Experiment No. : 9

Title: Write a program to use Data Control Language (DCL) Statements and Transaction Control Language (TCL) Statements.

Objectives:

1. To learn Data Control Language (DCL) Statements.
2. To learn Transaction Control Language (TCL) statements.

Key Concepts: grant, revoke, transaction.

Theory:

Authorization

To maintain the security of database management systems, authorization is required to be strictly followed. Any user should get access to the database object, only if he/she has permission to access it.

User can have various authorities called as privileges to perform different tasks like:

Read authorization/privilege

Allows reading, but not modification of data.

Insert authorization/privilege

Allows insertion of new data, but not modification of existing data.

Update authorization/privilege

Allows modification, but not deletion of data.

Delete authorization/privilege

Allows deletion of data.

References authorization/privilege

Allows to refer the attribute while creating foreign key.

Data Control Language (DCL) Statements

GRANT and REVOKE are the Authorization related statements in SQL.

GRANT:

The GRANT statement is used to grant specific privileges to users or to roles, or to grant roles to users or to roles.

Syntax: grant <privilege list> on <relation name or view name> to <user/role list>
grant select on branch to U1 with grant option

Example: grant select on account to U1, U2, U3
grant update (amount) on loan to U1, U2, U3
grant references (branch-name) on branch to U1

REVOKE:

The REVOKE statement is used to revoke privileges and role grants.

Syntax:

revoke <privilege list> on <relation name or view name> from <user/role list> [restrict | cascade]

Example: revoke select on account from U1

Algorithm for Authorization:

1. Start
2. Create 4 users
3. Create table owned by each user
4. Grant different privileges to different users.
5. Try to perform the operation for which the user is authorized.
6. Try to perform the operation for which the user is not authorized.
7. Stop

Transaction

Transaction is a unit of program execution that accesses and possibly updates various data items. A transaction is a logical unit of work. All changes made to the database can be referred to as a transaction. Transaction changes can be made permanent to the database only if they are committed a transaction begins with an executable SQL statement and ends explicitly with either rollback or commit statement.

Properties of the transactions:

- **Atomicity**: Either all operations of the transaction are reflected properly in the database, or none are.
- **Consistency**: Execution of a transaction in isolation (that is, with no other transaction executing concurrently) preserves the consistency of the database
- **Isolation**: The concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially, i.e., one after the other
- **Durability**: After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

Transaction Control Language (TCL) Statements:

Transaction Control Language (TCL) commands are used to manage transactions in database. These are used to manage the changes made by DML statements. There are following commands used to control transactions:

COMMIT:

Used to save the changes.

The COMMIT command is the transactional command used to save changes invoked by a transaction to the database.

Syntax: SQL> COMMIT;

Example: SQL> COMMIT;

ROLLBACK:

Used to roll back the changes to the database.

The ROLLBACK command is the transactional command used to undo transactions that have not already been saved to the database.

```
Syntax:     ROLLBACK
            ROLLBACK TO SAVEPOINT ID;
Example:    SQL> ROLLBACK;
            SQL> ROLLBACK TO SAVEPOINT SV1;
```

SAVEPOINT:

A SAVEPOINT is a point in a transaction where you can roll the transaction back to a certain point without rolling back the entire transaction.

Used to create the checkpoints to help the recovery system.

It helps to rollback to particular checkpoint.

Syntax: SQL> SAVEPOINT ID;

Example: SQL> SAVEPOINT SV1;

SET TRANSACTION:

Used to set the current transaction as read-only or read/write, establish its isolation level, or assign it to a specified rollback segment.

If want to use SET TRANSACTION statement, then it must be the first statement in the transaction.

A transaction need not have a SET TRANSACTION statement.

Example: SET TRANSACTION READ ONLY NAME 'dkte';

Algorithm for Transaction Processing:

1. Begin Transaction
2. Create Table
3. Insert Sample Data
4. Retrieve Data
5. Create SAVEPOINT
6. Perform some Insert/Delete/Update operations also create some savepoints
7. Demonstrate COMMIT, ROLLBACK, ROLLBACK TO SAVEPOINT
8. End Transaction