Recovery Models in MSSQL and their Impact on Database Restore

Understanding the MSSQL Recovery Models

- The recovery model of a database determines the options a DBA has when recovering a database.
- Each database within SQL Server has a recovery model setting.
  - Simple
  - Full
  - Bulklogged
- What backup and restore options are available for a database.
- How the database engine handles storing transaction log records in the transaction log.
- The transaction log is a detailed log file that is used to record how the database is updated for each transaction.
- The SQL Server engine uses this log to maintain the integrity of the database.
- In the event that a transaction needs to be backed out, or the database crashes for some reason, the transaction log is used to restore the database to a consistent state.

Understanding the MSSQL Recovery Models (contd..)

- The recovery model for a database determines how much data the SQL Server engine needs to write to the transaction log.
- whether or not a point-in-time restore can be performed.
- The initial recovery model setting for a new database is set based on the recovery model of the *model* system database.
- A database’s recovery model setting can be changed easily, by either using SSMS or TSQL code.

Simple Recovery Model

- The simple recovery model is the most basic of recovery models.
- When this recovery model is used, each transaction is still written to the transaction log.
- The transaction logs records will eventually be removed automatically when using the simple recovery model.
- That removal process happens for all completed transactions when a checkpoint occurs.
- Because log records are removed when a checkpoint occurs, transaction log backups are not supported when using the simple recovery model.
- This means point-in-time restores cannot be performed when a database has its recovery model set to SIMPLE.
- Because the transaction log is automatically cleaned up in this mode, this helps keep the transaction log small and from growing out of control.
- you can only perform full and differential backups.
- you can only restore a database to the point-in-time when a full or differential backup has completed.
- Most suited for Development and Test databases, where data loss is acceptable

Full Recovery Model

- The full recovery model supports all the options for backing up and restoring a database.
- Just like the simple recovery model, all transactions are written to the transaction log.
- Unlike the simple recovery model, they stay in the transaction log after the transaction is completed.
- The transaction log records stay in the transaction log until a log backup is performed.
- When a transaction log backup is performed against a database that is in full recovery mode, the log records are written to the transaction log backup, and the completed transaction log records are removed from the transaction log.
- Since every transaction is being written to the transaction log, the full recovery model supports point-in-time restores.
- Meaning a database that is fully logged can be restored to any point in time.
- Also means the transaction log needs to be big enough to support logging of all the transactions until a transaction log backup is run.
- If an application performs lots of transactions, there is the possibility that the transaction log will become full.

Full Recovery Model (contd..)

- When the transaction log becomes full, the database stops accepting transactions until a transaction log backup is taken, the transaction log is expanded, or the transaction log is truncated.
- Therefore, when a database uses the full recovery model, you need to ensure transaction log backups are taken frequently enough to remove the completed transactions from the transaction log before it fills up.
- In addition to inserts and update transaction filling up the log, other operations like index create/alter and bulk load operations also write lots of information to the transaction log.
- Transaction log keeps filling up due to the index and bulk load operations, consider switching to the bulk-logged recovery model while these operations are being performed.

Bulk-Logged Recovery Model

- The bulk-logged recovery model minimizes transaction log space usage when bulk-logged operations like BULK INSERT, SELECT INTO, or CREATE INDEX are executed.
- Bulk-logged recovery model functions similar to the full recovery model with the exception that transactions log records are minimally logged while bulk-logged operations are running.
- Minimal logging helps keep the log smaller, by not logging as much information.
- improves the performance of large bulk loading operations by reducing the amount of logging performed.
- bulk-logged transactions are not fully logged, it reduces the amount of space written to the transaction log, which reduces the chance of the transaction log running out of space.
- Because bulk-logged operations are minimally logged, it affects point-in-time recoveries.
- Point-in-time recoveries can still be performed when using the bulk-logged recovery model in some situations.
- If the database should become damaged while a minimal bulk-logged operation is being performed, the database can only be recovered to the last transaction log backup created prior to the first bulk-logged operation.
- When the transaction log backup contains a bulk logged operation, the stopat options cannot be used.

Bulk-Logged Recovery Model (Contd..)

- If no bulk-logged operations are performed at all while a database is using the bulk-logged recovery model, then you can still do a point-in-time restore just like you can in the full recovery model.
- But keep in mind, during the time a bulk-load operation has occurred, a point-in-time restore cannot be done.
- Therefore to minimize data lose when using bulk-load operations you should take a transaction log backup just prior to a bulk-load operation, and then another one right after the bulk-load operation completes.
- By doing this, a point-in-time recovery can be perform using any transaction log backups taken prior to the bulk-load operation.
- As well as for any transaction log backups taken after the special log backup has been taken following the completion of the bulk-load operation.