Construct SQL queries on Controlling data: commit, rollback and savepoint

Transaction

- A transaction is a logical unit of work from a user.
- A transaction as seen by DBMS is to perform a series of **read** (Select)and **write**(insert, delete, update) **operations**.
- A transaction begins with a DML statement & ends explicitly or implicitly with either rollback or commit statement.
- Transaction changes can be made permanent to the database only if they are committed.

TCL commands

- Commit
- Rollback
- Savepoint

COMMIT

- **COMMIT** is a TCL command in SQL.
- It is used to save all the changes made by a transaction in a database or table.
- Once you execute the COMMIT, the changes become permanent in the database and database cannot go back to its previous state.

COMMIT

Syntax:

SQL> COMMIT;

create a table student as follows

SQL> CREATE TABLE student

(rollno NUMBER(10),

name VARCHAR(20));

COMMIT

Example:

SQL> INSERT INTO student VALUES(1, ’ram’);

SQL> INSERT INTO student VALUES(2, ’sita’);

SQL> COMMIT;

To see the changes saved or not , use the select

SQL> SELECT * FROM student;

ROLLBACK

- **ROLLBACK** is a TCL command in SQL.
- It is used to undo the changes made by the current transaction which is not saved yet in the database.
- One can make use of this command if they wish to undo any changes since the execution of the last **COMMIT.**

Syntax

- SQL> rollback;

ROLLBACK

Example:

SQL>SELECT * FROM student;

ROLLNO	NAME
1	ram
2	sita

SQL> Delete from student where name=‘ram’;

SQL> Delete from student where name=‘sita’;

SQL > rollback;

ROLLBACK

- After rollback is done, use **SELECT** command on the table **student** to see the changes undo or not.
- SQL>SELECT * FROM student;

ROLLNO	NAME
1	ram
2	sita

Savepoint

- **Savepoint** is a TCL command in SQL that is used with the rollback command.
- Save points are like marks to divide a very lengthy transaction to smaller one.
- A **savepoint** is a point in a transaction to which you can rollback the transaction without rolling back the entire transaction.

Savepoint

Syntax for creation of Savepoint

- **SQL>** SAVEPOINT *savepoint_name*;

Syntax for rolling back to Savepoint

- **SQL**> ROLLBACK TO *savepoint_name*;

Savepoint

- SQL>SELECT * FROM student;

1	ram
2	sita
3	ravan

Savepoint

Example:

- SQL**> SAVEPOINT S1;**
- SQL> Delete from student where name=‘**ravan**’;
- SQL> **SAVEPOINT S2;**
- SQL> Delete from student where name=**‘ram**’;
- SQL> **SAVEPOINT S3;**
- SQL> Delete from student where name=‘**sita**’;
- SQL > **ROLLBACK TO S2** ;

Savepoint

SQL> SELECT * FROM student;

ROLLNO	NAME
1	ram
2	sita