Database Systems Lab

Lab # 10

Data Control Language (DCL) and Transaction Control Language (TCL)

Instructor: Engr. Muhammad Usman

Email: usman.rafiq@nu.edu.pk

Course Code: CL2005

Semester Fall 2021

Department of Computer Science,

National University of Computer and Emerging Sciences FAST Peshawar Campus

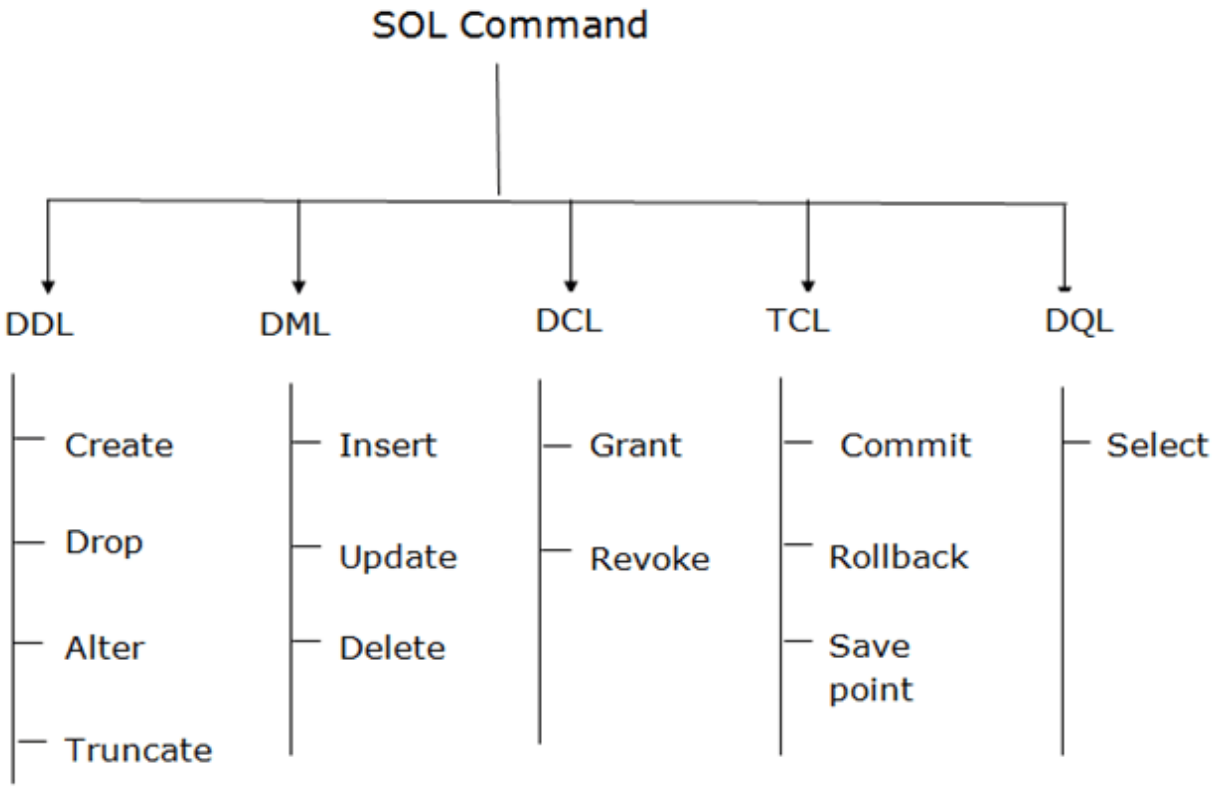
Contents

SQL Overview

Structured Query Language (SQL) as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, and Insert etc. to carry out the required tasks.

These SQL commands are mainly categorized into five categories as:

1. DDL – Data Definition Language **(covered)**
2. DQL – Data Query Language **(covered)**
3. DML – Data Manipulation Language **(covered)**
4. **DCL – Data Control Language** **(Today's Topic)**
5. **TCL – Transaction Control Language** **(Today's Topic)**



1. DDL (Data Definition Language):

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

Examples of DDL commands:

- CREATE – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
- DROP – is used to delete objects from the database.
- ALTER-is used to alter the structure of the database.
- TRUNCATE–is used to remove all records from a table, including all spaces allocated for the records are removed.
- COMMENT –is used to add comments to the data dictionary.
- RENAME –is used to rename an object existing in the database

2. DQL (Data Query Language):

DML statements are used for performing queries on the data within schema objects. The purpose of DQL Command is to get some schema relation based on the query passed to it.

Example of DQL:

- SELECT – is used to retrieve data from the database.

3. DML(Data Manipulation Language):

The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

Examples of DML:

- INSERT – is used to insert data into a table.
- UPDATE – is used to update existing data within a table.
- DELETE – is used to delete records from a database table.

4. DCL(Data Control Language) :

DCL stands for **Data Control Language.**
DCL is used to control user access in a database.
This command is related to the security issues.
Using DCL command, it allows or restricts the user from accessing data in database schema.

DCL commands are as follows,
1. GRANT
2. REVOKE

It is used to grant or revoke access permissions from **any database user.**

Examples of DCL commands:

- **GRANT**-gives user's access privileges to database.
- **REVOKE**-withdraw user's access privileges given by using the GRANT command.

Before going into more detail of DCL, first we will see how to create a user in a database.

Create User

Syntax

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

For Example

```
CREATE USER nutec@'localhost' IDENTIFIED BY 'doe';
```

To change the password of an existing database user use the alter user command:

```
ALTER USER nutec @'localhost' identified by 'allen';
```

In order to switch to a different user in the mysql console do the following:

Type quit which will end the current user's session and write the following command:

```
mysql -u username -p
```

you will be asked for a password, enter the password and you will be logged in.

In order to delete a user, type the following command:

```
DROP USER 'username'@'host';
```

To show/list the users in a MySQL database,

run this MySQL query:

```
Select * from mysql.user;
```

However, note that this query shows all of the columns from the *mysql.user* table, which makes for a lot of output, so as a practical matter you may want to trim down some of the fields to display, something like this:

```
Select host,user,password from mysql.user;
```

5. Data Control Language

GRANT & REVOKE Statements

GRANT Statement

SQL GRANT is a statement used to provide access or privileges on the database objects to the users.

Syntax

```
GRANT privilege_name ON object_name TO ‘[username]’@'localhost’;
```

For Example

```
grant select on themepark.employee to 'nutec'@'localhost';
```

In the above example user ‘nutec’ has been given permission to view the record in the employee table of themepark database.

REVOKE Statement

The REVOKE statement removes user access rights or privileges to the database objects.

Syntax

```
REVOKE privilege_name ON object_name FROM ‘[username]’@'localhost’;
```

For Example

```
revoke select on themepark.employee from 'nutec'@'localhost';
```

this specific command removes user access to view the record in the employee table.

Once you have finalized the permissions that you want to set up for your new users, always be sure to reload all the privileges.

```
FLUSH PRIVILEGES;
```

Your changes will now be in effect.

Example

```
GRANT ALL PRIVILEGES ON * . * TO ‘nutec’@’localhost’;
```

The asterisks in this command refer to the database and table (respectively) that they can access—this specific command allows to the user to read, edit, execute and perform all tasks across all the databases and tables.

```
REVOKE ALL on *.* from ‘nutec’@’localhost’
```

this specific command removes user access to read, edit, execute and perform all tasks across all the databases and tables.

Privileges provided by MySQL

The privileges granted to a MySQL account determine which operations the account can perform. MySQL privileges differ in the contexts in which they apply and at different levels of operation:

- Database privileges apply to a database and to all objects within it. These privileges can be granted for specific databases, or globally so that they apply to all databases.
- Privileges for database objects such as tables, indexes, views, and stored routines can be granted for specific objects within a database, for all objects of a given type within a database (for example, all tables in a database), or globally for all objects of a given type in all databases).

Information about account privileges is stored in the user, db, tables_priv, columns_priv, and procs_priv tables in the mysql system database. The MySQL server reads the contents of these tables into memory when it starts and reloads them “When Privilege Changes Take Effect”. Access-control decisions are based on the in-memory copies of the grant tables.

The following list shows some of the privilege names used in GRANT and REVOKE statements, along with the column name associated with each privilege in the grant tables and the context in which the privilege applies.

- The ALL or ALL PRIVILEGES privilege specifier is shorthand. It stands for "all privileges available at a given privilege level" (except GRANT OPTION). For example, granting ALL at the global or table level grants all global privileges or all table-level privileges.
- The ALTER privilege enables use of the ALTER TABLE statement to change the structure of tables. ALTER TABLE also requires the CREATE and INSERT privileges. Renaming a table requires ALTER and DROP on the old table, CREATE, and INSERT on the new table.
- The ALTER ROUTINE privilege is needed to alter or drop stored routines (procedures and functions).
- The CREATE privilege enables creation of new databases and tables.
- The CREATE ROUTINE privilege is needed to create stored routines (procedures and functions).
- The CREATE USER privilege enables use of the ALTER USER, CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES statements.
- The CREATE VIEW privilege enables use of the CREATE VIEW statement.
- The DELETE privilege enables rows to be deleted from tables in a database.
- The DROP privilege enables you to drop (remove) existing databases, tables, and views.
- The EXECUTE privilege is required to execute stored routines (procedures and functions).
- The GRANT OPTION privilege enables you to give to other users or remove from other users those privileges that you yourself possess.
- The INDEX privilege enables you to create or drop (remove) indexes. INDEX applies to existing tables. If you have the CREATE privilege for a table, you can include index definitions in the CREATE TABLE statement.
- The INSERT privilege enables rows to be inserted into tables in a database.
- The SELECT privilege enables you to select rows from tables in a database. SELECT statements require the SELECT privilege only if they actually retrieve rows from a table. Some SELECT statements do not access tables and can be executed without permission for any database. For example, you can use SELECT as a simple calculator to evaluate expressions that make no reference to tables:

```
SELECT 1+1;
SELECT PI()*2;
```

The SELECT privilege is also needed for other statements that read column values. For example, SELECT is needed for columns referenced on the right hand side of ***col_name = expr*** assignment in UPDATE statements or for columns named in the WHERE clause of DELETE or UPDATE statements.

- The SHOW DATABASES privilege enables the account to see database names by issuing the SHOW DATABASE statement.
- The TRIGGER privilege enables trigger operations. You must have this privilege for a table to create, drop, execute, or display triggers for that table.

  When a trigger is activated (by a user who has privileges to execute INSERT, UPDATE, or DELETE statements for the table associated with the trigger), trigger execution requires that the user who defined the trigger still have the TRIGGER privilege.
- The UPDATE privilege enables rows to be updated in tables in a database.

Advantages of Data Control Language

- ✓ The most important role that the Data Control Language plays is that it provides the security of the database by controlling the access level of the data in the database.
- ✓ The authority to grant or revoke privileges is maintained by the Database Administrator or the owner of the database which effectively prevents any issues those might generate from exposing the database to many users or to multiple database environments.

TCL (Transaction Control Language):

TCL commands deals with the transaction within the database.
These commands are to keep a check on other commands and their effect on the database.
Note: before TCL command use we change/set option that is [SET AUTOCOMMIT = 0;]

Examples of TCL commands:

- **COMMIT**– commits a Transaction.
- **ROLLBACK**– rollbacks a transaction in case of any error occurs.
- **SAVEPOINT**–sets a savepoint within a transaction.

`COMMIT` command

COMMIT command is used to permanently save any transaction into the database.

When we use any DML command like INSERT, UPDATE or DELETE, the changes made by these commands are not permanent, until the current session is closed, the changes made by these commands can be rolled back.

To avoid that, we use the COMMIT command to mark the changes as permanent.

```
COMMIT;
```

ROLLBACK command

This command restores the database to last commited state. It is also used with SAVEPOINT command to jump to a savepoint in an ongoing transaction.

If we have used the UPDATE command to make some changes into the database, and realise that those changes were not required, then we can use the ROLLBACK command to rollback those changes, if they were not commited using the COMMIT command.

Following is rollback command's syntax,

```
ROLLBACK TO savepoint_name;
```

SAVEPOINT command

SAVEPOINT command is used to temporarily save a transaction so that you can rollback to that point whenever required.

Following is savepoint command's syntax,

```
SAVEPOINT savepoint_name;
```

In short, using this command we can **name** the different states of our data in any table and then rollback to that state using the ROLLBACK command whenever required.

A full list of MySQL privileges can be seen here

https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html

For more details:

https://www.tutorialride.com/dbms/sql-data-control-language-dcl.htm
https://www.lifewire.com/data-control-language-dcl-1019477
https://www.tutorialcup.com/dbms/data-control-language-dcl.htm
https://www.educba.com/data-control-language/

Exercises

1. Create a user, user name should be your name.
2. Grants privileges to the user you created
    - To access tables from database themepark (at least 1).
    - To access the views you have created in themepark (at least 1 view).
    - To access full database of blog_app (which you have used in last lab)
    - To create a database
3. Revoke the right to access the view you granted in exercise 2.
4. Grant the insert permission on table ticket and allow him to grant this permission to other users.
5. Give a demonstration of commit, rollback and savepoint commands.
    - You can use any already define database for the demonstration of TCL commands.
    - Disable the autocommit option as shown in lab manual.
    - Insert some data in any table, update some data in any table
    - Now apply rollback and see the result
    - Do some more transactions of insert, update and delete. Afterwards add a savepoint
    - Insert 2 to 3 rows and then go back to previous savepoint and see the results.