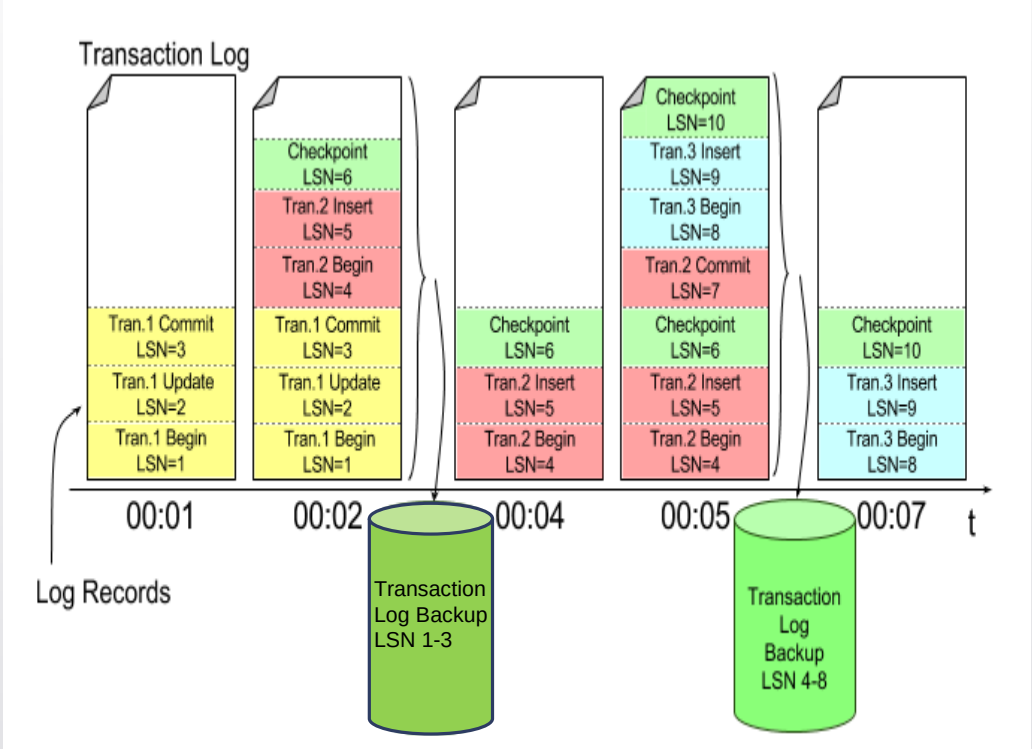
Transaction Log Backups, Log Backup Chain and Tail-log Backup

Transaction Log Backups

- Minimally, you must have created at least one full backup before you can create any log backups.
- After that, the transaction log can be backed up at any time unless the log is already being backed up.
- take log backups frequently, both to minimize work loss exposure and to truncate the transaction log.
- Typically creates a full database backup occasionally, such as weekly.
- creates a series of differential database backup at a shorter interval, such as daily.
- Independent of the database backups, the database administrator backs up the transaction log at frequent intervals.
- For a given type of backup, the optimal interval depends on factors such as the importance of the data, the size of the database, and the workload of the server.

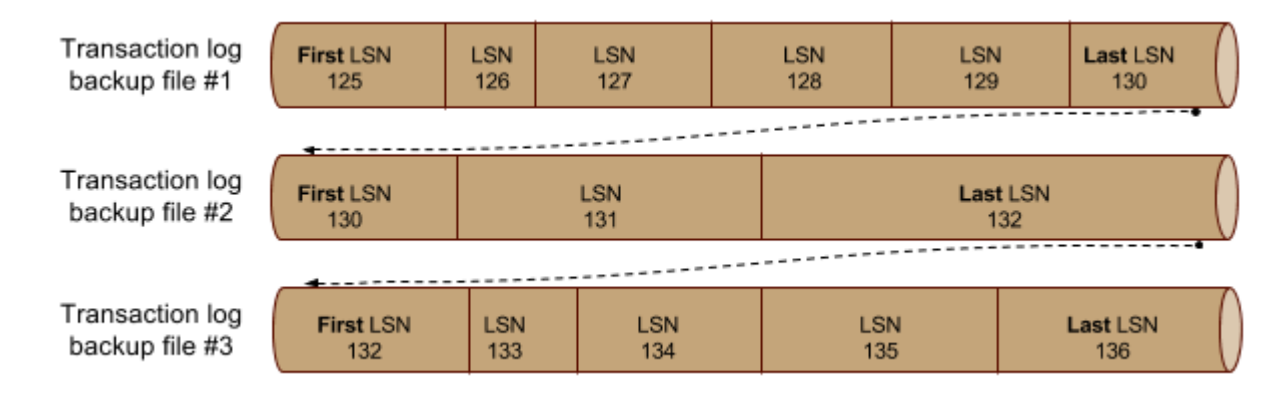
Overview of Transaction Log Backup Backup

- A transaction log backup contains all transaction log records that have been made between the last transaction log backup or the first full backup and the last log record that is created upon completion of the backup process.
- The transaction log backup allows to restore a database to a particular point-in-time before the failure has occurred.
- It is incremental, meaning that in order to restore a database to a certain point-in-time, all transaction log records are required to replay database changes up to that particular point-in-time.



Transaction Log Backup Chain

- A transaction log backup chain is a continuous sequence of transaction log backups starting with first full database backup.
- Each backup from the chain has its own FirstLSN (oldest log record in the backup set) and LastLSN (the number of the next log record after the backup set), that help to restore transaction log backup files in the right sequence.



Transaction Log Backup Chain (contd..)

- To get backup's FirstLSN and LastLSN values you can use the following command:
- RESTORE HEADERONLY FROM DISK = 'log1.bak'
- Some instances when the log backup chain is broken.
- Adding T-SQL options TRUNCATE_ONLY or NO_LOG after a BACKUP LOG command.
- Switching from full or bulk-logged recovery models to simple and back again.
- Two common instances in which the log backup chain is not broken:
- Starting a full, differential or file/filegroup backup.
- Switching from full recovery model to bulk-logged and back again.

Tail–log Backup

- A *tail-log backup* captures any log records that have not yet been backed up (the *tail of the log*) to prevent work loss and to keep the log chain intact.
- Before you can recover a SQL Server database to its latest point in time, you must back up the tail of its transaction log.
- The tail-log backup will be the last backup of interest in the recovery plan for the database.
- Not all restore scenarios require a tail-log backup. You do not need a tail-log backup if the recovery point is contained in an earlier log backup.
- Also, a tail-log backup is unnecessary if you are moving or replacing (overwriting) a database and do not need to restore it to a point of time after its most recent backup.

Log Backup database command

- BACKUP LOG AdventureWork TO DISK = ‘log1.trn’