

Recovery Models

Backups

Restores

Recovery Models

- The recovery model tells SQL Server what data to keep in the transaction log file and for how long.
- Based on the recovery model that is selected, this will also determine what types of backups you can perform and also what types of database restores can be performed.
- The three types of recovery models that you can choose from are:
 - Full
 - Bulk-Logged
 - Simple

- Each database can have only one recovery model, but each of your databases can use a different recovery model, so depending on the backup needs you can select the appropriate recovery model per database.
- The only exception to this is the TempDB database which has to use the Simple recovery model.
- Also, the database recovery model can be changed at any time, but this will impact your backup chain, so it is a good practice to issue a full backup after you change your recovery model.

- The recovery model can be changed by either using T-SQL or SQL Server Management Studio.
- Following is how recover model is change using T-SQL examples on how to do this.

```
ALTER DATABASE Movies SET RECOVERY FULL  
GO
```

Full Recovery Model

- The Full recovery model tells SQL Server to keep all transaction data in the transaction log until either a transaction log backup occurs or the transaction log is truncated.
- The way this works is that all transactions that are issued against SQL Server first get entered into the transaction log and then the data is written to the appropriate data file.
- This allows SQL Server to rollback each step of the process in case there was an error or the transaction was cancelled for some reason.
- When the database is set to the Full recovery model since all transactions have been saved you have the ability to do point in time recovery which means you can recover to a point right before a transaction occurred like an accidental deletion of all data from a table.

Reasons why you may choose this recovery model:

- ▶ Data is critical and can not be lost.
- ▶ You always need the ability to do a point-in-time recovery.
- ▶ You are using database mirroring/Log Shipping

Caution - Since every single transaction against a database is going to be stored, the log file will grow too fast and needs to be backed up regularly.

Type of backups you can run when the data is in the Full recovery model:

- ▶ Full backups
- ▶ Differential backups
- ▶ Transaction log backups

Bulk-Logged Recovery Model

- The bulk-logged recovery model is a special-purpose recovery model that should be used only intermittently to improve the performance of certain large-scale bulk operations, such as bulk imports of large amounts of data.
- With this model there are certain bulk operations such as BULK INSERT, CREATE INDEX, SELECT INTO, etc... that are not fully logged in the transaction log and therefore do not take as much space in the transaction log.
- Compared to the full recovery model, which fully logs all transactions, the bulk-logged recovery model minimally logs bulk operations, although fully logging other transactions.

Simple Recovery Model

- The Simple recovery model is the most basic recovery model for SQL Server.
- Every transaction is still written to the transaction log, but once the transaction is complete and the data has been written to the data file the space that was used in the transaction log file is now re-usable by new transactions.
- Since this space is reused it is not possible to do a point in time recovery, therefore the most recent restore point will either be the complete backup or the latest differential backup that was completed.
- Also, since the space in the transaction log can be reused, the transaction log will not grow forever as was mentioned in the Full recovery model.

Reasons why you may choose this recovery model:

- ▶ Your data is not critical and can easily be recreated
- ▶ The database is only used for test or development
- ▶ Data is static and does not change(Reporting databases)
- ▶ Losing any or all transactions since the last backup is not a problem
- ▶ Data is derived and can easily be recreated

Type of backups you can run when the data is in the Simple recovery model:

- ▶ Full backup
- ▶ Differential backup

Backup

- A copy of SQL Server data that can be used to restore and recover a data.
- Backup copies all the objects in the database along with data.
- Backup provides a duplicate copy of the database.

SQL Server can backup a database to:

- local hard disk,
- network hard disk,
- file server hard disk,
- tapes

Why backup a database?

- ▶ Mistakenly deleting a database/Table/View
- ▶ Hardware/Disk failures on the server that hosts a database
- ▶ Human/User error i.e. data deletion, data corruption
- ▶ Natural Disaster
- ▶ The system databases need to be backed up just as user databases are backed up.
- ▶ This allows the system to be rebuilt in the event of system or database failure, for example, if a hard disk fails.

Types of Backups

- The kind of backup that can be configured is dependent up on the recovery model selected.
- The different types of backups that you can create are as follows:
 - ▶ Full backups
 - ▶ Differential backups
 - ▶ Transaction log backups

Full Backup

- The most common types of SQL Server backups are complete or full backups
- A Full backup is a full copy of the database as it currently stands. It allows to completely restoring a database to the state it was at the completion time of the last full backup.
- Full backup create a complete backup of your database as well as part of the transaction log, so the database can be recovered. This allows for the simplest form of database restoration, since all of the contents are contained in one backup.

- ▶ Full backup operations sometimes fails, if sufficient space is not available
- ▶ Takes most time to perform since all information is copied
- ▶ Requires more storage space
- ▶ Schedule to backup a database on a regular basis. For production databases it is common to schedule Full backup everyday.
- ▶ It is the most important type of backup, because without full database backup, you cannot run any other type of backup or can not do any type of restoration.

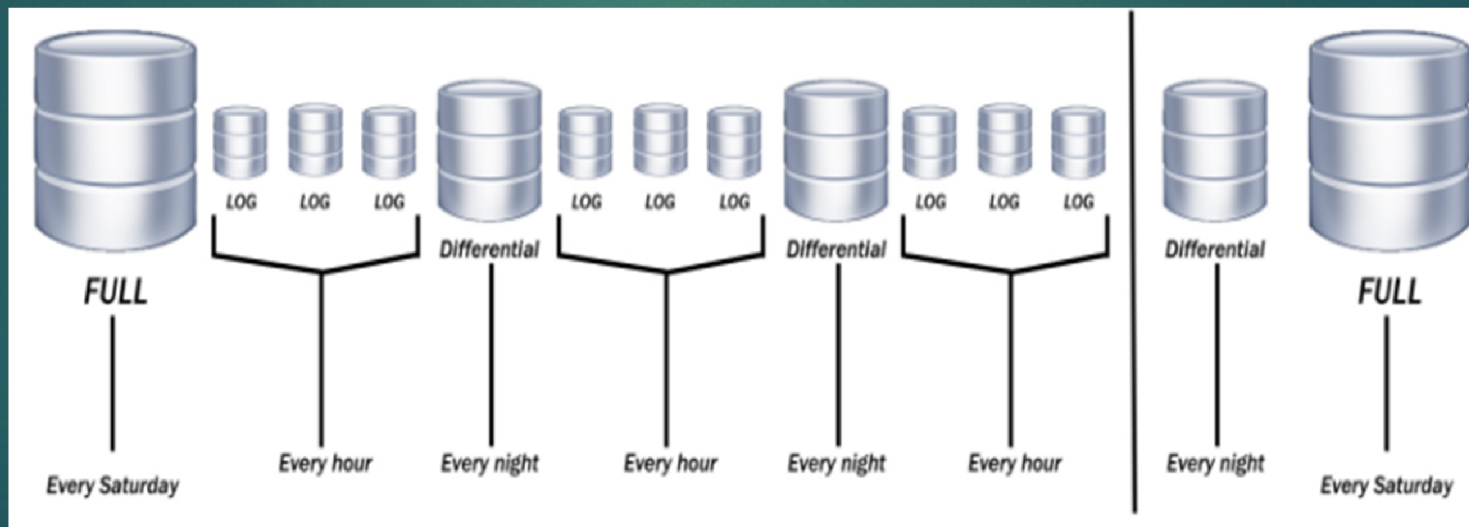
Differential backups

- ▶ A Differential backup is a backup of any data that has changed **since the last Full backup** was created.
- ▶ Must take full backup before you start taking differential backup
- ▶ When a full database backup is performed, any previous differential backups become irrelevant

Transaction log backup

- ▶ A transaction log backup allows you to backup the active part of the transaction log. So after you issue a Full or Differential backup the transaction log backup will have any transactions that were created after those other backups completed.
- ▶ After the transaction log backup is issued, the space within the transaction log can be reused for other processes. If a transaction log backup is not taken, the transaction log will continue to grow creating a serious database problem.

- ▶ Transaction log backups can only be taken once a full database backup has been created, and Transaction log backups can only be restored to a database if a valid Full database backup has been restored first.
- ▶ Only copies the contents of transaction log
- ▶ Transaction logs by themselves cannot restore a database



➤ Full Back Up

```
BACKUP DATABASE AdventureWorks  
TO DISK = 'C:\AdventureWorks.BAK'  
GO
```

➤ Differential backups

```
BACKUP DATABASE AdventureWorks  
TO DISK = 'C:\AdventureWorks.BAK'  
WITH DIFFERENTIAL  
GO
```

➤ Transaction log backup

```
BACKUP LOG [AdventureWorks] TO DISK = N'C:\ AdventureWorks.trn'
```

Backup Strategy

- ▶ A backup strategy/plan is about decision that needs to be made about type of backup, frequency of backup, retention period of backup, and where to backup a database.
- ▶ A backup is utmost important for any system in case there is any disaster, human error, data or disk corruption.
- ▶ Backup strategy differs for different environment. One can decide to do a daily Full backup for Production databases but that may not needed for development or test environment.

Restore

- ▶ Restoration is the process of copying files from a backup for the purpose of recovery or a process of recreating a database from a backup file.
- ▶ Database restore re-creates a previously created database
- ▶ Database restoration is the best process to deploy a database from development to test

Log backup restore

- ▶ Log must be restore in sequence
- ▶ You can not skip a log during the restore
- ▶ The final log backup file can be partially restored(Point in time Recovery)

Key Points

- ▶ Always restore a Full backup first.
- ▶ You may only restore a single differential
- ▶ You don't need to restore any log backups taken BEFORE your differential

- ▶ If you restore a database with “With Recovery” option then the database will be back online and you can not restore any differential or log backups.(This is the default)
- ▶ If there are additional differential or log backup to restore then make sure you select “With NoRecovery”
- ▶ Do not forget to use “NoRecovery” in restoring a sequence of backups EXCEPT the last restore.