기 **직전**의 시점까지 복구하기로 결정합니다. 또한 DBA는 주문 테이블이 삭제된 시점 **이후**부터 현재의 바이너리 로그가 끝날 때까지 나머지 테이블에 발생한 트랜잭션을 복구하려고 합니다. 이러한 결정을 통해 삭제된 테이블을 복구하는 것 외에도 가능한 한 많은 트랜잭션을 복구할 수 있습니다. 따라서 다음과 같은 단계를 수행합니다.

##### 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

##### 2단계: 수요일의 시간 기반 PIT 복원

- 1 **수요일 밤에 수행된 차등 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 수요일의 차등 백업에 해당하는 백업 저장 집합을 선택합니다.

**i | 중요:** DBA가 **월요일**과 **화요일** 밤의 차등 백업을 복원하지 *않아도* 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 *그리고* 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.

- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **PIT 복구 수행:** PIT 복구를 지정하고 모든 관련 옵션을 설정합니다.
  - **바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용):** 사용할 백업에 포함된 바이너리 로그를 지정하도록 선택합니다.
  - **현재 바이너리 로그 포함:** 이 옵션을 선택하면 현재 바이너리 로그를 사용하여 수요일의 백업 완료 시간과 Drop Table 명령이 실행된 시간 사이에 발생한 항목을 적용할 수 있습니다.
  - **시간 기반 PIT:** 유형으로 선택합니다.
  - **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 날짜/시간**을 "5:59" 및 "11 Jan 2007"로 설정합니다. 즉, 목요일 오전 6시의 1분 전입니다.
  - **오류/잘못된 SQL 문 이후로 복구 활성화:** **주문** 테이블이 삭제된 **이후**에 발생한 트랜잭션을 복구하도록 선택하고 **이후** 시간과 날짜를 **시작 날짜/시간**에 입력했습니다. 마지막으로, 현재 바이너리 로그 마지막까지 복구를 수행하기 때문에 **중지 날짜/시간**에 대해 **없음** 옵션이 선택되었습니다.

- 3 작업을 제출합니다.

#### 전체 복원 및 위치 기반 특정 시점 복구

다음 예에서는 전체 백업 및 증분 백업 시나리오가 제시되며 DBA는 데이터를 특정 시점으로 복구하려고 합니다. 하지만 보다 명확한 방법으로 시간을 정의해야 합니다. 이 절차는 MySQL 바이너리 로그에 존재하는 식별된 "위치 값"을 사용하여 수행됩니다.

##### 방법 1: 복원된 바이너리 로그만 사용하여 오류 문 이전으로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **수요일 오후 8시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행되기 **직전**의 시점까지 복구하기로 결정합니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 다음 단계에서는 이 프로세스를 설명합니다.

### 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 **작업을 제출하고 완료될 때까지 기다립니다.**

### 2단계: 백업된 바이너리 로그를 복원하여 오류 문의 위치를 파악합니다.

이 단계에서는 수요일 밤의 차등 백업에 기록된 바이너리 로그만 임시 위치에 복원됩니다. 이 프로세스를 통해 DBA는 주문 테이블이 삭제된 시점을 표시하는 로그의 특정 위치를 찾을 수 있습니다.

- 1 **수요일 밤에 수행된 차등 백업 선택: 복원 작업 생성 - 저장 집합 선택** 페이지에서 DBA는 수요일의 차등 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **시간 또는 위치를 식별하기 위해 로그를 임시 디렉터리로 복원:** 수요일 밤의 차등 백업에 포함된 바이너리 로그만 복원하도록 선택합니다.
  - **시간 기반 PIT:** 유형으로 선택했지만 **시간 기반 PIT 세부 정보** 섹션에 있는 모든 옵션이 **삭제**되었습니다.
- 3 **작업을 제출하고 완료될 때까지 기다립니다.**

### 3단계: 복원된 바이너리 로그에서 Drop Table 명령의 위치 식별

**복원된 바이너리 로그에 mysqlbinlog 유틸리티 사용:** 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 **위치**를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 *MySQL 참조 안내서*를 참조하십시오.) 이 프로세스에서 DBA는 MySQL Server의 임시 위치로 복원된 **"MYSQLSVR-bin.000009"** 바이너리 로그의 로그 위치 **"805"**로 Drop Table 명령을 식별했으며 두 값이 모두 기록되었습니다.

### 4단계: 위치 기반 PIT 복원 수행

복원된 바이너리 로그에서 위치를 식별하고 수요일의 차등 백업을 사용하여 PIT 복원을 수행합니다.

- 1 **수요일 밤에 수행된 차등 백업 선택:** DBA가 **복원 작업 생성 - 저장 집합 선택 페이지** 페이지에서 수요일의 차등 백업에 해당하는 백업 저장 집합을 다시 선택합니다.
  - i** **중요:** DBA가 **월요일**과 **화요일** 밤의 차등 백업을 복원하지 **않아도** 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 **그리고** 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.
- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **임시 디렉터리의 바이너리 로그 적용:** 이 절차의 마지막 단계에서 임시 위치로 복원된 바이너리 로그를 대상으로 하도록 선택합니다. 복원된 바이너리 로그를 사용하여 Drop Table 명령이 사용된 특정 위치를 식별했기 때문에 플러그인에 동일한 바이너리 로그를 사용하도록 지시하기 위해 이 옵션을 선택합니다.

- **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 위치**를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 **앞**에 있는 바이너리 로그의 위치인 **"804"**로 설정하십시오. **중지 위치가 포함된 바이너리 로그** 옵션은 임시 디렉터리에 복원된 바이너리 로그, **"MYSQLSVR-bin.000009"**를 선택하는 데 사용되었습니다.

### 3 작업을 제출합니다.

## 방법 2: 복원된 바이너리 로그만 사용하여 오류 문 *이전* 및 *이후*로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **수요일 오후 8시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행되기 **직전**의 시점까지 복구하기로 결정합니다. 또한 DBA는 주문 테이블이 삭제된 시점 **이후**부터 백업 바이너리 로그가 끝날 때까지 나머지 테이블에 발생한 트랜잭션을 복구하려고 합니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 다음 단계에서는 이 프로세스를 설명합니다.

### 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택:** 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 2단계: 백업된 바이너리 로그를 복원하여 오류 문의 위치를 파악합니다.

이 단계에서는 수요일 밤의 증분 백업에 기록된 바이너리 로그만 임시 위치에 복원됩니다. 이 프로세스를 통해 DBA는 주문 테이블이 삭제된 시점을 표시하는 로그의 특정 위치를 찾을 수 있습니다.

- 1 **수요일 밤에 수행된 차등 백업 선택:** 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 수요일의 차등 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.
  - **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
  - **시간 또는 위치를 식별하기 위해 로그를 임시 디렉터리로 복원:** 수요일 밤의 차등 백업에 포함된 바이너리 로그만 복원하도록 선택합니다.
  - **시간 기반 PIT:** 유형으로 선택했지만 **시간 기반 PIT 세부 정보** 섹션에 있는 모든 옵션이 **삭제**되었습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

### 3단계: 복원된 바이너리 로그에서 Drop Table 명령의 위치 식별

**복원된 바이너리 로그에 mysqlbinlog 유틸리티 사용:** 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 **위치**를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 **MySQL 참조 안내서**를 참조하십시오.) 이 프로세스에서 DBA는 MySQL Server의 임시 위치로 복원된 **"MYSQLSVR-bin.000009"** 바이너리 로그의 로그 위치 **"805"**로 Drop Table 명령을 식별했으며 두 값이 모두 기록되었습니다.

#### 4단계: 위치 기반 PIT 복원 수행

복원된 바이너리 로그에서 위치를 식별하고 수요일의 증분 백업을 사용하여 PIT 복원을 수행합니다.

- 1 수요일 밤에 수행된 차등 백업 선택: DBA가 복원 작업 생성 - 저장 집합 선택 페이지 페이지에서 수요일의 차등 백업에 해당하는 백업 저장 집합을 다시 선택합니다.

**i** | **중요:** DBA가 월요일과 화요일 밤의 차등 백업을 복원하지 *않아도* 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 *그리고* 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.

- 2 복원 관련 옵션 탭에서 특정 옵션 설정: DBA가 다음 옵션을 설정합니다.

- **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
- **임시 디렉터리의 바이너리 로그 적용:** 이 절차의 마지막 단계에서 임시 위치로 복원된 바이너리 로그를 대상으로 하도록 선택합니다. 복원된 바이너리 로그를 사용하여 Drop Table 명령이 사용된 특정 위치를 식별했기 때문에 플러그인에 동일한 바이너리 로그를 사용하도록 지시하기 위해 이 옵션을 선택합니다.
- **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고 **중지 위치**를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 *앞*에 있는 바이너리 로그의 위치인 **"804"**로 설정하십시오. **중지 위치가 포함된 바이너리 로그** 옵션은 임시 디렉터리에 복원된 바이너리 로그, **"MYSQLSVR-bin.000009"**를 선택하는 데 사용되었습니다.
- **오류/잘못된 SQL 문 이후로 복구 활성화:** 이 옵션을 선택하고 **시작 위치**를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 *뒤*에 있는 바이너리 로그의 위치인 **"806"**으로 설정하십시오. **중지 위치가 포함된 바이너리 로그** 옵션은 임시 디렉터리에 복원된 바이너리 로그, **"MYSQLSVR-bin.000009"**를 선택하는 데 사용되었습니다. 마지막으로, 명명된 바이너리 로그 마지막까지 복구를 수행하기 때문에 **중지 위치**에 대해 **없음** 옵션이 선택되었습니다.

**i** | **중요:** 중지 및 시작 위치는 원치 않는 트랜잭션의 위치보다 큰 임의의 숫자가 아니라 바이너리 로그에 나열된 *실제 위치*여야 합니다.

- 3 작업을 제출합니다.

#### 방법 3: 복원된 바이너리 로그 및 현재 바이너리 로그를 사용하여 오류 문 *이전*으로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **목요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 개발자가 테이블을 삭제한 **목요일 오전 6시 직전** 시점까지 데이터베이스를 복원하는 복구를 수행해야 합니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 다음 단계에서는 이 프로세스를 설명합니다.

#### 1단계: 일요일의 전체 복원

- 1 일요일 밤에 수행된 전체 백업 선택: 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 모든 복원 관련 옵션을 기본값으로 유지: 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 2단계: 현재 바이너리 로그에서 Drop Table 명령의 위치 식별

**현재 바이너리 로그에 mysqlbinlog 유틸리티 사용:** 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 *위치*를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 *MySQL 참조 안내서*를 참조하십시오.) 이 프로세스에서 DBA는 Drop Table 명령을 현재 바이너리 로그 **"MYSQLSVR-bin.000009"**의 로그 위치 **"805"**로 식별했습니다.

### 3단계: 위치 기반 PIT 복원 수행

복원된 바이너리 로그에서 위치를 식별하고 수요일의 차등 백업을 사용하여 PIT 복원을 수행합니다.

- 1 **수요일 밤에 수행된 차등 백업 선택:** DBA가 복원 작업 생성 - 저장 집합 선택 페이지 페이지에서 수요일의 차등 백업에 해당하는 백업 저장 집합을 다시 선택합니다.

**i** **중요:** DBA가 월요일과 화요일 밤의 차등 백업을 복원하지 *않아도* 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 *그리고* 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.

- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.

- **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
- **바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용):** 백업에 포함된 바이너리 로그를 사용하도록 플러그인에 지시하기 위해 선택합니다.
- **현재 바이너리 로그 포함:** 현재 바이너리 로그를 사용하여 수요일 밤의 차등 백업 *이후*에 발생한 모든 데이터베이스 트랜잭션에 적용하도록 NetVault Backup에 지시하기 위해 선택합니다. 이 단계는 수요일 밤의 차등 백업 완료 시간과 Drop Table 명령이 실행된 시간 사이에 발생한 모든 트랜잭션을 복구합니다.
- **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고, 중지 위치를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 *앞*에 있는 현재 바이너리 로그의 위치인 **"804"**로 설정하십시오. 중지 위치가 포함된 바이너리 로그를 다른 파일로 설정하고 텍스트 상자에 현재 바이너리 파일 이름 (예: **"MYSQLSVR-bin.000009"**)을 입력합니다.

### 방법 4: 복원된 바이너리 로그 및 현재 바이너리 로그를 사용하여 오류 문 *이전* 및 *이후*로 복구

DBA는 목요일 오전 9시에 주문 테이블에 "테이블을 찾을 수 없음" 오류가 발생한 사실을 확인했습니다. 그런 다음 DBA는 **목요일 오전 6시**에 개발자가 실수로 테이블을 삭제하여 해당 테이블이 더 이상 존재하지 않는다는 것을 알게 됩니다.

DBA는 Drop Table 명령이 실행되기 *직전*의 시점까지 복구하기로 결정합니다. 또한 DBA는 주문 테이블이 삭제된 시점 *이후*부터 현재의 바이너리 로그가 끝날 때까지 나머지 테이블에 발생한 트랜잭션을 복구하려고 합니다. 또한 DBA는 보다 정확한 복원을 원하므로 위치 기반 복구를 사용하기로 결정했습니다. 다음 단계에서는 이 프로세스를 설명합니다.

#### 1단계: 일요일의 전체 복원

- 1 **일요일 밤에 수행된 전체 백업 선택:** 복원 작업 생성 - 저장 집합 선택 페이지에서 DBA는 일요일의 전체 백업에 해당하는 백업 저장 집합을 선택합니다.
- 2 **모든 복원 관련 옵션을 기본값으로 유지:** 이러한 옵션 중 어떤 것도 사용되지 않습니다.
- 3 작업을 제출하고 완료될 때까지 기다립니다.

#### 2단계: 현재 바이너리 로그에서 Drop Table 명령의 위치 식별

**현재 바이너리 로그에 mysqlbinlog 유틸리티 사용:** 이 단계는 NetVault Backup 외부에서 수행되어 DBA가 복원하지 않으려는 Drop Table 명령의 *위치*를 식별할 수 있습니다. (이 유틸리티 및 프로세스에 대한 자세한 내용은 *MySQL 참조 안내서*를 참조하십시오.) 이 프로세스에서 DBA는 Drop Table 명령을 현재 바이너리 로그 **"MYSQLSVR-bin.000009"**의 로그 위치 **"805"**로 식별했습니다.

### 3단계: 위치 기반 PIT 복원 수행

복원된 바이너리 로그에서 위치를 식별하고 수요일의 차등 백업을 사용하여 PIT 복원을 수행합니다.

- 1 **수요일 밤에 수행된 차등 백업 선택:** DBA가 **복원 작업 생성 - 저장 집합 선택 페이지** 페이지에서 수요일의 차등 백업에 해당하는 백업 저장 집합을 다시 선택합니다.

**i | 중요:** DBA가 **월요일**과 **화요일** 밤의 차등 백업을 복원하지 *않아도* 됩니다. 차등 백업을 수행하도록 선택하면 매일 밤의 백업이 누적되어 일요일 밤의 전체 백업으로 돌아갑니다. 즉, 수요일 밤의 백업에는 월요일, 화요일 *그리고* 수요일에 생성된 모든 바이너리 로그가 포함되어 일요일의 전체 백업으로 돌아갑니다.

- 2 **복원 관련 옵션 탭에서 특정 옵션 설정:** DBA가 다음 옵션을 설정합니다.

- **PIT 복구 수행:** 이 형태의 복원 및 모든 관련 옵션을 사용하도록 선택합니다.
- **바이너리 로그 복원 및 적용(시간 또는 위치를 이미 알고 있을 때 사용):** 백업에 포함된 바이너리 로그를 사용하도록 플러그인에 지시하기 위해 선택합니다.
- **현재 바이너리 로그 포함:** 현재 바이너리 로그를 사용하여 수요일 밤의 차등 백업 *이후*에 발생한 모든 데이터베이스 트랜잭션에 적용하도록 NetVault Backup에 지시하기 위해 선택합니다. 이 단계는 수요일 밤의 차등 백업 완료 시간과 Drop Table 명령이 실행된 시간 사이에 발생한 모든 트랜잭션을 복구합니다.
- **오류/잘못된 SQL 문 이전으로 복구 활성화:** 이 옵션을 선택하고, **중지 위치**를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 *앞*에 있는 현재 바이너리 로그의 위치인 **"804"**로 설정하십시오. **중지 위치가 포함된 바이너리 로그를 다른 파일로 설정**하고 텍스트 상자에 현재 바이너리 파일 이름 (예: **"MYSQLSVR-bin.000009"**)을 입력합니다.
- **오류/잘못된 SQL 문 이후로 복구 활성화:** 이 옵션을 선택하고, **시작 위치**를, **mysqlbinlog**를 사용하여 식별된 Drop Table 명령 위치 *뒤*에 있는 현재 바이너리 로그의 위치인 **"806"**으로 설정하십시오. **중지 위치가 포함된 바이너리 로그를 다른 파일로 설정**하고 텍스트 상자에 현재 바이너리 파일 이름 (예: **"MYSQLSVR-bin.000009"**)을 입력합니다. 마지막으로, 현재 바이너리 로그 마지막까지 복구를 수행하기 때문에 **중지 위치**에 대해 **없음** 옵션이 선택되었습니다.

**i | 중요:** 중지 및 시작 위치는 원치 않는 트랜잭션의 위치보다 큰 임의의 숫자가 아니라 바이너리 로그에 나열된 *실제 위치*여야 합니다.

## MIXED 바이너리 로깅 형식이 사용되고 데이터베이스 간 업데이트가 실행되는 경우의 PIT 복원

**i | 중요:** 사용자 사이트에서 **MIXED** 바이너리 로깅 형식을 사용하며 모든 데이터베이스 사용자와 프로그램이 수정된 테이블이 **USE**에서 선택된 데이터베이스에 있고, 데이터베이스 간 업데이트가 실행되지 않도록 하는 최상의 방법을 따르는 경우 이 항목은 사용자 사이트에 적용되지 않습니다. (자세한 내용은 **MIXED 바이너리 로깅 형식 사용**을 참조하십시오.) PIT 복원 작업을 실행할 수 있으며, 작업 중 선택한 데이터베이스의 지정된 지점으로 바이너리 로그를 재생할 수 있습니다.

이전에 설명한 대로, 사용자 환경의 사용자와 프로그램이 **USE**에서 선택되지 않은 데이터베이스의 테이블을 수정하고 데이터베이스 간 업데이트를 실행하면 PIT 복원 작업을 실행할 때 트랜잭션이 지정된 지점으로 재생되지 않을 수 있습니다. Quest의 권장 사항에 따라 모든 데이터베이스 사용자와 프로그램에서 수정된 테이블이 **USE**에서 선택된 데이터베이스에 있도록 합니다. 또한 데이터베이스 간 업데이트가 실행되지 않도록 하는 것이 좋습니다. 이 지침이 사용자 환경에 적합하지 않을 경우 Quest에서는 **MIXED** 바이너리 로깅 형식을 사용하지 않도록 권장합니다.

**i | 중요:** 다음 절차에서는 **"--database"** 옵션 없이 **mysqlbinlog**를 사용합니다. 따라서 바이너리 로그의 모든 내용을 적용하고 모든 데이터베이스를 수정할 수 있습니다. 대체 MySQL Server에 이 절차를 적용하고 대체 MySQL Server에서 해당 데이터를 추출해 보십시오. 프로덕션 MySQL Server에 다음 절차를 적용할 경우 모든 데이터베이스가 지정된 지점으로 롤백됩니다. 모든 MySQL Server 데이터베이스를 지정된 지점으로 롤백하려는 경우를 제외하고 프로덕션 환경에 이 절차를 적용하지 마십시오.

- 1 탐색 창에서 **복원 작업 생성**을 클릭합니다.
- 2 **복원 작업 생성 - 저장 집합 선택** 페이지에서 **테이블 필터링**을 클릭하고 **필터 편집**을 선택합니다.
- 3 **플러그인 유형** 목록에서 **MySQL용 플러그인**-을 선택합니다.
- 4 저장 집합 테이블에서 바이너리 로그에 증분 또는 차등 백업이 포함된 저장 집합을 선택하고 **다음**을 클릭합니다.
- 5 **선택 집합 만들기** 페이지에서 **바이너리 로그**를 선택합니다.  
바이너리 로그는 모든 MySQL Server 데이터베이스에 공통입니다.
- 6 **선택 집합 만들기** 페이지에서 **플러그인 옵션 편집**을 클릭합니다.
- 7 **특정 시점 복구** 탭에서 **PIT 복구 수행 및 시간 또는 위치를 식별하기 위해 로그를 임시 디렉터리로 복원 옵션**을 선택합니다.  
바이너리 로그는 다음 위치에 있는 임시 디렉터리에 복원됩니다.  
**<NetVaultBackupInstallationDirectory>/tmp/mysql/<savesetName>**
- 8 **mysqlbinlog** 명령 프롬프트에서 바이너리 로그를 수동으로 적용하려면 다음을 입력합니다.  

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

  
예:  

```
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 복원 시퀀스에 복원해야 하는 증분 백업이 두 개 이상 포함되어 있는 경우 각 증분 백업에 대해 이 절차를 반복합니다.  
서로 다른 증분 백업 저장 집합은 **<NetVaultBackupInstallationDirectory>/tmp/mysql** 디렉터리의 다른 하위 디렉터리에 복원됩니다. 그런 다음 각 디렉터리에 **mysqlbinlog** 명령을 적용하거나 모든 바이너리 로그를 일반 디렉터리로 복사하거나 이동하여 **mysqlbinlog**를 실행할 수 있습니다.

## MySQL Enterprise Backup 복원 시나리오 예시

실패 또는 데이터 손상에서 복구하려면 복원을 위해 선택한 데이터 및 **옵션** 탭에서 사용 가능한 옵션과 관련하여 작업을 설정할 때 다양한 설정을 수행해야 합니다.

- 전체 백업 전용 복원 시나리오
- 전체 및 증분 백업 복원 시나리오
- TTS 전용 복원 시나리오
- Linux 및 UNIX 환경을 위한 추가 단계

### 전체 백업 전용 복원 시나리오

- 1 복원할 준비가 완료된 전체 백업을 생성하려면 **옵션** 탭에서 **복원, 원시 전체 백업 추출...** 옵션을 선택한 작업을 제출합니다.
- 2 MySQL을 종료하고 MySQL Server 리포지토리에 준비된 전체 백업을 복사하려면 **옵션** 탭에서 **MySQL Server 종료 및 다시 복사...** 옵션을 선택한 작업을 제출합니다.
- 3 명령 프롬프트에서 해당 명령을 입력하여 MySQL Server를 다시 시작합니다.

## 전체 및 증분 백업 복원 시나리오

- 1 복원할 준비가 완료된 전체 백업을 생성하려면 **옵션** 탭에서 **복원, 원시 전체 백업 추출...** 옵션을 선택한 작업을 제출합니다.
- 2 백업된 순서에 따라 필요한 증분 백업을 준비된 전체 백업에 적용하려면 **옵션** 탭의 **복원, 원시 전체 백업 추출...** 옵션을 선택한 작업을 해당 개수만큼 제출합니다.
- 3 MySQL을 종료하고 MySQL Server 리포지토리에 준비된 전체 백업을 복사하려면 **옵션** 탭에서 **MySQL Server 종료 및 다시 복사...** 옵션을 선택한 작업을 제출합니다.
- 4 명령 프롬프트에서 해당 명령을 입력하여 MySQL Server를 다시 시작합니다.

## TTS 전용 복원 시나리오

- 1 복원할 준비가 완료된 전체 백업을 생성하려면 **옵션** 탭에서 **복원, 원시 전체 백업 추출...** 옵션을 선택한 작업을 제출합니다.
- 2 준비된 전체 백업을 MySQL Server 리포지토리에 복사하고 하나의 테이블 이름을 바꾸려면 다음을 수행한 작업을 제출합니다.
  - **옵션** 탭에서 **MySQL Server 종료 및 다시 복사...** 옵션을 선택합니다.
  - **테이블 포함** 필드에 정규식 패턴을 입력하여 복원에 포함하려는 테이블을 나타냅니다.  
테이블 포함 필드의 예: `database_name\`
  - **테이블 이름 바꾸기** 필드에 이름 바꾸기 요청을 입력합니다.  
테이블 이름 바꾸기 필드의 예: `original_name to new_name`

## Linux 및 UNIX 환경을 위한 추가 단계

Linux 또는 UNIX 환경에서 MySQL Enterprise Backup을 사용하는 경우 복원된 데이터에 대한 파일 소유권 및 권한 정보가 데이터 백업 전의 정보와 일치하는지 확인하십시오. **mysqlbackup** 스크립트에서 백업 프로세스 중에 이 정보를 기록하지 않으므로 복원이 완료된 후에 해당 정보가 다를 수 있습니다. 자세한 내용은 [https://docs.oracle.com/cd/E17952\\_01/mysql-enterprise-backup-3.11-en/bugs.backup.html](https://docs.oracle.com/cd/E17952_01/mysql-enterprise-backup-3.11-en/bugs.backup.html) 페이지를 참조하십시오.

# MySQL Standard/Community에 대한 고급 복원 절차 사용

이 항목에서는 **MySQL Standard/Community** 옵션용 플러그인으로 수행할 수 있는 다른 복원 작업에 대해 설명합니다.

- 복원 중에 데이터베이스 이름 바꾸기
- 동일한 서버에서 다른 MySQL 인스턴스로 복원
- 대체 MySQL Server로 복구

# 복원 중에 데이터베이스 이름 바꾸기

NetVault Backup에서는 백업된 MySQL 데이터베이스를 선택하고 해당 데이터베이스의 기존 버전을 덮어쓰지 않도록 복원 시 이름을 바꿀 수 있습니다. 이 프로세스는 기존 데이터베이스의 복사본을 생성할 때 유용할 수 있습니다. 이 프로세스를 완료하려면 다음 항목에 설명된 단계를 수행하십시오.

**중요:** 복원을 위해 전체 데이터베이스만 이름을 바꿀 수 있습니다. 개별 테이블의 이름을 바꾸면 오류 메시지가 표시됩니다.

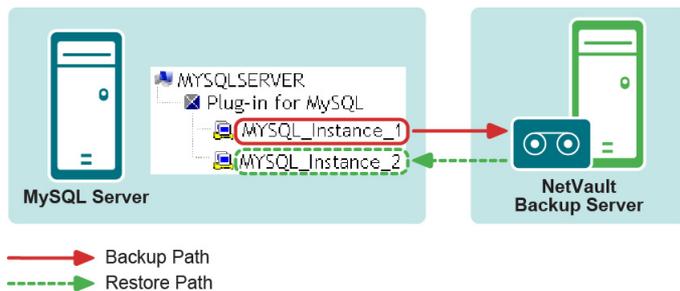
복원 이름을 바꾸기 전에 이 작업에 대한 알려진 제한 사항 및 용도를 검토하십시오.

- 유효한 복원 시퀀스는 전체 또는 개별 데이터베이스/테이블 복사 전용 백업으로 제한됩니다.
  - 증분 및 차등 복원 중에는 허용되지 않습니다.
  - 다른 MySQL 인스턴스 또는 MySQL Server에 대한 복원과 함께 사용할 수 있습니다.
- 1 탐색 창에서 **복원 작업 생성**을 클릭하고 **플러그인 유형** 목록에서 **MySQL용** 플러그인-를 선택하고 적용 가능한 저장 집합을 선택한 후 **다음**을 클릭합니다.  
자세한 내용은 **복원 데이터 선택**(를) 참조하십시오.
  - 2 **선택 집합 만들기** 페이지에서 이름을 바꾸려는 데이터베이스를 클릭하고 상황에 맞는 메뉴에서 **이름 바꾸기**를 선택합니다.
  - 3 **이름 바꾸기/재배치** 대화 상자에서 **이름 바꾸기** 상자에 새 이름을 입력하고 **확인**을 클릭합니다.  
데이터베이스 항목에서 괄호 안에 이름 변경 정보가 함께 표시됩니다.
  - 4 **MySQL에서 데이터 복원**에 설명된 대로 복원 절차를 계속합니다.

# 동일한 서버에서 다른 MySQL 인스턴스로 복원

이 형태의 재배치 복원에서는 MySQL용 플러그인- 백업을 **동일한** MySQL Server 시스템에 복원하는 것이 아니라 구성된 MySQL의 **다른 인스턴스**로 복원해야 합니다.

그림 1. 하나의 MySQL 인스턴스에 백업된 데이터를 다른 인스턴스로 복구



이 프로세스를 완료하려면 다음 항목에 설명된 단계를 수행하십시오.

## 알려진 제한 사항 및 용도

재배치 복원을 수행하기 전에 이 작업에 대한 알려진 제한 사항 및 용도를 검토하십시오.

- 유효한 복원 시퀀스에는 전체, 증분, 차등 및 개별 데이터베이스/테이블 복사 전용 백업이 포함될 수 있습니다.
- 증분 또는 차등 백업에서 복원된 바이너리 로그만 대상 인스턴스에 적용할 수 있습니다. 즉, 소스 인스턴스의 현재 바이너리 로그는 대상 인스턴스에 적용할 수 없습니다.

## 필수 구성 요소

이 유형의 복원을 설정하여 실행하기 전에 다음 전제 조건을 충족해야 합니다.

- **기존 및 대상 시스템의 설치 구성이 동일해야 함:** 두 시스템 모두 MySQL과 관련하여 다음과 같은 설정이 필요합니다.
  - **동일한 OS 설치**
  - **동일한 버전의 MySQL 설치**
- **새 대상 인스턴스가 MySQL용 플러그인-에 성공적으로 구성되어야 합니다.** 플러그인 구성에 설명된 프로세스를 성공적으로 수행하여 새 MySQL 인스턴스를 추가해야 합니다. 즉, 대상 인스턴스는 반드시 표시되어야 하며 **NetVault Backup 선택** 페이지의 **MySQL용 플러그인-** 노드 내에서 액세스할 수 있어야 합니다.

## 복원 설정 및 시작

전제 조건을 충족하는 경우 다음 단계를 수행하여 이 재배치 복원 작업을 설정합니다.

- 1 탐색 창에서 **복원 작업 생성**을 클릭합니다.
- 2 **복원 작업 생성 - 저장 집합 선택** 페이지의 **플러그인 유형** 목록에서 **MySQL용 플러그인-**을 선택합니다.
- 3 저장 집합 테이블에 표시된 항목을 추가로 필터링하려면 **클라이언트**, **날짜** 및 **작업 ID** 목록을 사용하십시오. 테이블에는 저장 집합 이름(작업 이름 및 저장 집합 ID), 만든 날짜 및 시간, 크기가 표시됩니다. 기본적으로 목록은 만든 날짜별로 정렬됩니다.
- 4 저장 집합 테이블에서 해당 항목을 선택합니다.  
저장 집합을 선택하면 다음 세부 정보가 **저장 집합 정보** 영역에 표시됩니다. 작업 ID, 작업 제목, 서버 이름, 클라이언트 이름, 플러그인 이름, 저장 집합 날짜 및 시간, 만료 설정, 증분 백업 여부, 아카이브 여부, 저장 집합 크기, 스냅샷 기반 백업 여부.
- 5 **다음**을 클릭합니다.
- 6 **선택 집합 만들기** 페이지에서 복원할 데이터를 선택합니다.  
백업 대상인 개별 MySQL 인스턴스를 표시하고, 해당 데이터 항목을 찾을 때까지 선택 트리를 탐색하고, 포함할 항목을 선택합니다.
- 7 해당 데이터베이스를 선택한 상태에서 **플러그인 옵션 편집**을 클릭한 다음 **복원 대상** 탭을 클릭합니다.
- 8 **복원 대상 세부 정보** 섹션에서 다음을 입력합니다.
  - **사용자 이름:** 대상 MySQL 인스턴스에 액세스하는 데 사용되는 로그인 계정 이름을 입력합니다.
  - **암호:** 로그인 계정과 연관된 암호를 입력합니다.
  - **인스턴스 이름:** 구성 중에 설정한 항목을 기반으로 MySQL의 새 인스턴스 이름으로 설정한 NetVault Backup 이름을 NetVault Backup에 입력합니다. 이 이름은 구성 대화 상자에서 **MySQL 인스턴스 이름**으로 설정된 이름입니다. 자세한 내용은 **플러그인 구성**을 참조하십시오.
- 9 해당하는 경우 **특정 시점 복구** 탭에서 해당하는 옵션을 선택합니다.  
이 형태의 복원을 수행할 때 이러한 옵션은 필요하지 않습니다. 자세한 내용은 **복원 옵션 설정** 항목을 참조하십시오.
- 10 설정을 저장하려면 **확인**을 클릭 한 후 **다음**을 클릭합니다.
- 11 기본 설정을 사용하지 않으려는 경우 **작업 이름**에 작업의 이름을 지정합니다.  
진행 상황을 모니터링할 때 작업을 쉽게 식별할 수 있도록 설명 이름을 지정합니다. 작업 이름에는 영숫자 및 영숫자가 아닌 문자가 포함될 수 있지만 라틴 문자가 아닌 문자는 포함될 수 없습니다. Linux에서 이름의 길이는 최대 200자입니다. Windows에서는 길이 제한이 없습니다. 그러나 모든 플랫폼에서 최대 40자를 권장합니다.

**i | 중요:** 대상 OS의 파일 이름에 지원되지 않는 특수 문자를 사용하지 마십시오. 예를 들어 /, \, \*, @ 문자는 Windows에서 사용할 수 없습니다. 이 요구 사항은 **MySQL용 플래그인-이** 데이터를 임시로 복원하기 위해 작업 제목과 동일한 이름의 폴더를 생성하려고 하기 때문입니다.

12 대상 클라이언트 목록에서 데이터를 복원할 시스템을 선택합니다.

**i | 팁:** 선택을 클릭한 다음 **대상 클라이언트 선택** 대화 상자에서 해당 클라이언트를 찾아 선택합니다.

13 일정, 소스 옵션 및 고급 옵션 목록을 사용하여 필요한 추가 옵션을 구성합니다.

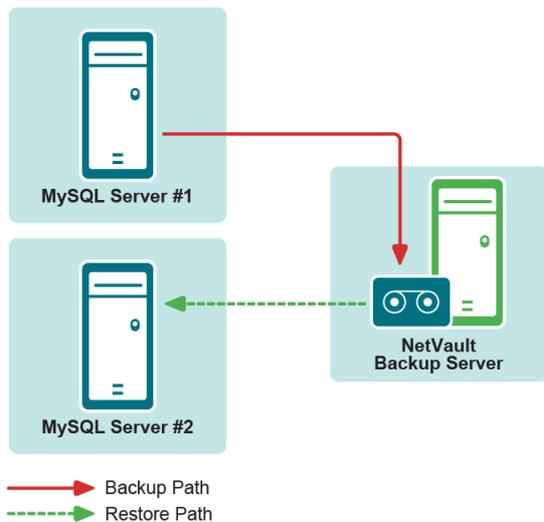
14 해당되는 경우 **저장** 또는 **저장 및 제출**을 클릭합니다.

**작업 상태** 페이지에서 진행률을 모니터링하고 **로그 보기** 페이지에서 로그를 볼 수 있습니다. 자세한 내용은 *Quest NetVault Backup 관리자 안내서*를 참조하십시오.

## 대체 MySQL Server로 복구

데이터베이스 또는 개별 테이블을 같은 MySQL Server의 다른 MySQL 인스턴스로 복원할 수 있는 플래그인의 기능과 마찬가지로 복원 프로세스 중에 **다른 SQL Server**를 대상으로 지정할 수 있습니다. 이 옵션은 재해 복구 작업 중에 사용됩니다.

그림 2. 이 재배치 복원 형태의 데이터 경로 예시



이 프로세스를 완료하려면 다음 항목에 설명된 단계를 수행하십시오.

## 알려진 제한 사항 및 용도

다른 MySQL Server에 대한 재배치 복원을 수행하기 전에 이 작업에 대한 알려진 제한 사항 및 용도를 검토하십시오.

- 유효한 복원 시퀀스에는 전체, 증분, 차등 및 개별 데이터베이스/테이블 복사 전용 백업이 포함될 수 있습니다.
- 증분 또는 차등 백업에서 복원된 바이너리 로그만 대상 인스턴스에 적용할 수 있습니다. 즉, 소스 인스턴스의 현재 바이너리 로그는 대상 MySQL 인스턴스에 적용할 수 없습니다.

# 소프트웨어 설치 및 구성 전제 조건

이 유형의 복원을 설정하여 실행하기 전에 다음 전제 조건을 충족해야 합니다.

- **기존 및 대상 시스템의 설치 구성이 동일해야 함:** 두 시스템 모두 MySQL과 관련하여 다음과 같은 설정이 필요합니다.
  - 동일한 OS 설치
  - 동일한 버전의 MySQL 설치
  - 동일한 설치 및 기본 디렉터리
  - 동일한 MySQL "데이터" 디렉토리
- 모든 클라이언트에 설치된 **NetVault Backup 소프트웨어 및 MySQL용 플러그인**:- NetVault Backup, 클라이언트나 서버 버전, 플러그인이 이 프로세스에 사용되는 **두 시스템**(즉, **기존 MySQL 시스템** 및 **새 복원 대상**)에 모두 설치되고 구성되어야 합니다.
- **NetVault Backup Server에 추가된 모든 클라이언트 시스템:** 모든 소프트웨어 설치 요구 사항을 충족하는 경우 대상 NetVault Backup 클라이언트 시스템(즉, **기존 MySQL 시스템**과 **새 복구 대상**)을 NetVault Backup WebUI를 통해 NetVault Backup Server에 추가해야 합니다.
- **MySQL 인스턴스가 새 복원 대상에 있어야 함:** 재배치 프로세스를 수행하려면 MySQL 인스턴스가 **새 복원 대상**에 있어야 합니다. 이 인스턴스는 재배치 복원의 대상 역할을 합니다. 이 인스턴스는 MySQL에서 올바르게 설정 및 구성되어 있어야 하며 새 복원 대상에 있는 플러그인에 추가되어야 합니다. **플러그인 구성**에 나와 있는 단계를 따르십시오.

**i** **중요:** 새 복원 대상에 있는 인스턴스의 구성 대화 상자에서 다음 값을 확인합니다: **사용자 이름, 암호 및 인스턴스 이름**

재배치 복원을 설정하는 동안 플러그인에서는 대상 MySQL 인스턴스에 대한 적절한 액세스를 확보하기 위해 **옵션** 탭에 이러한 값을 입력해야 합니다.

## 복원 실행

전제 조건을 충족하는 경우 다음 단계를 수행하여 MySQL 백업을 다른 시스템에 복원합니다.

- 1 탐색 창에서 **복원 작업 생성**을 클릭합니다.
- 2 **복원 작업 생성 - 저장 집합 선택** 페이지의 **플러그인 유형** 목록에서 **MySQL용** 플러그인-을 선택합니다.
- 3 저장 집합 테이블에 표시된 항목을 추가로 필터링하려면 **클라이언트, 날짜** 및 **작업 ID** 목록을 사용하십시오. 테이블에는 저장 집합 이름(작업 이름 및 저장 집합 ID), 만든 날짜 및 시간, 크기가 표시됩니다. 기본적으로 목록은 만든 날짜별로 정렬됩니다.
- 4 저장 집합 테이블에서 해당 항목을 선택합니다. 저장 집합을 선택하면 다음 세부 정보가 **저장 집합 정보** 영역에 표시됩니다. 작업 ID, 작업 이름, 서버 이름, 클라이언트 이름, 플러그인 이름, 저장 집합 날짜 및 시간, 만료 설정, 증분 백업 여부, 아카이브 여부, 저장 집합 크기, 그리고 스냅샷 기반 백업 여부.
- 5 **다음**을 클릭합니다.
- 6 **선택 집합 만들기** 페이지에서 복원할 데이터를 선택합니다. 백업 대상인 개별 MySQL 인스턴스를 표시하고, 해당 데이터 항목을 찾을 때까지 선택 트리를 탐색하고, 포함할 항목을 선택합니다.
- 7 해당 데이터베이스를 선택한 상태에서 **플러그인 옵션 편집**을 클릭한 다음 **복원 대상** 탭을 클릭합니다.

8 **복원 대상 세부 정보** 섹션에서 다음을 입력합니다.

- **사용자 이름:** **새 복원 대상**에서 대상 인스턴스에 대해 설정된 사용자 이름(즉, 구성 대화 상자의 **사용자 이름** 필드에 설정된 이름)을 입력합니다.
- **암호:** **새 복원 대상**의 대상 인스턴스에 대해 설정된 암호를 입력합니다.
- **인스턴스 이름:** **새 복원 대상**에 있는 MySQL의 대상 인스턴스에 대해 설정된 NetVault Backup 이름을 입력합니다.

9 해당하는 경우 **특정 시점 복구** 탭에서 해당하는 옵션을 선택합니다.

이 형태의 복원을 수행할 때 이러한 옵션은 필요하지 않습니다. 자세한 내용은 **복원 옵션 설정** 항목을 참조하십시오.

10 설정을 저장하려면 **확인**을 클릭 한 후 **다음**을 클릭합니다.

11 기본 설정을 사용하지 않으려는 경우 **작업 이름**에 작업의 이름을 지정합니다.

진행 상황을 모니터링할 때 작업을 쉽게 식별할 수 있도록 설명 이름을 지정합니다. 작업 이름에는 영숫자 및 영숫자가 아닌 문자가 포함될 수 있지만 라틴 문자가 아닌 문자는 포함될 수 없습니다. Linux에서 이름의 길이는 최대 200자입니다. Windows에서는 길이 제한이 없습니다. 그러나 모든 플랫폼에서 최대 40자를 권장합니다.

- **중요:** 대상 OS의 파일 이름에 지원되지 않는 특수 문자를 사용하지 마십시오. 예를 들어 /, \, \*, @ 문자는 Windows에서 사용할 수 없습니다. 이 요구 사항은 *MySQL용* 플러그인-이 데이터를 임시로 복원하기 위해 작업 제목과 동일한 이름의 폴더를 생성하려고 하기 때문입니다.

12 **대상 클라이언트** 목록에서 데이터를 복원할 시스템을 선택합니다.

- **팁:** **선택**을 클릭한 다음 **대상 클라이언트 선택** 대화 상자에서 해당 클라이언트를 찾아 선택합니다.

13 **일정**, **소스 옵션** 및 **고급 옵션** 목록을 사용하여 필요한 추가 옵션을 구성합니다.

14 해당되는 경우 **저장** 또는 **저장 및 제출**을 클릭합니다.

**작업 상태** 페이지에서 진행률을 모니터링하고 **로그 보기** 페이지에서 로그를 볼 수 있습니다. 자세한 내용은 *Quest NetVault Backup 관리자 안내서*를 참조하십시오.

# 기본 MySQL 복제 작업

- 기본 환경에서 플러그인 사용: 개요
- 복제 지원 활성화
- 복제 서버 백업
- 복제 서버 복원

## 기본 환경에서 플러그인 사용: 개요

복제를 사용하는 경우 복제된 테이블에 대한 모든 업데이트를 마스터 서버에서 수행해야 합니다. 그렇지 않으면 사용자가 마스터의 테이블에 수행한 업데이트와 슬레이브의 테이블에 수행한 업데이트 간의 충돌을 방지해야 합니다.

복제를 통해 견고성, 속도 및 시스템 관리에 따른 이점을 얻을 수 있습니다.

- 마스터 및 슬레이브 설정으로 견고성이 향상되었습니다. 마스터에서 문제가 발생하면 백업으로 슬레이브로 전환할 수 있습니다.
- 마스터 서버와 슬레이브 서버 간에 클라이언트 쿼리를 처리하기 위한 부하를 분할하여 클라이언트의 응답 시간을 개선할 수 있습니다. 마스터의 쿼리 처리 부하를 줄이기 위해 슬레이브로 일부 쿼리를 보낼 수 있습니다. 데이터를 수정하는 문도 마스터와 슬레이브가 동기화되지 않도록 마스터에 전송되어야 합니다. 이 부하 분산 전략은 업데이트되지 않은 쿼리가 대부분인 경우에 효과적이며, 이는 일반적인 경우입니다.
- 복제 사용에 따른 추가적인 이점은 마스터 서버를 중단하지 않고 슬레이브 서버를 사용하여 데이터베이스 백업을 수행할 수 있다는 것입니다. 마스터는 백업이 수행되는 동안 업데이트를 계속 처리합니다.

MySQL용 플러그인-은 단일 마스터 복제 환경의 백업 및 복구를 지원합니다.

## 복제 지원 활성화

구성 대화 상자를 사용하여 복제 지원을 활성화합니다. 이 대화 상자에 액세스하는 방법에 대한 자세한 내용은 [플러그인 구성](#)을 참조하십시오.

- **MySQL 복제 활성화:** 이 인스턴스에 기본 MySQL 복제가 활성화되어 있으면 이 확인란을 선택합니다.
  - **슬레이브 인스턴스:** 인스턴스가 *슬레이브*로 구성된 경우 이 옵션을 선택합니다.
  - **마스터 인스턴스:** 인스턴스가 *마스터*로 구성된 경우 이 옵션을 선택합니다.
- **특정 시점 복구 활성화:** PIT 백업 및 복원을 사용하려면 이 확인란을 선택합니다.
- **바이너리 로그 인덱스 경로:** 특정 시점 복구 활성화 확인란을 선택한 경우 이 필드를 사용하여 바이너리 로그 인덱스 파일의 전체 경로를 지정합니다.
- **릴레이 로그 인덱스 경로:** 슬레이브 인스턴스를 구성하는 경우, 릴레이 로그 인덱스 파일에 대한 전체 경로를 입력하여 백업에 포함시킵니다.

# 복제 서버 백업

기본 MySQL 복제 환경 백업 지원에 대한 제한 사항은 다음과 같습니다.

- **슬레이브 복제 서버:** 지원되는 백업 유형은 다음과 같습니다.
  - 전체
  - 증분
  - 차등
  - 개별 데이터베이스/테이블 복사 전용
- **마스터 복제 서버:** 지원되는 백업 유형은 다음과 같습니다.
  - 개별 데이터베이스/테이블 복사 전용

슬레이브 서버에서 증분 및 차등 백업을 사용하려면 MySQL에서 **"-log-slave-update"** 옵션을 활성화해야 합니다. 이 옵션은 SQL 스레드가 자체 바이너리 로그에 수행한 업데이트를 기록하도록 슬레이브에 지시합니다. 이 옵션이 작동하려면, 슬레이브는 **"--log-bin"** 옵션으로 시작되어 바이너리 로그를 활성화해야 합니다. 일반적으로 이 옵션은 복제 서버를 연결하는 데 사용됩니다. 하지만 슬레이브에 적용되기 전에 마스터 서버에서 바이너리 로그를 삭제하는 복잡한 작업 없이도 복제된 환경의 PIT 복구를 활성화하는 바이너리 로그 백업에도 사용할 수 있습니다.

## 복제 구성 백업

**릴레이 로그 인덱스 경로** 옵션을 사용하여 릴레이 로그 인덱스 파일의 전체 경로 이름을 지정하여 백업에 포함할 수 있습니다. 기본적으로 상태 파일, **"master.info"** 및 **"relay-log.info"**는 같은 위치에 있습니다. 릴레이 로그 인덱스 경로 옵션을 사용하고 기본 파일 이름 및 위치가 보존되는 경우 플러그인은 슬레이브 복제 서버에 대해 이러한 모든 파일을 자동으로 백업 및 복원합니다.

## 복제 서버 복원

MySQL 복제 슬레이브 인스턴스의 전체, 증분 및 차등 백업을 사용하여 MySQL 복제 마스터 인스턴스에 대한 재해 복구를 수행할 수 있습니다. 마스터 인스턴스가 복원된 후에는 동일한 백업 세트를 사용하여 각 슬레이브 인스턴스를 마스터 인스턴스와 동일한 수준으로 복원한 다음 복제를 다시 시작하거나 *MySQL 참조 안내서*에 제공된 다른 초기화 방법을 사용하여 슬레이브 인스턴스를 다시 초기화할 수 있습니다.

마스터 및 슬레이브 모두에서 개별 데이터베이스/테이블 백업을 사용하여 개별 데이터베이스 및 테이블을 마스터로 복원할 수 있습니다. 슬레이브의 개별 테이블 또는 데이터베이스를 다시 동기화하려는 경우 Quest에서는 슬레이브로 복원한 다음 슬레이브를 마스터와 동기화하는 대신 MySQL의 복제 프로세스를 사용해 다시 동기화할 것을 권장합니다.

# 장애 조치 클러스터 환경에서 플러그인 사용

- MySQL Server 장애 조치 클러스터링: 개요
- 플러그인 설치 또는 업그레이드
- 플러그인 구성
- 데이터 백업
- 데이터 복원

## MySQL Server 장애 조치 클러스터링: 개요

MySQL 장애 조치 클러스터링(활성/수동)에서는 전체 MySQL Server 인스턴스에 대한 고가용성을 제공합니다. 예를 들어 하드웨어 장애, OS 장애 또는 계획된 업그레이드 중에 장애 조치 클러스터의 한 노드에 있는 MySQL Server 인스턴스를 클러스터의 다른 노드로 장애 조치하도록 구성할 수 있습니다.

장애 조치 클러스터는 두 개 이상의 공유 디스크가 있는 하나 이상의 노드(호스트) 조합입니다. IP, 공유 저장소 및 응용 프로그램과 같은 노드가 호스팅하는 다양한 리소스를 그룹화(이 경우는 MySQL)하여 클러스터링 서비스를 생성할 수 있습니다. 가상 서비스는 응용 프로그램을 실행하는 단일 컴퓨터처럼 네트워크에 표시되지만 현재 노드를 사용할 수 없게 되면 한 노드에서 다른 노드로의 장애 조치를 제공합니다.

MySQL 용 플러그인-은 MySQL Server 장애 조치 클러스터링을 지원합니다. 플러그인은 장애 조치 클러스터 네트워크 이름을 사용하여 MySQL Server 클러스터링 서비스를 제어하는 현재 노드를 확인하고 백업 대상으로 만들 수 있습니다.

이 항목에서는 장애 조치 클러스터 환경과 기존 환경에서 플러그인의 설치 및 사용의 차이점에 대해 설명합니다. **MySQL Standard/Community** 옵션에 대한 지침에 나와 있는 항목을 나타냅니다.

### 중요 고려 사항

- 클러스터링 데이터의 플러그인으로 수행된 백업 및 복원은 해당 항목에 설명된 경우를 제외하고, 기존 MySQL Server 데이터로 수행한 백업 및 복원과 동일합니다.
- 다음 항목에서는 장애 조치 클러스터 환경에서 이 플러그인을 사용하는 데 필요한 MySQL 관련 설정에 대한 정보만 제공합니다. MySQL Server와 관련 없는 데이터 및 파일의 백업 및 복원을 관리하기 위해 NetVault Backup의 **응용 프로그램 클러스터 지원**을 설정하는 방법에 관한 지침은 제공하지 않습니다. 이 프로세스는 플러그인과 관련이 없으며 *Quest NetVault Backup 관리자 안내서*에서 자세한 내용을 찾을 수 있습니다.
- 계속하기 전에 *Quest NetVault Backup 관리자 안내서*에 제공된 모든 클러스터 관련 정보를 검토하여 다음 정보가 MySQL Server 장애 조치 클러스터 기능에서 어떻게 작동하는지 확인하십시오.

# 플러그인 설치 또는 업그레이드

플러그인을 설치하려면 다음 항목에 설명된 단계를 완료합니다.

## 설치 필수 구성 요소

클러스터링 환경에 MySQL용 플러그인-을 설치하기 전에 다음 필수 구성 요소를 충족해야 합니다.

- **MySQL 장애 조치 클러스터링 환경 설치:** 올바르게 구성된 MySQL 클러스터 환경이 있어야 합니다.
  - **중요:** 이 기능에 대한 지원은 Red Hat Enterprise Linux(RHEL) v5.x에서 Red Hat Clustering 및 Clustered Storage Suite를 사용하고, 데이터베이스 데이터 파일 및 로그를 포함하는 공유 저장소에 2개 노드 MySQL(v5.5) 클러스터 구성을 활용하여 테스트되었습니다. 다른 구성에서 클러스터링을 사용하려면 프로덕션 환경에 구축하기 전에 백업 및 복원을 테스트하십시오.
- **별도의 NetVault Backup 서버 시스템:** NetVault Backup 서버 역할을 할 시스템이 올바르게 구성되어 있어야 합니다. 이 시스템은 MySQL Server 클러스터 **외부에 있어야 하지만** 클러스터 내에 있는 노드(호스트)에 대한 네트워크 연결이 있어야 합니다.

## 소프트웨어 설치

클러스터링 환경용 플러그인 설치 는 이 플러그인의 기존 설치와 동일합니다. 자세한 내용은 [플러그인 설치 및 제거](#) 를 참조하십시오.

## 플러그인 구성

기본 노드에서 다음 단계를 수행합니다.

- 1 NetVault Backup Server에 있는 NetVault Backup WebUI 탐색 창에서 **백업 작업 생성**을 클릭하고 **선택** 목록 옆에 있는 **새로 만들기**를 클릭합니다.
- 2 선택 트리에서 기본 노드를 엽니다.
- 3 **MySQL용 플러그인-**을 엽니다.
- 4 **모든 인스턴스** 노드를 클릭하고 상황에 맞는 메뉴에서 **구성**을 선택합니다.
- 5 **구성** 대화 상자에서 해당 구성 옵션을 설정합니다.

사용 가능한 옵션은 [플러그인 구성](#)에 설명된 옵션과 동일합니다.

- **중요:** 구성 대화 상자의 **인스턴스** 필드에 모든 클러스터 인스턴스를 추가합니다. 인스턴스를 추가하기 위해 MySQL 클러스터링 서비스 이름을 VIRTUAL SERVER NAME\INSTANCE NAME으로 지정합니다.
- 6 보조 노드에서 기존 백업 작업을 수정하거나 추가로 백업 작업을 생성해야 할 것으로 예상되면 다음 단계를 수행합니다.
    - a 기본 노드를 보조 노드로 장애 조치합니다.
    - b **단계 1~단계 5**를 반복합니다.
    - c 기본 노드로 장애 복구를 수행합니다.
  - 7 설정을 저장하려면 **확인**을 클릭합니다.

# 데이터 백업

**NetVault Backup 선택** 페이지에서 *MySQL* 용 플러그인- 노드를 열고 *MySQL Server Virtual Server* 또는 내부에 포함된 항목을 선택하여 백업에 포함합니다.

이 페이지에 표시되는 인스턴스 이름은 실제로 *MySQL* 클러스터링 서비스입니다. 다른 *MySQL Server* 클러스터링 서비스가 이 노드에서 실행 중인 경우 해당 인스턴스 또한 *MySQL* 용 플러그인- 노드 내에 표시됩니다. 다른 인스턴스의 데이터는 백업에 포함되도록 **선택되지 않아야 합니다**.

- i | 참고:** 데이터를 백업하거나 복원할 때 기본 노드를 사용하여 프로세스를 실행합니다. 노드 중 하나를 열거나 확장하고 계층 구조를 따라 드릴다운하면 *MySQL* 클러스터링 서비스가 표시됩니다. 활성 상태인 노드에 따라 드릴다운하고 항목을 선택할 수 있습니다. 시스템에서 로그 정보를 유지 관리하는 데 이 인스턴스를 사용할 수 있지만 이 수준에서는 어떤 프로세스도 실행하지 마십시오.

# 데이터 복원

*MySQL* 용 플러그인-에서 복원에 사용할 수 있는 모든 옵션은 장애 조치 클러스터링 환경에도 사용할 수 있으며, 데이터 선택은 같은 방식으로 수행됩니다. 유일한 차이점은 복원 가능한 백업이 각 백업 중에 활성화된 기본 노드의 이름 아래에 있는 **복원 작업 생성 - 저장 집합 선택** 페이지에 표시된다는 것입니다. 복원 작업이 시작되면 *NetVault Backup*은 모든 구성원 클라이언트와 통신하여 장애 조치 클러스터를 제어하는 시스템을 확인한 다음에 이 시스템을 복원 대상으로 지정합니다.

복원을 수행하기 위해 제공되는 모든 지침은 장애 조치 클러스터의 복구에 사용할 수 있습니다. 자세한 내용은 **데이터 복원**의 다양한 항목을 확인하십시오. 장애 조치 클러스터를 독립 실행형 *NetVault Backup* 클라이언트로 복원하려면 대체 *MySQL Server*로 복구에 제공된 지침을 활용하십시오.

## 문제 해결

이 항목에서는 몇 가지 일반적인 오류 및 해결 방법을 설명합니다. 오류가 발생하고 이 표에 설명되어 있지 않은 경우 NetVault Backup 로그에서 MySQL 오류 번호를 확인한 다음 관련 MySQL 설명서를 참조하십시오.

표 2. 문제 해결

오류 메시지	설명
<ul style="list-style-type: none"> <li>백업 레코드를 추가하지 못했음</li> <li>백업 인덱스를 데이터베이스에 기록하지 못했습니다.</li> </ul>	<p>이러한 메시지는 선택한 데이터가 백업되었지만 작업 인덱스 정보가 NetVault Backup에서 데이터베이스에 올바르게 추가되지 않았음을 나타냅니다. 이 인덱스 정보가 없으면 데이터를 올바르게 복원할 수 없습니다.</p> <p><b>방법 1:</b></p> <p>NetVault Backup WebUI의 <b>장치 관리</b> 페이지에 액세스하고 작업의 대상이 되는 미디어 스캔을 수행합니다. NetVault Backup에서는 백업 작업에 대한 인덱스 정보를 두 위치(NetVault 데이터베이스 및 백업의 대상이 되는 미디어)에 저장합니다. 이 스캔을 수행하면 인덱스 정보가 NetVault 데이터베이스에 추가됩니다. 정보가 추가되었는지 확인하려면 <b>복원 작업 생성 - 저장 집합 선택</b> 페이지를 열고 특정 작업을 찾습니다. 탐색 작업을 찾아서 복원 작업을 설정하면 스캔 프로세스를 통해 문제를 해결할 수 있습니다.</p> <p><b>방법 2:</b></p> <p>이전 방법이 실패한 경우에는 백업 작업을 다시 실행합니다.</p>
복제 오류와 함께 백업에 실패합니다.	백업이 실패하고 "복제 슬레이브 서버 시작 실패"와 유사한 메시지가 표시되는 경우 <b>MySQL 복제 활성화</b> 확인란을 선택했지만 복제를 구성하지 않은 것일 수 있습니다. 이 문제를 해결하려면 <b>구성</b> 대화 상자에서 <b>MySQL 복제 활성화</b> 확인란을 선택 취소하거나 복제를 설정한 뒤 백업 작업을 다시 실행합니다. 업데이트에 대한 자세한 내용은 <b>플러그인 구성</b> 을 참조하십시오. 또한 복제에 대한 자세한 내용은 <b>기본 MySQL 복제 작업</b> 을 참조하십시오.
Linux 또는 UNIX 환경에서 백업 또는 복원 작업이 실패하고 다음 오류가 표시됩니다.	작업에서 MySQL Server 소켓 파일에 대한 기본 위치 "/tmp/mysql.sock"에 액세스하려 하지만 이 파일은 다른 곳에 있습니다. 파일은 "/var/lib/mysql/mysql.sock"이나 "/opt/mysql/mysql.sock" 또는 다른 위치에 있을 수 있습니다. 이 문제를 해결하려면 다음 명령을 사용하여 해당 작업이 소켓 파일에 액세스할 수 있도록 기호 링크를 생성합니다.
MySQL Server에 연결을 구성할 수 없습니다. "'/tmp/mysql.sock' (2) 소켓을 통해 로컬 MySQL Server에 연결할 수 없음" 오류로 연결 실행에 실패했습니다.	<pre>ln -s &lt;existingFile&gt; &lt;symbolicLinkFile&gt;</pre> <p>경로 및 파일 이름 업데이트에 대한 자세한 내용은 <b>플러그인 구성</b>을 참조하십시오.</p>

Quest는 급변하는 엔터프라이즈 IT 업계에 소프트웨어 솔루션을 제공합니다. 데이터와 클라우드의 확장, 하이브리드 데이터 센터, 보안 위협, 규제 요구 사항 등으로 인해 발생하는 문제들을 간소화해 드립니다. Fortune 500의 95%, Global 1000의 90%에 해당하는 업체를 포함하여 100여 개 국가에 소재한 130,000개 업체에 솔루션을 공급하고 있습니다. 1987년부터 포트폴리오에 데이터베이스 관리, 데이터 보호, ID 및 접근 관리, Microsoft 플랫폼 관리, 통합 끝점 관리 등을 위한 솔루션을 마련했습니다. Quest와 함께라면, IT 관리 시간을 줄이고 비즈니스 혁신에 더 많은 시간을 투자하실 수 있습니다. 자세한 내용은 [www.quest.com](http://www.quest.com)을 참조하십시오.

## 기술 지원 리소스

유효한 유지 관리 계약을 보유한 Quest 고객과 평가판을 보유하고 있는 고객은 기술 지원을 이용할 수 있습니다. <https://support.quest.com>에서 Quest 지원 포털에 액세스하면 됩니다.

지원 포털에서는 1년 365일, 하루 24시간 동안 언제든지 문제를 신속하게 직접 해결하기 위해 사용할 수 있는 자가 진단 도구를 제공합니다. 지원 포털을 사용하면 다음을 수행할 수 있습니다.

- 서비스 요청 번호를 제출하고 관리합니다.
- 기술 문서를 봅니다.
- 등록하고 제품 알림을 받습니다.
- 소프트웨어 및 기술 설명서를 다운로드합니다.
- 사용 방법 비디오를 시청합니다.
- 커뮤니티 토론에 참여합니다.
- 지원 엔지니어와 온라인으로 채팅합니다.
- Quest 제품으로 지원할 수 있는 서비스를 봅니다.