**Transaction control language**

**TCL**

Any DML change on a table is not a permanent one.

We need to save the DML changes in order to make it permanent

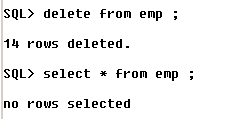
We can also undo (ignore) the same DML changes on a table.

The DDL changes cannot be undone as they are implicitly saved.

**ROLLBACK**

It undoes the DML changes performed on a table.

Let us see in the below example how **rollback** works,

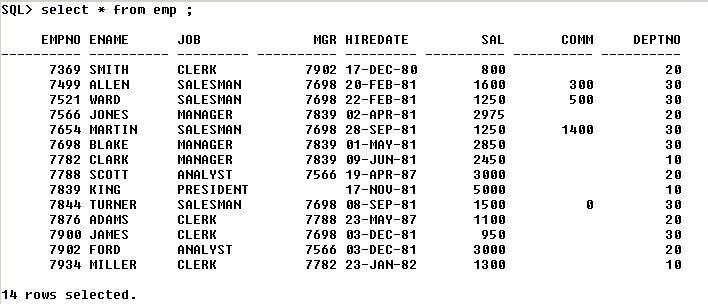


Let us delete the employee table. When we perform **select** operation on emp, we can see that all the rows have been deleted.

We now perform the **rollback** operation,



Now let us perform the **select** operation,



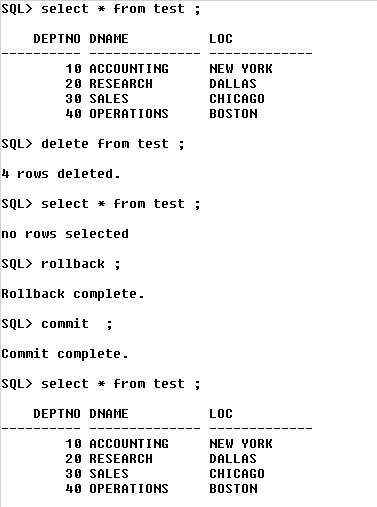
Thus performing the **rollback** operation, we can retrieve all the records which had been deleted.

**COMMIT**

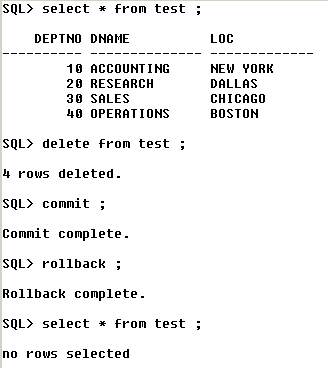
It saves the DML changes permanently to the database.

**Committing after rollback & vice versa will not have any effect**

Let us explain the above statement with an example,



We can see that **commit** has no effect after **rollback** operation.



Thus, from above – we can see that **rollback** has no effect after **commit** operation.

During an abnormal exit – i.e, shutdown or if the SQL window is closed by mouse click – then all the DML’s will be rolled back automatically.

During a normal exit – **exit ;** - all the DML’s will be auto-committed – and there will be no rollback.

**Ex – 1)** INSERT

UPDATE

ALTER

DELETE

ROLLBACK

When we perform the following operations in the same order for a table – then INSERT, UPDATE will be committed – because ALTER is a DDL – and thus all the DML’s above it will also be committed – because DDL operations cannot be undone.

Here – only DELETE will be rolled back because it’s a DML.

**2)** INSERT

UPDATE

DELETE

ROLLBACK

Here, all are rolled back.

**SAVEPOINT** :

It is like a pointer (break-point) till where a DML will be rolled back.

**Ex :-**

Insert …

Save point x ;

Update …

Delete ..

Rollback to x ;

…

…

Here, only DELETE & UPDATE are rolled back.

INSERT is neither rolled back nor committed.

**Data transaction Language(DTL)**

**SavePoint:**

The SAVEPOINT statement names and marks the current point in the processing of a transaction.

With the ROLLBACK TO statement, savepoints undo parts of a transaction instead of the whole transaction

Ex:Rollback to savepoint