Ex no:	1B – Managing Tables using DML, DCL and TCL Commands
Date	19/12/2023

AIM:

The aim is to efficiently manage database tables using Data Control Language (DCL), Data Manipulation Language (DML), and Transaction Control Language (TCL) for secure data access, manipulation, and transaction consistency

DESCRIPTION:

This involves granting and revoking user permissions (DCL), manipulating data through SELECT, INSERT, UPDATE, and DELETE operations (DML), and ensuring transactional integrity with COMMIT, ROLLBACK, and SAVEPOINT commands (TCL). The goal is to maintain a secure, organized, and reliable database system.

In this schema, we have four main tables: User, Event, Venue, and Ticket.

User table:

Column	Data Type
UserID	NUMBER(10)
Name	VARCHAR2(255)
Email	VARCHAR2(255)
Password	VARCHAR2(255)
Phone	VARCHAR2(20)

Event table:

Column	Data Type
EventID	NUMBER(10)
Name	VARCHAR2(255)
Date	DATE

Time	VARCHAR
VenueID	NUMBER(10) (Foreign key)
Description	VARCHAR2(500)

Venue table:

Column	Data Type
VenueID	NUMBER(10)
Name	VARCHAR2(255)
Address	VARCHAR2(255)
City	VARCHAR2(255)
State	VARCHAR2(255)
Country	VARCHAR2(255)

Ticket table:

Column	Data Type
TicketID	NUMBER(10)
EventID	NUMBER(10) (Foreign key)
UserID	NUMBER(10)
SeatNumber	VARCHAR2(20)
Price	NUMBER(10, 2)
Status	VARCHAR2(50)

QUERIS AND OUTPUT SCREENSHOT:

1. Insert a new user into the “User”; table.

```
SQL> INSERT INTO User_urk22ai1048 (UserID, Name, Email) VALUES (1, 'NewUser', 'newuser@example.com');

1 row created.

SQL> select * from user_urk22ai1048
  2  ;

    USERID
----------
NAME
--------------------------------------------------------------------------------
EMAIL
--------------------------------------------------------------------------------
PASSWORD
--------------------------------------------------------------------------------
PHONE
--------------------
         1
NewUser
newuser@example.com
```

2.Update the email address of a user with UserID 1048

```
SQL> UPDATE User_urk22ai1048 SET Email = 'hk@example.com' WHERE UserID = 1;

1 row updated.

SQL> select * from user_urk22ai1048
  2  ;

    USERID
----------
NAME
--------------------------------------------------------------------------------
EMAIL
--------------------------------------------------------------------------------
PASSWORD
--------------------------------------------------------------------------------
PHONE
--------------------
         1
NewUser
hk@example.com
```

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

```
SQL> DELETE FROM User_urk22ai1048 WHERE Email = 'hk@example.com';

1 row deleted.
```

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

```
SQL> INSERT INTO Event_urk22ai1048 (EventID, Name, Description) VALUES (1, 'NewEvent', 'Description of the new event.');

1 row created.

   EVENTID
----------
NAME
--------------------------------------------------------------------------------
EVENTDATE
---------
TIME
--------------------------------------------------------------------------------
   VENUEID
----------
DESCRIPTION
--------------------------------------------------------------------------------
Description of the new event.
```

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

```
SQL> UPDATE Event_urk22ai1048 SET Description = 'Updated description' WHERE EventID = 1;

1 row updated.

   EVENTID
----------
NAME
--------------------------------------------------------------------------------
EVENTDATE
---------
TIME
--------------------------------------------------------------------------------
   VENUEID
----------
DESCRIPTION
--------------------------------------------------------------------------------
Updated description
```

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

```
SQL> GRANT SELECT ON User_urk22ai1048 TO URK22AI1023;

Grant succeeded.
```

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

```
SQL> GRANT INSERT ON EVENT_urk22ai1048 TO URK22AI1023;

Grant succeeded.

SQL> REVOKE INSERT ON Event_URK22AI1048 FROM URK22AI1023;

Revoke succeeded.
```

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

```
CREATE USER jane IDENTIFIED BY password;
```

```
GRANT ALL PRIVILEGES ON Ticket_urk22ai1026 TO jane;
```

9. Allow the user “Jane” to perform update operation on the “Ticket” table.

```
SQL> GRANT UPDATE ON Ticket_URK22AI1048 TO URK22AI1023;

Grant succeeded.
```

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

```
SQL> INSERT INTO ticket_urk22ai1048 (ticketID, status) VALUES (1,  'open');

1 row created.

SQL> UPDATE Ticket_urk22ai1048 SET Status = 'Closed' WHERE TicketID = 1;

1 row updated.

SQL> select * from ticket_urk22ai1048;

  TICKETID    EVENTID     USERID SEATNUMBER                PRICE
---------- ---------- ---------- -------------------- ----------
STATUS
--------------------------------------------------
         1
Closed
```

11. Perform commit a transaction in the database.

```
SQL> COMMIT;

Commit complete.
```

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

```
SQL>
SQL> SAVEPOINT h1;

Savepoint created.

SQL> UPDATE Ticket_urk22ai1048 SET Status = 'open' WHERE TicketID = 1;

1 row updated.

SQL> select * from ticket_urk22ai1048;

  TICKETID    EVENTID     USERID SEATNUMBER                PRICE
---------- ---------- ---------- -------------------- ----------
STATUS
--------------------------------------------------
         1
open


SQL> ROLLBACK TO h1;

Rollback complete.

SQL> select * from ticket_urk22ai1048;

  TICKETID    EVENTID     USERID SEATNUMBER                PRICE
---------- ---------- ---------- -------------------- ----------
STATUS
--------------------------------------------------
         1
Closed
```

13. Perform set a savepoint within a transaction.

```
SQL> SAVEPOINT h1;

Savepoint created.
```

14. Enable autocommit mode in the database.

```
SQL> SET AUTOCOMMIT = ON;
SP2-0158: unknown SET autocommit option "="
Usage: SET AUTO[COMMIT] { OFF | ON | IMM[EDIATE] | n }
```

15. Disable autocommit mode in the database.

```
SQL> SET AUTOCOMMIT = Off;
SP2-0158: unknown SET autocommit option "="
Usage: SET AUTO[COMMIT] { OFF | ON | IMM[EDIATE] | n }
SQL>
```

RESULT:

Managing Tables using DML, DCL and TCL Commands executed successfully to create ,Insert and alter table.