# SQL COMMANDS | DDL, DQL, DML, DCL AND TCL COMMANDS

**SQL commands** are very used to interact with the database. These commands allow users to perform various actions on a database. This article will teach us about **SQL commands** or **SQL sublanguage commands** like **DDL**, **DQL**, **DML**, **DCL**, and **TCL**.

All important SQL commands with their syntax and examples are covered in this article.

But before heading to the SQL command section, let's briefly introduce SQL.

## Table Of Content



## Short Overview of SQL

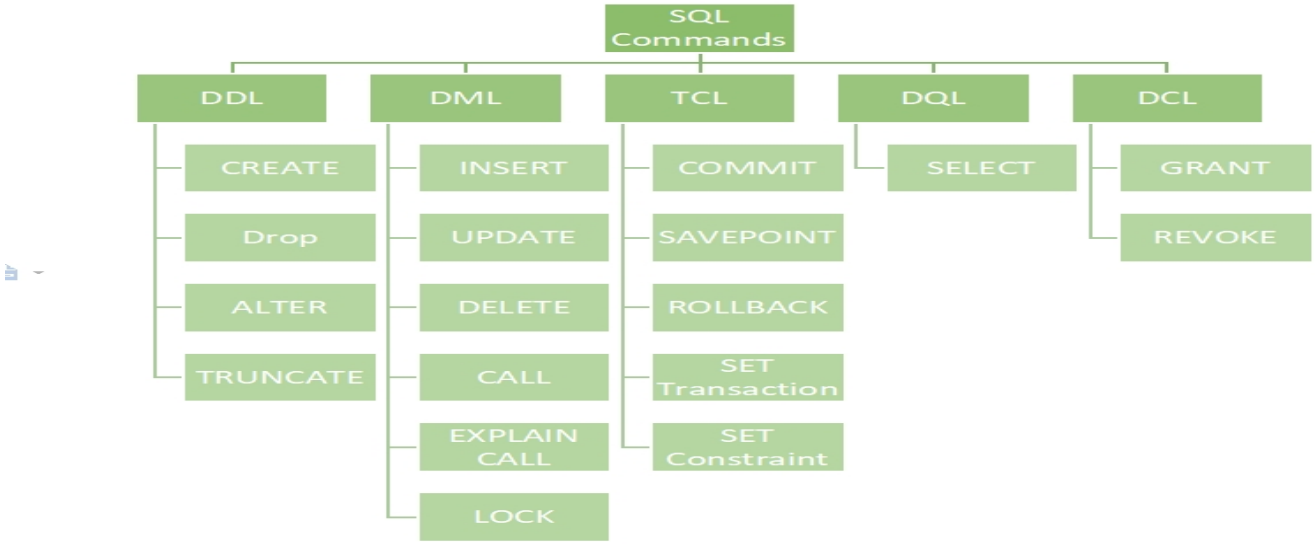
**Structured Query Language (SQL)**, as we all know, is the database language by which we can perform certain operations on the existing database, and we can also use this language to create a database. SQL uses certain commands like CREATE, DROP, INSERT, etc. to carry out the required tasks.

SQL commands are like instructions to a table. 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.

These SQL commands are mainly categorized into five categories:

1. **DDL** – Data Definition Language
2. **DQL** – Data Query Language
3. **DML** – Data Manipulation Language
4. **DCL** – Data Control Language
5. **TCL** – Transaction Control Language

Now, we will see all of these in detail.




# 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.

DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.

### List of DDL Commands

Some DDL commands and their syntax are:

| 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;` |
| **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;` |

# 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. We can define DQL as follows it is a component of SQL statement that allows getting data from the database and imposing order upon it. It includes the SELECT statement.

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 i.e. a front-end.

### DQL Command

There is only one DQL command in SQL i.e.

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

## 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.

### List of DML commands

Some DML commands and their syntax are:

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

## DCL (Data Control Language)

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

### List of DCL commands:

Two important DCL commands and their syntax are:

| 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];` |

## 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. You can explore more about transactions here. Hence, the following TCL commands are used to control the execution of a transaction:

### List of TCL Commands

Some TCL commands and their syntax are:

| Command | Description | Syntax |
|---|---|---|
| **BEGIN TRANSACTION** | Starts a new transaction. | `BEGIN TRANSACTION [transaction_name];` |
| **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;` |

## Important SQL Commands

Some of the most important SQL commands are:

1. **SELECT**: Used to retrieve data from a database.
2. **INSERT**: Used to add new data to a database.
3. **UPDATE**: Used to modify existing data in a database.
4. **DELETE**: Used to remove data from a database.
5. **CREATE TABLE**: Used to create a new table in a database.

6. **ALTER TABLE**: Used to modify the structure of an existing table.
7. **DROP TABLE**: Used to delete an entire table from a database.
8. **WHERE**: Used to filter rows based on a specified condition.
9. **ORDER BY**: Used to sort the result set in ascending or descending order.
10. **JOIN**: Used to combine rows from two or more tables based on a related column between them.

## SQL Commands With Examples

| SQL Command | Example |
|---|---|
| **SELECT** | `SELECT * FROM employees;` |
| **INSERT** | `INSERT INTO employees (first_name, last_name, email) VALUES ('John', 'Doe', 'john.doe@example.com');` |
| **UPDATE** | `UPDATE employees SET email = 'jane.doe@example.com' WHERE first_name = 'Jane' AND last_name = 'Doe';` |
| **DELETE** | `DELETE FROM employees WHERE employee_id = 123;` |
| **CREATE TABLE** | `CREATE TABLE employees ( employee_id INT PRIMARY KEY, first_name VARCHAR(50), last_name VARCHAR(50));` |
| **ALTER TABLE** | `ALTER TABLE employees ADD COLUMN phone VARCHAR(20);` |
| **DROP TABLE** | `DROP TABLE employees;` |
| **WHERE** | `SELECT * FROM employees WHERE department = 'Sales';` |
| **ORDER BY** | `SELECT * FROM employees ORDER BY hire_date DESC;` |
| **JOIN** | `SELECT e.first_name, e.last_name, d.department_name FROM employees e JOIN departments d ON e.department_id = d.department_id;` |

These are common examples of some important SQL commands. The examples provide better understanding of the SQL commands and teaches correct way to use them.

## Conclusion

SQL commands are the foundation of an effective database management system. Whether you 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.