TRANSACTION CONTROL LANGUAGE

1. Introduction to Transaction Control

- Oracle server considers transactions for ensuring data consistency.
- A transaction is a set of one or more DML statements which Oracle treats as a single logical unit of work.
- A transaction groups SQL statements so that they are either all committed (applied to the database) or rolled back (undone from the database).
- All changes made by DML statements are made permanent and visible to other users only after committing the changes.
- DCL and DDL commands always force a commit and explicit committing is not required.

2. Transaction Start and End Scenarios

A Transaction begins when the first executable SQL statement is encountered.

The Transaction terminates when any of the following actions occur.

- A COMMIT or ROLL BACK issued.
- A DDL statement issued
- A DCL statement issued
- Normal shut down from database
- Failure of machine or system crashes

3. Explicit Transaction Control Statements

- COMMIT
- SAVEPOINT
- ROLLBACK

COMMIT:

- It ends the current Transaction by making all pending data changes permanent.
- Once committed, all users can view the recent changes.

Syntax:

COMMIT;

Example:

```
UPDATE employee SET salary =1000 WHERE emp_id = 10;
COMMIT;
```

SAVEPOINT :

- SAVEPOINT is used to mark a point in the current transaction to which we can later roll back.
- We can create multiple savepoints in a single transaction.
- Savepoints can be used to control the reverting of changes.

Syntax:

SAVEPOINT <NAME>;

Example: SAVEPOINT S1;

ROLLBACK:

- Ends the current transaction by discarding all pending data changes.
- The data changes are undone.
- Restores the state of the database to the last commit point.
- The locks on the affected rows are automatically released.

Syntax:

ROLLBACK or **ROLLBACK to <SAVEPOINT NAME>**;

Example:

```
UPDATE employee SET salary =1000 WHERE emp_id = 10;
ROLLBACK;
```

Using SAVEPOINT and ROLLBACK statements, the transaction can be discarded up to a marker.

Example:

A customer gets an offer to purchase items for a maximum of Rs 3000 at free of cost. The customer selects the final item after checking if the net total price is less than 3000. Based on that check, the last item is permanently added to the order.

Below shown is the sample code for the above problem statement.

```
SQL> insert into order111(orderNo,itemNo,Quantity,Price_per_quantity,Total_price) values (10,3,1,1000,1000);

1 row inserted

SQL> insert into order111(orderNo,itemNo,Quantity,Price_per_quantity,Total_price) values (10,5,1,1750,1750);

1 row inserted

SQL> savepoint s1;

Savepoint created

SQL> insert into order111(orderNo,itemNo,Quantity,Price_per_quantity,Total_price) values (10,7,1,300,300);

1 row inserted

SQL> select sum(Total_price) from order111 where orderNo=10;

SUM(TOTAL_PRICE)
----------------
            3050

SQL> rollback to s1;

Rollback complete

SQL> insert into order111(orderNo,itemNo,Quantity,Price_per_quantity,Total_price) values (10,8,1,250,250);

1 row inserted

SQL> commit;

Commit complete
```