1. Basic Transaction — Commit / Rollback

Create a table BankAccount with sample records.
Write a transaction that transfers money from one account to another.
If the source account balance becomes negative, roll back the transaction; otherwise commit.

2. Using SAVEPOINT

Insert three new records into a table Orders.
Create a SAVEPOINT after each insert.
Rollback only the second insert using the SAVEPOINT, then commit the remaining inserts.

3. Handling Errors with TRY…CATCH

Write a transaction that updates prices in a Products table.
Introduce a division-by-zero error inside the transaction.
Use TRY…CATCH to rollback the transaction and log the error message in a separate ErrorLog table

4. Nested Transactions

Create nested transactions:

- Outer transaction inserts a customer
- Inner transaction inserts an order for the customer
- Force an error in the inner transaction
  Practice observing whether the outer transaction is committed or rolled back.

5.Isolation Level – Dirty Read

Use two sessions:

- Session 1: Open a transaction, update a row, but don’t commit
- Session 2: Use SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED and read the same row

Check whether dirty reads are allowed.

6. Isolation Level – Non-repeatable Read

Using two sessions:

- Session 1 reads a row twice inside a transaction
- Session 2 updates and commits the same row in between

Observe changes and understand non-repeatable reads.

7. Isolation Level – Phantom Read

Create a table Sales.
Using two sessions:

- Session 1 selects rows between a range inside a transaction
- Session 2 inserts a new row within the range and commits
  See if the first session sees new rows depending on isolation level.

8.Savepoint with Partial Rollback

Inside a transaction:

- Update 5 employee salaries
- Create a savepoint after each update
- Rollback to savepoint 3
- Commit the rest

Check which rows were updated finally.

9.Insert multiple product records using a single transaction.
Force an error in one insert (duplicate key or null value).
Ensure that **no records** are inserted into the table.

10.Savepoint in TRY…CATCH

Inside a long transaction:

- Insert 3 orders
- Savepoint after each
- Force an error before the third insert
  Use savepoint rollback to keep first 2 inserts.

Cursors

1.Basic Cursor – Print All Employee Names

Create a cursor on an Employees table to print each employee's name one by one.

2. Cursor to Update Salary

Create a cursor that increases each employee's salary by 10%.
Update the table inside the cursor loop.

3.Cursor with Conditional Logic

Fetch all orders.
While looping:

- If quantity > 10 → give 5% discount
- If quantity <= 10 → give 2% discount
  Update each order accordingly.

4.Cursor to Copy Data From One Table to Another

Read records from OldProducts table using a cursor and insert them into NewProducts.

5. Cursor to Delete Specific Rows

Create a cursor that loops through customers.
Delete customers whose LastOrderDate is more than 2 years old.