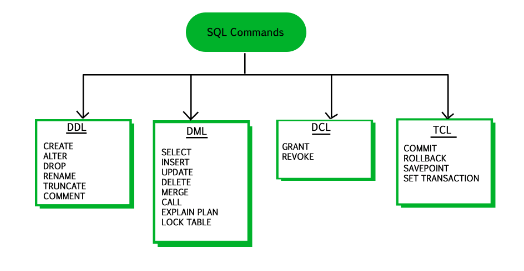
DCL Commands in SQL



DCL is an abbreviation for Data Control Language in SQL. It is used to provide different users access to the stored data. It enables the data administrator to grant or revoke the required access to act as the database. When DCL commands are implemented in the database, there is no feature to perform a rollback. The administrator must implement the other DCL command to reverse the action.

Advertisement

* DCL, DDL, DML, DQL, and TCL commands form the SQL (Structured Query Language).
* DCL commands are primarily used to implement access control on the data stored in the database. It is implemented along the DML (Data Manipulation Language) and DDL (Data Definition Language) commands.
* It has a simple syntax and is easiest to implement in a database.
* The administrator can implement DCL commands to add or remove database permissions on a specific user that uses the database when required.
* DCL commands are implemented to grant, revoke and deny permission to retrieve or modify the data in the database.

Types of DCL Commands in SQL

Two types of DCL commands can be used by the user in SQL. These commands are useful, especially when several users access the database. It enables the administrator to manage access control. The two types of DCL commands are as follows:

* GRANT
* REVOKE

GRANT Command

GRANT, as the name itself suggests, provides. This command allows the administrator to provide particular privileges or permissions over a database object, such as a table, view, or procedure. It can provide user access to perform certain database or component operations.

In simple language, the GRANT command allows the user to implement other SQL commands on the database or its objects. The primary function of the GRANT command in SQL is to provide administrators the ability to ensure the security and integrity of the data is maintained in the database.

To have a better understanding of implementing the GRANT statement in the database. Let us use an example.

Implementing GRANT Statement

Consider a scenario where you are the database administrator, and a student table is in the database. Suppose you want a specific user Aman to only SELECT (read)/ retrieve the data from the student table. Then you can use GRANT in the below GRANT statement.

1. **GRANT** **SELECT** **ON** student **TO** Aman;

This command will allow Aman to implement the SELECT queries on the student table. This will enable the user to read or retrieve information from the student table.

Note: Implementing the above statement will also limit Aman's operations. Aman won't be able to modify the data stored in the table. It will prevent the user from user to insert, to update, or deleting the data in the student table in the database.

REVOKE Command

As the name suggests, revoke is to take away. The REVOKE command enables the database administrator to remove the previously provided privileges or permissions from a user over a database or database object, such as a table, view, or procedure. The REVOKE commands prevent the user from accessing or performing a specific operation on an element in the database.

In simple language, the REVOKE command terminates the ability of the user to perform the mentioned SQL command in the REVOKE query on the database or its component. The primary reason for implementing the REVOKE query in the database is to ensure the data's security and integrity.

Let us use an example to better understand how to implement the REVOKE command in SQL.

Advertisement

Implementing REVOKE Command

Consider a scenario where the user is the database administrator. In the above implementation of the GRANT command, the user Aman was provided permission to implement a SELECT query on the student table that allowed Aman to read or retrieve the data from the table. Due to certain circumstances, the administrator wants to revoke the abovementioned permission. To do so, the administrator can implement the below REVOKE statement:

1. **REVOKE** **SELECT** **ON** student **FROM** Aman;

This will stop the user Aman from implementing the SELECT query on the student table. The user may be able to implement other queries in the database.

Benefits of Implementing DCL Commands

There are several advantages of implementing Data Control Language commands in a database. Let's see some most common reasons why the user implements DCL commands on the database.

1. **Security:** the primary reason to implement DCL commands in the database is to manage the access to the database and its object between different users. This limits the actions that can be performed by specific users on the different elements in the database. It ensures the security and integrity of the data stored in the database.
2. **Granular control:** DCL commands provide granular control to the data administrator over the database. It allows the administrator to provide or remove specific privileges or permissions from other users using a database for information. Thus, it enables the admin to create different levels of access to the database.
3. **Flexibility:** The data administrator can implement DCL commands on specific commands and queries in the database. It allows the administrator to grant or revoke user permissions and privileges as per their needs. It provides flexibility to the administrator that allows them to manage access to the database.

Disadvantages of Implementing DCL Commands

Along with the benefits of implementing DCL commands in the database, they have some disadvantages. Some of the common disadvantages of implementing DCL commands are as follows:

1. **Complexity:** It increases the complexity of database management. If many users are accessing the database, keeping track of permission and privileges provided to every user in the database becomes very complex.
2. **Time-Consuming:** In most organizations, several users access the database, and different users have different access levels to organization data. It is time-consuming to assign the permissions and privileges to each user separately.
3. **Risk of human error:** Human administrators execute DCL commands and can make mistakes in granting or revoking privileges. Thus, giving unauthorized access to data or imposing unintended restrictions on access.
4. **Lack of audit trail:** There may be no built-in mechanism to track changes to privileges and permissions over time. Thus, it is extremely difficult to determine who has access to the data and when that access was granted or revoked.