Experiment No.7

Perform DCL and TCL commands

Aim :- Write a query to implement Data Control Language(DCL) and Transaction Control Language(TCL) commands

Objective :- To learn DCL commands like Grant and Revoke privileges to the user and TCL commands to commit the transactions and recover it using rollback and save points.

Theory:

Data Control Language:

DCL commands are used to grant and take back authority from any database user.

- Grant
- Revoke

a. Grant: It is used to give user access privileges to a database.

Example

```
1. GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER,
   ANOTHER_USER;
```

b. Revoke: It is used to take back permissions from the user.

Example

```
1. REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

a. Commit: Commit command is used to save all the transactions to the database.

Syntax:

```
1. COMMIT;
```

Example:

```
1. DELETE FROM
   CUSTOMERS
2. WHERE AGE = 25;
   3. COMMIT;
```

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

```
1. ROLLBACK;
```

Example:

```
1. DELETE FROM
   CUSTOMERS
2. WHERE AGE = 25;
   3. ROLLBACK;
```

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

```
2. SAVEPOINT SAVEPOINT_NAME;
```

Implementation:

MY SQL CODE:

```
CREATE TABLE documents (
  doc_id INT PRIMARY KEY,
  doc_name VARCHAR(255),
```

```
  category_id INT,
  user_id INT
);
CREATE TABLE categories (
  category_id INT PRIMARY KEY,
  category_name VARCHAR(50)
);
CREATE TABLE users (
  user_id INT PRIMARY KEY,
  username VARCHAR(50),
  email VARCHAR(100)
);
GRANT SELECT ON documents TO 'myuser'@'localhost';
REVOKE SELECT ON documents FROM 'myuser'@'localhost';
START TRANSACTION;
UPDATE documents SET doc_name = 'Updated Document' WHERE doc_id = 101;
SAVEPOINT my_savepoint;
ROLLBACK TO my_savepoint;
COMMIT;
DESCRIBE documents;
DESCRIBE categories;
DESCRIBE users;
```

Output:

```
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| doc_id      | int          | NO   | PRI | NULL    |       |
| doc_name    | varchar(255) | YES  |     | NULL    |       |
| category_id | int          | YES  |     | NULL    |       |
| user_id     | int          | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| category_id   | int         | NO   | PRI | NULL    |       |
| category_name | varchar(50) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| user_id  | int          | NO   | PRI | NULL    |       |
| username | varchar(50)  | YES  |     | NULL    |       |
| email    | varchar(100) | YES  |     | NULL    |       |
```

Conclusion:

1. Explain about issues faced during rollback in mysql and how it got resolved.

➔ **Issues Faced During Rollback in MySQL and How They Are Resolved**:

- **Rollback** is an essential operation in database transactions. It allows you to undo changes made during a transaction if an error occurs or if you decide not to commit the changes.
- However, there are potential issues related to rollback:
  - **Rollback Failure**:
    - If a rollback fails (e.g., due to a network outage or other unexpected issues), it can lead to data inconsistencies.

        - For example, if a network interruption occurs while the rollback command is running, the database may not receive the rollback instruction.
        - In such cases, the database remains in an "unfinished" state, and the changes made during the transaction may not be fully discarded.
    - **Database Corruption**:
        - If a rollback fails, it does not necessarily mean that the database becomes corrupted.
        - Databases are designed to handle such scenarios by maintaining both the original state of the data and the changes made during the transaction.
        - The server ensures that the final step to commit or rollback is atomic, so either it succeeds or is discarded.
    - **ACID Properties**:
        - ACID (Atomicity, Consistency, Isolation, Durability) properties ensure that databases remain consistent even if physical failures occur during commit or rollback.
        - An ACID-compliant database ensures data integrity and consistency.
- **Resolution**:
    - Properly handle exceptions raised by rollback() in your application code.
    - Ensure that the rollback request reaches the MySQL server.
    - Design your transactions to be as short as possible to minimize the impact of rollback failures.

2. Explain how to create a user in sql.

➔ **Creating a User in SQL**:

To create a user in SQL, you can use the `CREATE USER` command.

Syntax:

```
CREATE USER 'username' IDENTIFIED BY 'password';
```

Example:

```
CREATE USER 'myuser' IDENTIFIED BY 'mypassword';
```

Replace `'username'` with a unique username and `'password'` with a strong password.

The created user can then be granted specific privileges using the `GRANT` command.