



CHECKPOINT & SAVEPOINT



MAY 5, 2024
HATEM AL-DOUH

Savepoints:

Savepoints are primarily used within a transaction to provide a way to roll back only a portion of the transaction while preserving the changes made earlier. You would typically use savepoints in the following scenarios:

Complex Transactions: When you have a transaction that involves multiple stages or steps, and you want the ability to undo changes made up to a specific point within the transaction.

Error Handling: If an error occurs during a transaction, you can set a savepoint before the potentially problematic operation. This allows you to roll back to the savepoint if needed, while still preserving the work done before the error.

Nested Transactions: Savepoints can be used to create nested transactions within a larger transaction. This enables you to roll back to a specific savepoint without discarding the work done in earlier stages.

Checkpoints:

Checkpoints, on the other hand, are used at the database system level to ensure data consistency, optimize recovery, and manage log files. You would typically use checkpoints in the following scenarios:

Crash Recovery: Checkpoints provide a starting point for recovery after a system crash or failure. They allow the database system to quickly restore the database to a consistent state by replaying or rolling back transactions from the transaction log.

Backup Operations: Before initiating a database backup, it is common practice to trigger a checkpoint. This ensures that all committed transactions are written to disk, creating a consistent snapshot for the backup.

Log Management: Checkpoints help manage the size of the transaction log. After a checkpoint, the system can truncate or reuse log space occupied by transactions that have been written to disk, allowing for efficient log management and preventing excessive log growth.

If you reuse the name of a savepoint in SQL, the previous savepoint with the same name will be discarded, and a new savepoint will be created with the updated name. The previous savepoint will no longer be accessible or available for rollback operations.

EX:

```
SAVEPOINT sp1;
```

```
-- Some operations
```

```
SAVEPOINT sp1; -- Reusing the name
```

```
-- More operations
```

```
ROLLBACK TO SAVEPOINT sp1;
```

```
-- Rollback to the newly created savepoint with the updated name
```

In the example above, when you reuse the name "sp1" to create a new savepoint, the original savepoint with the name "sp1" is discarded. Any subsequent rollback operation referring to "sp1" will refer to the newly created savepoint, not the previous one.

It's important to note that savepoints are specific to the current transaction and are not accessible outside of it. Once the transaction is committed or rolled back, all savepoints within that transaction are invalidated.

References:

Savepoints

- [SAVE TRANSACTION \(Transact-SQL\) - SQL Server | Microsoft Learn](#)

Checkpoints

- [Database Checkpoints \(SQL Server\) - SQL Server | Microsoft Learn](#)

If you reuse the name of a savepoint in SQL

- [SAVE TRANSACTION \(Transact-SQL\) - SQL Server | Microsoft Learn](#)