

EX NO :1 B	Managing Tables using DML, DCL and TCL Commands
DATE	07/01/2025

### Aim

To execute all commands of data manipulation language to get the desired output.

### Description

#### 1. DML (Data Manipulation Language):

It is used to manipulate and retrieve data in a database.

**Commands used:** Select, Insert, update, delete.

These operations affect the data stored in tables, but not the structure of the database.

#### 2. DCL (Data Control Language):

It is used to manage access and permissions in a database.

**Commands used:** Grant, Revoke.

These operations Controls who can perform operations on the database.

#### 3. TCL (Transaction Control Language):

It is used to manage transactions in a database, ensuring data integrity.

**Commands used:** Commit, Rollback, save point, Set Transaction.

These operations helps to manage atomicity and consistency in database operations.

### Questions:

1. Insert a new user into the "User" table.

```
SQL> INSERT INTO User_URK23CS7129(ID, Name, Email, Password, Phone) VALUES (101, 'John Doe', 'john.doe@example.com', 'password123', 9876543210);
1 row created.

SQL> SELECT * FROM User_URK23CS7129

  ID NAME                                EMAIL
-----
101 John Doe                            john.doe@example.com
password123                            9876543210
```

2. Update the email address of a user with UserID 123.

```
SQL> UPDATE User_URK23CS7129 SET Email = 'updated.email@example.com' WHERE ID = 123;
1 row updated.

SQL> SELECT * FROM User_URK23CS7129 WHERE ID = 123;

  ID NAME                EMAIL
-----
123 Test User            updated.email@example.com
password123              9876543210
```

3. Delete a user with the email "example@example.com".

```
SQL> DELETE FROM User_URK23CS7129 WHERE Email = 'example@example.com';
1 row deleted.
```

4. Insert a new event into the "Event" table.

```
SQL> INSERT INTO Event_URK23CS7129(EventID, Name, DOE, Time, VenueID, Description)
2 VALUES (201, 'Tech Fest', TO_DATE('2024-01-01', 'YYYY-MM-DD'), SYSTIMESTAMP, 101, 'Annual Technology Festival');
1 row created.

SQL> SELECT * FROM Event_URK23CS7129
```

EVENTID	NAME	DOE
201	Tech Fest	01-JAN-24

```

TIME
-----
17-JAN-25 05.59.59.425433 PM
VENUEID
-----
101
DESCRIPTION
-----
Annual Technology Festival
```

5. Update the description of an event with EventID 456.

```
SQL> INSERT INTO Event_URK23CS7129(EventID, Name, DOE, Time, VenueID, Description)
2 VALUES (456, 'Tech Conference', TO_DATE('2024-05-01', 'YYYY-MM-DD'), SYSTIMESTAMP, 101, 'Technology Conference');
1 row created.

SQL> SELECT * FROM Event_URK23CS7129 WHERE EventID = 456;
```

EVENTID	NAME	DOE
456	Tech Conference	01-MAY-24

```

TIME
-----
17-JAN-25 06.01.35.020254 PM
VENUEID
-----
101
DESCRIPTION
-----
Technology Conference
```

6. Grant SELECT privileges on the "User" table to a user named "john".

```
SQL> CREATE USER C##john IDENTIFIED BY john123;
User created.

SQL> GRANT SELECT ON User_URK23CS7129 TO C##john;
Grant succeeded.
```

7. Revoke INSERT privileges on the "Event" table from a user named "mary".

```
SQL> GRANT INSERT ON Event_URK23CS7129 TO C##mary;
Grant succeeded.

SQL> REVOKE INSERT ON Event_URK23CS7129 FROM C##mary;
Revoke succeeded.
```

8. Create a new user with the username "jane" and grant them all privileges on the "Ticket" table.

```
SQL> CREATE USER C##jane IDENTIFIED BY jane_password;
User created.
SQL> GRANT ALL PRIVILEGES ON Ticket_URK23CS7129 TO C##jane;
Grant succeeded.
```

9. Allow the user "Jane" to perform an update operation on the "Ticket" table.

```
SQL> GRANT UPDATE ON Ticket_URK23CS7129 TO C##jane;
Grant succeeded.
```

10. Perform update operation on the "Ticket" table.

```
SQL> UPDATE Ticket_URK23CS7129
2 SET Status = 'Confirmed'
3 WHERE TicketID = 1;
1 row updated.
SQL> SELECT * FROM Ticket_URK23CS7129 WHERE TicketID = 1;
  TICKETID    EVENTID    USERID    PRICE
-----
STATUS
-----
BARCODE
-----
          1         201
Confirmed
```

11. Perform a transaction in the database.

```
SQL> UPDATE User_URK23CS7129 SET Phone = 9876543211 WHERE ID = 101;
1 row updated.
SQL> COMMIT;
Commit complete.
SQL> SELECT * FROM User_URK23CS7129 WHERE ID = 101;
  ID NAME
-----
PASSWORD
-----
PHONE
-----
101 John Doe
password123
9876543211
EMAIL
-----
john.doe@example.com
```

12. Perform roll back a transaction to a specific savepoint.

```
SQL> BEGIN
2   UPDATE User_URK23CS7129 SET Email = 'rollback@gmail.com' WHERE ID = 1;
3   SAVEPOINT savepoint1;
4   UPDATE Ticket_URK23CS7129 SET Status = 'pending' WHERE TicketID = 101;
5   SAVEPOINT savepoint2;
6   ROLLBACK TO savepoint1;
7   COMMIT;
8 END;
9 /

PL/SQL procedure successfully completed.
```

13. Perform set a savepoint within a transaction.

```
SQL> savepoint dev;

Savepoint created.
```

14. Truncate event table.

```
SQL> truncate table event_URK23CS7129;

Table truncated.
```

15. Drop ticket table

```
SQL> drop table ticket_URK23CS7129;

Table dropped.
```

**Result:** Successfully executed DML, DCL, and TCL commands to manipulate, manage, and ensure the integrity of database operations.