Full Stack
Web Programming

SQL Commands: Part 3

Lesson 9

Lesson outline

- Introduction
- DCL – Data Control Language
- TCL – Transaction Control Language

Data Control Language (DCL)

DCL stands for Data Control Language in Structured Query Language (SQL). As the name suggests these commands are used to control privilege in the database. The privileges (Right to access the data) are required for performing all the database operations like creating tables, views, or sequences.

DCL command is a statement that is used to perform the work related to the rights, permissions, and other control of the database system.

Data Control Language (DCL)

There are two types of Privileges in database:

System Privilege
Object Privilege

Data Control Language (DCL)

Why do we need Of DCL commands

- Unauthorized access to the data should be prevented in order to achieve security in our database
- DCL commands maintain the database effectively than anyone else other than database administrator is not allowed to access the data without permission.
- These commands provide flexibility to the data administrator to set and remove database permissions in granular fashion.

Data Control Language (DCL)

Commands in DCL

The two most important DCL commands are:

GRANT
REVOKE

SQL: DCL - GRANT

This command is used to grant permission to the user to perform a particular operation on a particular object. If you are a database administrator and you want to restrict user accessibility such as one who only views the data or may only update the data. You can give the privilege permission to the users according to your wish.

SQL: DCL - GRANT

Syntax:

```
GRANT privilege_list
ON Object_name
TO user_name;
```

SQL: DCL - REVOKE

REVOKE

This command is used to take permission/access back from the user. If you want to return permission from the database that you have granted to the users at that time you need to run REVOKE command.

Syntax:

```
REVOKE privilege_list
ON object_name
FROM user_name;
```

Data Control Language (DCL)

Following commands are granted to the user as a Privilege List:

EXECUTE
UPDATE
SELECT
DELETE
ALTER
ALL

Data Control Language (DCL)

Advantages Of DCL commands

- It allows to restrict the user from accessing data in database.
- It ensures security in database when the data is exposed to multiple users.
- It is the wholesome responsibility of the data owner or data administrator to maintain the authority of grant and revoke privileges to the users preventing any threat to data.
- It prevents other users to make changes in database who have no access to Database

Transaction Control Language (TCL)

TCL stands for Transaction Control Languages. These commands are used for maintaining consistency of the database and for the management of transactions made by the DML commands.

A Transaction is a set of SQL statements that are executed on the data stored in DBMS. Whenever any transaction is made these transactions are temporarily happen in database.So to make the changes permanent, we use TCL commands.

The TCL commands are:

COMMIT
ROLLBACK
SAVEPOINT

SQL: TCL - COMMIT

This command is used to save the data permanently.
Whenever we perform any of the DDL command like -INSERT, DELETE or UPDATE, these can be rollback if the data is not stored permanently. So in order to be at the safer side COMMIT command is used.

Syntax:

```
commit;
```

SQL: TCL - ROLLBACK

This command is used to get the data or restore the data to the last savepoint or last committed state. If due to some reasons the data inserted, deleted or updated is not correct, you can rollback the data to a particular savepoint or if savepoint is not done, then to the last committed state.

Syntax:

```
rollback;
```

SQL: TCL - ROLLBACK

This command is used to get the data or restore the data to the last savepoint or last committed state. If due to some reasons the data inserted, deleted or updated is not correct, you can rollback the data to a particular savepoint or if savepoint is not done, then to the last committed state.

Syntax:

```
rollback;
```

SQL: TCL - SAVEPOINT

This command is used to save the data at a particular point tempoararily, so that whenever needed can be rollback to that particular point.

Syntax:

```
Savepoint A;
```

SQL: TCL - EXAMPLE

Consider the following Table Student:

NAME	MARKS
John	79
Jolly	65
Shuzan	70

SQL: TCL - EXAMPLE

Consider the following Table Student:

NAME	MARKS
John	79
Jolly	65
Shuzan	70

```
UPDATE STUDENT
SET NAME = 'Sherlock'
WHERE NAME = 'Jolly';

COMMIT;
ROLLBACK;
```

SQL: TCL - EXAMPLE

By using this command you can update the record and save it permanently by using COMMIT command.

Now after COMMIT :

NAME	MARKS
John	79
Sherlock	65
Shuzan	70

SQL: TCL - EXAMPLE

If commit was not performed then the changes made by the update command can be rollback.

Now if no COMMIT is performed.

```
UPDATE STUDENT
SET NAME = 'Sherlock'
WHERE STUDENT_NAME = 'Jolly';
```

After update command the table will be:

NAME	MARKS
John	79
Sherlock	65
Shuzan	70

SQL: TCL - EXAMPLE

Now if ROLLBACK is performed on the above table:

```
rollback;
```

After Rollback:

NAME	MARKS
John	79
Jolly	65
Shuzan	70

SQL: TCL - EXAMPLE

If on the above table savepoint is performed:

```
INSERT into STUDENT
VALUES ('Jack', 95);

Commit;

UPDATE NAME
SET NAME= 'Rossie'
WHERE marks= 70;

SAVEPOINT A;

INSERT INTO STUDENT
VALUES ('Zack', 76);

Savepoint B;

INSERT INTO STUDENT
VALUES ('Bruno', 85);

Savepoint C;

SELECT *
FROM STUDENT;
```

SQL: TCL - EXAMPLE

NAME	MARKS
John	79
Jolly	65
Rossie	70
Jack	95
Zack	76
Bruno	85

Now if we Rollback to Savepoint B:

```
Rollback to B;
```

SQL: TCL - EXAMPLE

The resulting Table will be-

NAME	MARKS
John	79
Jolly	65
Rossie	70
Jack	95
Zack	76

Now if we Rollback to Savepoint A:

```
Rollback to A;
```

SQL: TCL - EXAMPLE

The resulting Table will be-

NAME	MARKS
John	79
Jolly	65
Rossie	70
Jack	95

So It was all about TCL commands in SQL (transaction control language) with examples.

Congratulations!