

# SQL Server

## Datenbankwiederherstellung: Viele Wege führen nach Rom

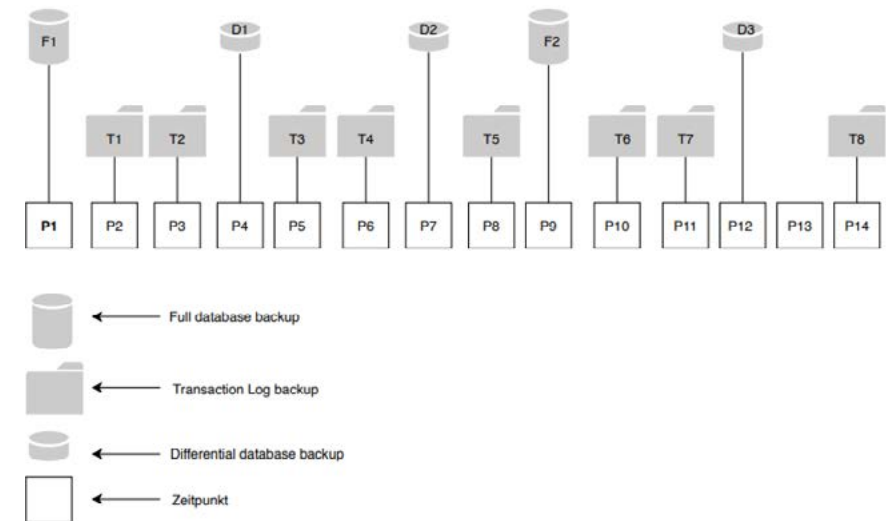
Die verschiedenen Wege, eine Datenwiederherstellung von SQL Servern durchzuführen, werden in diesem Beitrag mit Fokus auf einiger Möglichkeiten näher erörtert. Die einzelnen Prinzipien werden anhand von Datenbanken visualisiert, die das **vollständige Wiederherstellungsmodell (Full Recovery Model)** verwenden.

Dieses Diagramm zeigt für ein besseres Verständnis die verschiedenen Wiederherstellungsmethoden, auf die wir in diesem Artikel näher eingehen:

Das Diagramm repräsentiert drei verschiedene Sicherungstypen:

- × Komplette Sicherung (Full database backup)
- × Differentielle Sicherung (Differential database backup)
- × Transaktionsprotokoll Sicherung (Transaction Log backup)

Das Diagramm zeigt, zu welchem Zeitpunkt die entsprechende Sicherung erstellt wurde. Beispielsweise wurde die Komplette Sicherung F2 zum Zeitpunkt P9 erstellt.



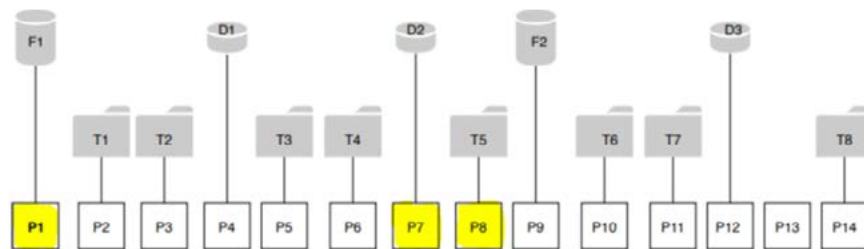
## Vereinfachung einer SQL Server Datenbank Wiederherstellung

Im Folgenden werden wir nun die drei wichtigsten Methoden zur Datenbankwiederherstellung betrachten und erläutern, wie sie unter den jeweiligen Umständen umgesetzt werden können.

### Methode 1: Wiederherstellung zum Zeitpunkt P8

Differenzielle Sicherungen sind kumulativ. Das bedeutet, dass jede differenzielle Sicherung, die nach einer vollständigen Sicherung durchgeführt wurde, lediglich die Änderungen seit dem Zeitpunkt der Erstellung der vollständigen Sicherung beinhaltet. Daher kann nach der Wiederherstellung einer vollständigen Sicherung nur eine einzige differenzielle Sicherung wiederhergestellt werden.

Um nun also eine Wiederherstellung zum Zeitpunkt P8 durchzuführen, sollte erst die Sicherung F1 und anschließend die Sicherung D2 wiederhergestellt werden. Anschließend kann die Wiederherstellung von T5 erfolgen und wir haben den Zustand zum Zeitpunkt P8 wiederhergestellt.



### Methode 2: Wiederherstellung zum Zeitpunkt P10

Im Falle des Verlustes der vollständigen Sicherung F2, kann eine Wiederherstellung von F1 durchgeführt werden. Nun muss, um den Zeitpunkt P10 zu erreichen, eine Kombination aus Differenzial- und Transaktionssicherungen verwendet werden.

Eine vollständige Datenbanksicherung wirkt sich nicht auf eine Kette von Transaktionsprotokollsicherungen aus. So können also Transaktionsprotokolle für frühere vollständige Sicherungen wiederhergestellt werden.

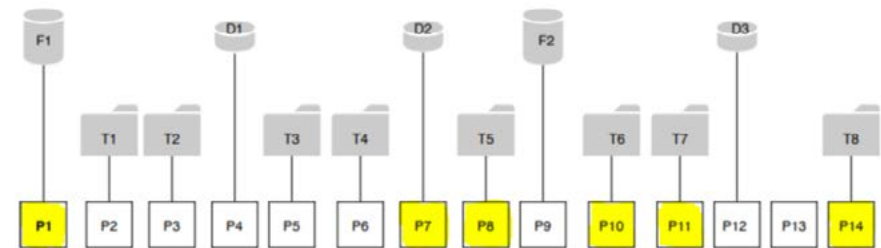
### Methode 3: Wiederherstellung zum Zeitpunkt P13

In diesem Beispiel wollen wir die Datenbank zum Zeitpunkt P13 wiederherstellen.

Allerdings ist nur eine Protokollsicherung an Zeitpunkt P14 verfügbar.

SQL Server bietet nicht die Möglichkeit, einen beliebigen Zeitpunkt durch Wiederherstellen aus einer vollständigen oder differenziellen Sicherung wiederherzustellen. Die einzige Möglichkeit, das zu erreichen, besteht darin, eine Transaktionsprotokoll-Sicherung zu verwenden und hier die STOPAT-Klausel anzugeben. Ausschließlich Transaktionsprotokoll Sicherungen ermöglichen die Wiederherstellung zu einem beliebigen Zeitpunkt.

Um nun also den gewünschten Zeitpunkt zu erreichen, muss eine sogenannte UNDO-Datei verwendet werden, um so die Datenbank im STANDBY-Modus wiederherzustellen und die STOPAT-Klausel angeben zu können. Anschließend kann ein RESTORE aus dem selben Transaktionsprotokoll immer und immer wieder ausgeführt werden, bis der Zeitpunkt P13 erreicht ist.



Dass eine vollständige Sicherung keinen Einfluss auf eine Kette von Transaktionsprotokoll-Sicherungen hat, wollen wir nun an einem kleinen Beispiel zeigen.

Hierfür werden wir den Wiederherstellungspfad F1 > D2 > T5 > T6 > T7 > T8 durchführen.

Hierbei kann angegeben werden, dass eine UNDO-Datei in der Mitte einer RESTORE Sequenz verwendet werden soll. In unserem Beispiel wird die UNDO-Datei beim Wiederherstellen aus dem Transaktionsprotokoll T7 angegeben.

Wir verwenden hier eine SQL Server 2019 Instanz mit dem Namen **TEST\_INSTANCE** und eine Datenbank mit dem Namen **Example**. Mit dem folgenden T-SQL Skript lässt sich der Wiederherstellungspfad durchführen:

```
USE [master]
GO

RESTORE DATABASE [Example] FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL15.TEST_INSTANCE\MSSQL\Backup\full_backup_F1.BAK' WITH FILE = 1,
NORECOVERY, NOUNLOAD, REPLACE, STATS = 5
GO

RESTORE DATABASE [Example] FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL15.TEST_INSTANCE\MSSQL\Backup\diff_backup_D2.BAK' WITH FILE = 1,
NORECOVERY, NOUNLOAD, STATS = 5
GO

RESTORE LOG [Example] FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL15.TEST_INSTANCE\MSSQL\Backup\log_backup_T5.BAK' WITH FILE = 1,
NORECOVERY, NOUNLOAD, STATS = 5
GO

RESTORE LOG [Example] FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL15.TEST_INSTANCE\MSSQL\Backup\log_backup_T6.BAK' WITH FILE = 1,
NORECOVERY, NOUNLOAD, STATS = 5
GO

RESTORE LOG [Example] FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL15.TEST_INSTANCE\MSSQL\Backup\log_backup_T7.BAK' WITH FILE = 1,
STANDBY = N'C:\Restore_log\Example_RollbackUndoFile.tuf', NOUNLOAD, STATS = 5
GO
```

Um nun zu überprüfen ob die Datenbank erfolgreich im Transaktionsprotokoll T7 auf den Zeitpunkt P11 wiederhergestellt wurde und alle erforderlichen Daten vorhanden sind, werfen wir einen Blick in eine – speziell zu diesem Zweck erstellte – Tabelle, in welcher wir gemäß der oberen Grafik sowohl den Zeitpunkt wie auch die Art des Backups protokolliert haben:

```
SELECT * FROM Example.dbo.TranRecord
GO
```

| Results |       | Messages                 |                         |
|---------|-------|--------------------------|-------------------------|
| col1    | Descr | TransactionTimeStamp     |                         |
| 1       | P1    | F1 - Full backup         | 2020-06-04 16:23:33.543 |
| 2       | P2    | T1 - Log Backup          | 2020-06-04 16:23:38.553 |
| 3       | P3    | T2 - Log Backup          | 2020-06-04 16:23:43.557 |
| 4       | P4    | D1 - Differential Backup | 2020-06-04 16:23:48.560 |
| 5       | P5    | T3 - Log Backup          | 2020-06-04 16:23:53.567 |
| 6       | P6    | T4 - Log Backup          | 2020-06-04 16:23:58.577 |
| 7       | P7    | D2 - Differential Backup | 2020-06-04 16:24:03.587 |
| 8       | P8    | T5 - Log Backup          | 2020-06-04 16:24:08.597 |
| 9       | P9    | F2 - Full backup         | 2020-06-04 16:24:13.607 |
| 10      | P10   | T6 - Log Backup          | 2020-06-04 16:24:18.613 |
| 11      | P11   | T7 - Log Backup          | 2020-06-04 16:24:23.623 |
| 12      | P12   | D3 - Differential backup | 2020-06-04 16:24:28.633 |
| 13      | P13   | Restore to here          | 2020-06-04 16:24:33.637 |

Nun können wir das Transaktionsprotokoll T8 mit der zuvor erstellten UNDO-Datei wiederherstellen. Die Transaktionsprotokoll-Wiederherstellung kann nun wiederholt durchgeführt werden, bis der erforderliche Zeitpunkt erreicht ist. Hierfür muss lediglich die in der STOPAT-Klausel angegebene Zeit erhöht werden.

Wir erhöhen nun in der folgenden Wiederherstellungsoperation die STOPAT-Zeit auf eine Sekunde nach dem Zeitpunkt P11:

```
USE [master]
RESTORE LOG [Example] FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL15.TEST_INSTANCE\MSSQL\Backup\log_backup_T8.BAK' WITH FILE = 1,
STANDBY = N'C:\Restore_log\Example_RollbackUndoFile.tuf',
NOUNLOAD, STATS = 10, STOPAT = N'2020-06-04T16:24:24'
GO
```

Als nächstes erhöhen wir das STOPAT-Intervall beim Wiederherstellen aus dem Transaktionsprotokoll T8 um einige Sekunden. Wird nun die STOPAT-Zeit erhöht, können wir zu dem Zeitpunkt P12 gelangen:

```
USE [master]
RESTORE LOG [Example] FROM DISK = N'C:\Program Files\Microsoft SQL Server\
MSSQL15.TEST_INSTANCE\MSSQL\Backup\log_backup_T8.BAK' WITH FILE = 1,
STANDBY = N'C:\Restore_log\Example_RollbackUndoFile.tuf',
NOUNLOAD, STATS = 10, STOPAT = N'2020-06-04T16:24:31'
GO
```

Sobald wir nun den gewünschten Zeitpunkt P13 erreicht haben können wir die Datenbank auf RECOVERY setzen.

```
USE [master]
RESTORE LOG [Example] FROM DISK = N'C:\Program Files\Microsoft SQL Server\
MSSQL15.TEST_INSTANCE\MSSQL\Backup\log_backup_T8.BAK' WITH FILE = 1,
STANDBY = N'C:\Restore_log\Example_RollbackUndoFile.tuf',
NOUNLOAD, STATS = 10, STOPAT = N'2020-06-04T16:24:35'
GO
```

Nachdem wir nun die gewünschten Datensätze haben, können wir mit dem Befehl RESTORE WITH RECOVERY verwenden, welcher die Wiederherstellungssequenz beendet und die Datenbank für Lese- und Schreibaktivitäten wieder verfügbar macht. Nach dem Beenden der Wiederherstellungssequenz können keine weiteren Transaktionsprotokolle wiederhergestellt werden.

```
USE [master] RESTORE DATABASE Example WITH RECOVERY
GO
```