**Rollback and Commit Savepoint**

**1. Explain the concept of SAVEPOINT in transaction management. How do ROLLBACK and**

**COMMIT interact with savepoints?**

A SAVEPOINT is a marker or a checkpoint within a database transaction that allows you to partially undo changes without rolling back the entire transaction. It lets you set multiple points inside a transaction so that you can selectively roll back to a specific savepoint if needed.

How SAVEPOINT Works with ROLLBACK and COMMIT

* ROLLBACK TO SAVEPOINT:  
  You can undo all changes made after a specific savepoint by rolling back to it, while keeping the changes made before that savepoint intact. This provides finer control over error handling inside a transaction.
* ROLLBACK (without savepoint):  
  Rolls back the entire transaction, undoing all changes since the transaction began, ignoring any savepoints.
* COMMIT:  
  Finalizes all changes made in the transaction, including those before and after savepoints. After a commit, savepoints are released and cannot be rolled back to.

**2. When is it useful to use savepoints in a database transaction?**

**Savepoints** are useful in a database transaction when you want **fine-grained control over error recovery** within a larger transaction. Specifically, they are helpful:

* When a transaction includes multiple steps, and you want the ability to **undo part of the work without canceling the entire transaction** if an error occurs in a later step.
* In **complex transactions** where certain operations can fail independently, allowing you to roll back only to the last successful savepoint rather than restarting everything.
* When implementing **conditional logic** that requires partial rollbacks depending on business rules.
* To **improve efficiency** by avoiding the cost of rolling back and restarting the entire transaction.
* During **nested operations or batch processing**, where some steps might need to be reversed while keeping others intact.