Common SQL Commands

The objective of this reading is to teach you how to name and explain the main commands in SQL. SQL is the most widely used database query language. It is designed for retrieving and managing data in a relational database. SQL can be used to perform different types of operations in the database such as accessing data, describing data, manipulating data and setting users roles and privileges (permissions).

Here you will learn about the main commands used in SQL. At a later stage you will explore relevant examples of how to use these commands with a detailed explanation of the SQL syntax for key operations such as to create, insert, update and delete data in the database.

The SQL Commands are grouped into four categories known as DDL, DML, DCL and TCL depending on their functionality, namely the type of operation they’re used to perform.  Let’s explore these commands in greater detail.

**Data Definition Language (DDL)**

The SQL DDL category provides commands for defining, deleting and modifying tables in a database. Use the following commands in this category.

**CREATE Command**

Purpose: To create the database or tables inside the database

Syntax to create a table with three columns:

1

CREATE TABLE table\_name (column\_name1 datatype(size), column\_name2 datatype(size), column\_name3 datatype(size));

**DROP Command**

Purpose: To delete a database or a table inside the database.

Syntax to drop a table:

1

DROP TABLE table\_name;

**ALTER Command**

Purpose: To change the structure of the tables in the database such as changing the name of a table, adding a primary key to a table, or adding or deleting a column in a table.

1. Syntax to add a column into a table:

1

ALTER TABLE table\_name ADD (column\_name datatype(size));

2. Syntax to add a primary key to a table:

1

ALTER TABLE table\_name ADD primary key (column\_name);

**TRUNCATE Command**

Purpose: To remove all records from a table, which will empty the table but not delete the table itself.

Syntax to truncate a table:

1

TRUNCATE TABLE table\_name;

**COMMENT Command**

Purpose: To add comments to explain or document SQL statements by using double dash (**--**) at the start of the line. Any text after the double dash will not be executed as part of the SQL statement. These comments are not there to build the database. They are only for your own use.

Syntax to **COMMENT** a line in SQL:

1

2

--Retrieve all data from a table

SELECT \* FROM table\_name;

**Data Query Language (DQL)**

The SQL DQL commands provide the ability to query and retrieve data from the database.  Use the following command in this category.

**SELECT Command**

Purpose: To retrieve data from tables in the database.

Syntax to select data from a table:

1

SELECT \* FROM table\_name;

**Data Manipulation Language (DML)**

The SQL DML commands provide the ability to query, delete and update data in the database.  Use the following commands in this category.

**INSERT Command**

Purpose: To add records of data into an existing table. Syntax to insert data into three columns in a table:

1

INSERT INTO table\_name (column1, column2, column3) VALUES (value1, value2, value3);

**UPDATE Command**

Purpose: To modify or update data contained within a table in the database.

Syntax to update data in two columns:

1

UPDATE table\_name SET column1 = value1, column2 = value2 WHERE condition;

**DELETE Command**

Purpose: To delete data from a table in the database.

Syntax to delete data:

1

DELETE FROM table\_name WHERE condition;

**Data Control Language (DCL)**

You use DCL to deal with the rights and permissions of users of a database system. You can execute SQL commands to perform different types of operations such as create and drop tables. To do this, you need to have user rights set up. This is called user privileges. This category deals with advanced functions or operations in the database. Note that this category can have a generic description of the two main commands. Use the following commands in this category:

**GRANT** Command to provide the user of the database with the privileges required to allow users to access and manipulate the database.

**REVOKE** Command to remove permissions from any user.

**Transaction Control Language (TCL)**

The TCL commands are used to manage transactions in the database. These are used to manage the changes made to the data in a table by utilizing the DML commands. It also allows SQL statements to be grouped together into logical transactions. This category deals with advanced functions or operations in a database. Note that this category can have a generic description of the two main commands. Use the following commands in this category:

**COMMIT** Command to save all the work you have already done in the database.

**ROLLBACK** Command to restore a database to the last committed state.

**Integrity constraints**

Every table in a database should abide by rules or constraints. These are known as integrity constraints.

There are three main integrity constraints:

1. Key constraints

2. Domain constraints

3. Referential integrity constraints

**What are key constraints?**

In every table there should be one or more columns or fields that can be used to fetch data from tables. In other words, a primary key. The key constraint specifies that there should be a column, or columns, in a table that can be used to fetch data for any row. This key attribute or primary key should never be NULL or the same for two different rows of data. For example, in the student table I can use the student ID to fetch data for each of the students. No value of student ID is null, and it is unique for every row, hence it can be the key attribute.

**What are domain constraints?**

Domain constraints refer to the rules defined for the values that can be stored for a certain column. For instance, you cannot store the home address of a student in the first name column. Similarly, a contact number cannot exceed ten digits.

**What are referential integrity constraints?**

When a table is related to another table via a foreign key column, then the referenced column value must exist in the other table. This means, according to the student and department examples, that values should exist in the student ID column in the student table because the two tables are related via the student ID column.

In this reading, you learned more about tables in a relational database as you explored the table in terms of its structure, data types, constraints, and the role of primary and foreign keys.

