SQL Commands | DDL, DQL, DML, DCL and TCL Commands

Last Updated : 18 Jun, 2025

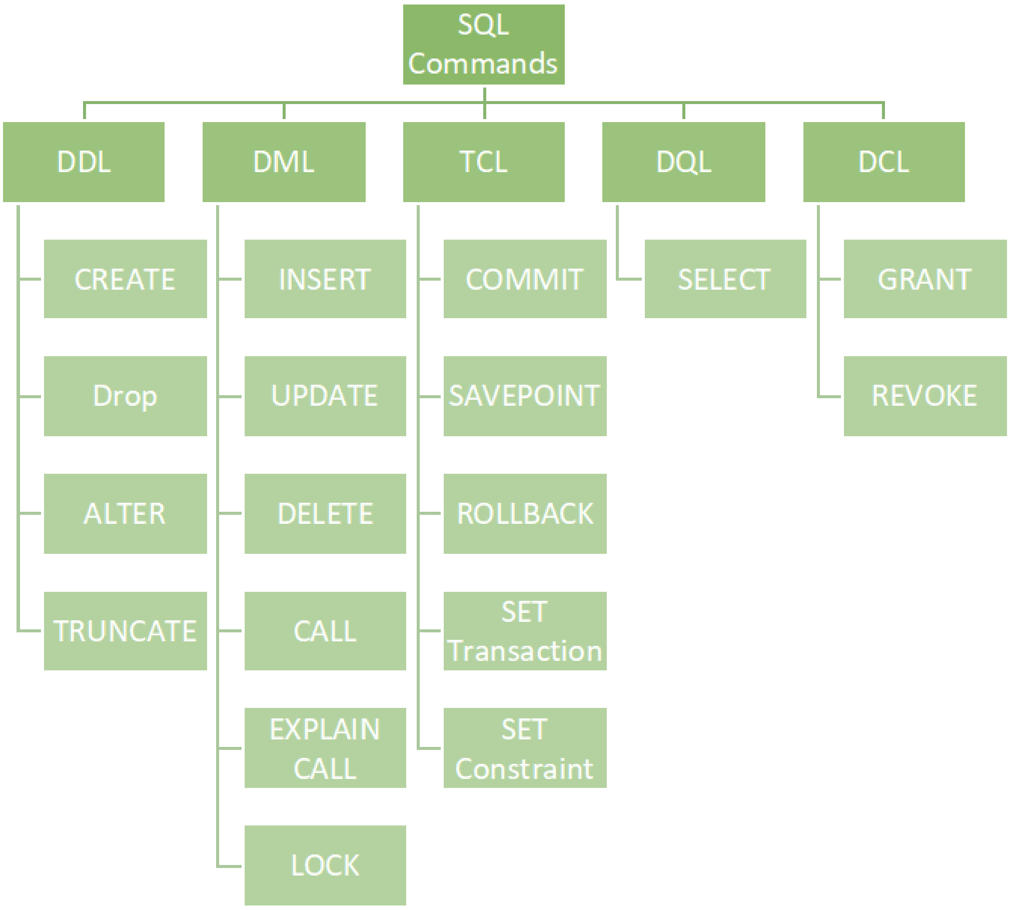
SQL commands are crucial for managing databases effectively. These commands are divided into categories such as Data Definition Language (**DDL**), Data Manipulation Language (**DML**), Data Control Language (**DCL**), Data Query Language (**DQL**), and Transaction Control Language (**TCL**). In this article, we will explain the different types of SQL commands, including DDL, DML, DCL, DQL, and TCL.

What are SQL Commands?

SQL commands are the fundamental building blocks for communicating with a database management system (DBMS). It is used to interact with the database with some operations. It is also used to perform **specific tasks**, **functions**, and **queries** of data. SQL can perform various tasks like creating a table, adding data to tables, dropping the table, modifying the table, set permission for users.

SQL Commands are mainly categorized into five categories:

- **DDL** – Data Definition Language
- **DQL** – Data Query Language
- **DML** – Data Manipulation Language
- **DCL** – Data Control Language
- **TCL** - Transaction Control Language



1. DDL - Data Definition Language

DDL or Data Definition Language actually consists of the SQL commands that can be used for defining, altering and deleting database structures such as tables, indexes and schemas. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database

Common DDL Commands

Command	Description	Syntax
CREATE	Create database or its objects (table, index, function, views, store procedure, and triggers)	`CREATE TABLE table_name (column1 data_type, column2 data_type, ...);`
DROP	Delete objects from the database	`DROP TABLE table_name;`

Command	Description	Syntax
ALTER	Alter the structure of the database	`ALTER TABLE table_name ADD COLUMN column_name data_type;`
TRUNCATE	Remove all records from a table, including all spaces allocated for the records are removed	`TRUNCATE TABLE table_name;`
COMMENT	Add comments to the data dictionary	`COMMENT 'comment_text' ON TABLE table_name;`
RENAME	Rename an object existing in the database	`RENAME TABLE old_table_name TO new_table_name;`

Example:

```
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    hire_date DATE
);
```

In this example, a new table called employees is created with columns for employee ID, first name, last name and hire date.

2. DQL - Data Query Language

DQL statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it. This command allows getting the data out of the database to perform operations with it. When a SELECT is fired against a table or tables the result is compiled into a further temporary table, which is displayed or perhaps received by the program.

DQL Command

Command	Description	Syntax
SELECT	It is used to retrieve data from the database	`SELECT column1, column2, ...FROM table_name WHERE condition;`

Example:

```
SELECT first_name, last_name, hire_date
FROM employees
WHERE department = 'Sales'
ORDER BY hire_date DESC;
```

This query retrieves employees' first and last names, along with their hire dates, from the employees table, specifically for those in the 'Sales' department, sorted by hire date.

3. DML - Data Manipulation Language

The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

Common DML Commands

Command	Description	Syntax
INSERT	Insert data into a table	`INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);`

Command	Description	Syntax
UPDATE	Update existing data within a table	`UPDATE table_name SET column1 = value1, column2 = value2 WHERE condition;`
DELETE	Delete records from a database table	`DELETE FROM table_name WHERE condition;`
LOCK	Table control concurrency	`LOCK TABLE table_name IN lock_mode;`
CALL	Call a PL/SQL or JAVA subprogram	`CALL procedure_name(arguments);`
EXPLAIN PLAN	Describe the access path to data	`EXPLAIN PLAN FOR SELECT * FROM table_name;`

Example:

```
INSERT INTO employees (first_name, last_name, department)
VALUES ('Jane', 'Smith', 'HR');
```

This query inserts a new record into the employees table with the first name 'Jane', last name 'Smith' and department 'HR'.

4. DCL - Data Control Language

DCL (Data Control Language) includes commands such as **GRANT** and **REVOKE** which mainly deal with the rights, permissions and other controls of the database system. These commands are used to control access to data in the database by granting or revoking permissions.

Common DCL Commands

Command	Description	Syntax
GRANT	Assigns new privileges to a user account, allowing access to specific database objects, actions, or functions.	`GRANT privilege_type [(column_list)] ON [object_type] object_name TO user [WITH GRANT OPTION];`
REVOKE	Removes previously granted privileges from a user account, taking away their access to certain database objects or actions.	`REVOKE [GRANT OPTION FOR] privilege_type [(column_list)] ON [object_type] object_name FROM user [CASCADE];`

Example of DCL

```
GRANT SELECT, UPDATE ON employees TO user_name;
```

This command grants the user user_name the permissions to select and update records in the employees table.

5. TCL - Transaction Control Language

Transactions group a set of tasks into a single execution unit. Each transaction begins with a specific task and ends when all the tasks in the group are successfully completed. If any of the tasks fail, the transaction fails. Therefore, a transaction has only two results: success or failure.

Common TCL Commands

Command	Description	Syntax
BEGIN TRANSACTION	Starts a new transaction	`BEGIN TRANSACTION [transaction_name];`

Command	Description	Syntax
COMMIT	Saves all changes made during the transaction	`COMMIT;`
ROLLBACK	Undoes all changes made during the transaction	`ROLLBACK;`
SAVEPOINT	Creates a savepoint within the current transaction	`SAVEPOINT savepoint_name;`

Example:

```
BEGIN TRANSACTION;
UPDATE employees SET department = 'Marketing' WHERE department =
'Sales';
SAVEPOINT before_update;
UPDATE employees SET department = 'IT' WHERE department = 'HR';
ROLLBACK TO SAVEPOINT before_update;
COMMIT;
```

In this example, a transaction is started, changes are made, and a savepoint is set. If needed, the transaction can be rolled back to the savepoint before being committed.

Most Important SQL Commands

There are also a few other SQL Commands we often rely on when writing **powerful queries**. While they don’t fit neatly into the five main categories, they’re absolutely essential for working with data effectively.

Command	Description
SELECT	Retrieves data from one or more tables.
INSERT	Adds new rows (records) to a table.

Command	Description
UPDATE	Modifies existing data in a table.
DELETE	Removes specific rows from a table.
CREATE TABLE	Creates a new table in the database.
ALTER TABLE	Modifies the structure of an existing table (e.g., add or remove columns).
DROP TABLE	Permanently deletes a table and its data.
TRUNCATE TABLE	Removes all rows from a table but keeps its structure intact.
WHERE	Filters records based on a condition.
ORDER BY	Sorts the result set in ascending or descending order.
GROUP BY	Groups rows that have the same values in specified columns.
HAVING	Filters grouped data (used with `GROUP BY`).
JOIN	Combines rows from two or more tables based on a related column.
DISTINCT	Removes duplicate values from the result set.
IN / BETWEEN / LIKE	Used for advanced filtering conditions.
UNION	Combines the result of two or more `SELECT` queries.
GRANT	Gives user privileges or permissions.

Command	Description
REVOKE	Removes user privileges.
COMMIT	Saves all changes made in the current transaction.
ROLLBACK	Undoes changes if something goes wrong in a transaction.
SAVEPOINT	Sets a point in a transaction to roll back to if needed.

Conclusion

SQL commands such as DDL, DML, DCL, DQL, and TCL are foundational for effective database management. From creating and modifying tables with DDL commands to managing transactions with TCL commands in SQL, understanding each type of command enhances our database skills. Whether we are manipulating data, or managing data, SQL provides all sets of tools. Now, with this detailed guide, we hope you have gained a deep understanding of SQL commands, their categories, and syntax with examples.

Comment More info

Advertise with us

Next Article

SQL Database Operations

GeeksforGeeks
Sanchhaya Education Private Limited
Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address: