**TRANSACTION CONTROLL LANGUAGE**

**TRANSACTION:** A transaction is a unit of work that is performed against a

database or set of statement (Insert, Update and Delete) which should be executed as one unit.

* A transaction is the propagation of one or more changes to the database. For

example, if you are inserting a record or updating a record or deleting a

record from the table, then you are performing transaction on the table. It is

important to control transactions to ensure data integrity and to handle

database errors.

* The rule of transaction tells that either all the statements in the transaction

should be execute successfully or none of those statement to be executed.

To manage transaction we have provide with transaction control language that

provides a commands like

* BEGIN TRANSACTION
* COMMIT
* ROLLBACK
* SAVE POINT

**BEGIN TRANSACTION:** Begin Transaction command is used to start the e

transaction. Begin Transaction with name is used to add nested transactions.

Syntax: Begin transaction

<Write Statements>

**COMMIT:** Commit command is used to end the transaction and save the data

permanent part of the database (or) it is used to make the transaction is permanent so we cannot undo or recall the records.

* Commit is used for saving the data that has been changed permanently

because whenever you perform any DML (Data Manipulation Language)

like UPDATE, INSERT OR DELETE then you are required to write

Commit at the end of all or every DML operation in order to save it

permanently.

* If you do not write Commit then your data will be restored into its previous

condition.

Syntax: Begin Transaction

<Write Statements>

Commit

Ex: BEGIN TRANSACTION

INSERT INTO EMPLOYEE VALUES(105,'KAMAL',62000,'MUMBAI')

INSERT INTO EMPLOYEE VALUES(106,'SUJATHA',82000,'DELHI')

COMMIT

* The above records are stored permanently into a table because we committed that records.so we cannot roll back in to its previous position.

**ROLLBACK:** Rollback command is used to undo the transactions and gets back

to the initial state where transaction started.

* Whereas if you want to restore your data into its previous condition then

you can write Rollback at any time after the DML queries has been written

but remember once Commit has been written then you cannot rollback the

data.

* Moreover you can only rollback the DML queries that have been written

after the last commit statement. The concept of commit and rollback is

designed for data consistency because many users manipulate data of the

same table, using the same database so the user must get updated data.

That is why commit and rollback are used.

Syntax: Begin Transaction

Rollback

Ex: BEGIN TRANSACTION

DELETE FROM EMPLOYEE WHERE EID=105

DELETE FROM EMPLOYEE WHERE EID=106

BEGIN TRANSACTION

ROLLBACK

* The above records we can rollback into a table because those records are not

commited.

**SAVEPOINT:** Save point is used for dividing (or) breaking a transaction into

multiple units. So that user will have a chance of roll backing a transaction up to a

location.

* When a user sets a save point with in a transaction the save point defines a

location to which a transaction can return if part of the transaction

conditionally canceled.

* If a transaction is roll back to a save point, it must be proceed to completion

of the transaction with commit statement or it must be cancelled altogether

by rolling the transaction back to its beginning

Syntax: Begin Transaction

Save transaction < transaction name>

<Write Statements>

Ex: BEGIN TRANSACTION

UPDATE EMPLOYEE SET SALARY=99000 WHERE EID=101

UPDATE EMPLOYEE SET SALARY=88000 WHERE EID=102

SAVE TRANSACTION S1

UPDATE EMPLOYEE SET SALARY=77000 WHERE EID=103

UPDATE EMPLOYEE SET SALARY=66000 WHERE EID=104

SAVE TRANSACTION S2

UPDATE EMPLOYEE SET SALARY=55000 WHERE EID=105

UPDATE EMPLOYEE SET SALARY=44000 WHERE EID=106

Rollback transaction s2

* In the above case we are dividing or breaking the transaction into three

units.so we have a chance of rollbacking either completely i.e six statements

get roll back (or) roll back save point S1 i.e four statements(103 to 106) (or)

rollback save point S2 i.e two records (105,106 ) only

CASE 1: BEGIN TRANSACTION

ROLLBACK

* All records will roll back i.e complete records(six records)

CASE 2: BEGIN TRANSACTION

ROLLBACK TRANSACTION S1

* We can roll back four records only i.e 103 to 106.

CASE 3: BEGIN TRANSACTION

ROLLBACK TRANSACTION S2

* We can roll back two records only i.e 105 and 106