

SQL DATA DEFINITION

DDL Commands in SQL

DDL is an abbreviation of **Data Definition Language**.

The DDL Commands in Structured Query Language are used to create and modify the schema of the database and its objects. The syntax of DDL commands is predefined for describing the data. The commands of Data Definition Language deal with how the data should exist in the database.

Following are the five DDL commands in SQL:

1. CREATE Command
2. DROP Command
3. ALTER Command
4. TRUNCATE Command
5. RENAME Command

CREATE Command

CREATE is a DDL command used to create databases, tables, triggers and other database objects.

Examples of CREATE Command in SQL

Example 1: This example describes how to create a new database using the CREATE DDL command.

Syntax to Create a Database:

1. **CREATE Database** Database_Name;

Suppose, you want to create a Books database in the SQL database. To do this, you have to write the following DDL Command:

1. **Create Database** Books;

Example 2: This example describes how to create a new table using the CREATE DDL command.

Syntax to create a new table:

1. **CREATE TABLE** table_name
2. (
3. column_Name1 data_type (**size of the column**) ,
4. column_Name2 data_type (**size of the column**) ,

5. column_Name3 data_type (**size of the column**) ,
6. ...
7. column_NameN data_type (**size of the column**)
8.);

Suppose, you want to create a **Student** table with five columns in the SQL database. To do this, you have to write the following DDL command:

1. **CREATE TABLE** Student
2. (
3. Roll_No. **Int** ,
4. First_Name **Varchar** (20) ,
5. Last_Name **Varchar** (20) ,
6. Age **Int** ,
7. Marks **Int** ,
8.);

Example 3: This example describes how to create a new index using the **CREATE DDL** command.

Syntax to Create a new index:

1. **CREATE INDEX** Name_of_Index **ON** Name_of_Table (column_name_1 , column_name_2 , ... , column_name_N);

Let's take the Student table:

Stu_Id	Name	Marks	City	State
100	Abhay	80	Noida	U.P
101	Sushil	75	Jaipur	Rajasthan
102	Ankit	90	Gurgaon	Haryana
103	Yogesh	93	Lucknow	U.P

Suppose, you want to create an index on the combination of the **City** and **State** field of the **Student** table. For this, we have to use the following DDL command:

1. **CREATE INDEX** index_city_State **ON** Employee (Emp_City, Emp_State);

Example 4: This example describes how to create a trigger in the SQL database using the **DDL CREATE** command.

Syntax to create a trigger:

1. **CREATE TRIGGER** [trigger_name]
2. [**BEFORE** | **AFTER**]
3. { **INSERT** | **UPDATE** | **DELETE** }

4. **ON** [table_name] ;

DROP Command

DROP is a DDL command used to delete/remove the database objects from the SQL database. We can easily remove the entire table, view, or index from the database using this DDL command.

Examples of DROP Command in SQL

Example 1: This example describes how to remove a database from the SQL database.

Syntax to remove a database:

1. **DROP DATABASE** Database_Name;

Suppose, you want to delete the Books database from the SQL database. To do this, you have to write the following DDL command:

1. **DROP DATABASE** Books;

Example 2: This example describes how to remove the existing table from the SQL database.

Syntax to remove a table:

1. **DROP TABLE** Table_Name;

Suppose, you want to delete the Student table from the SQL database. To do this, you have to write the following DDL command:

1. **DROP TABLE** Student;

Example 3: This example describes how to remove the existing index from the SQL database.

Syntax to remove an index:

1. **DROP INDEX** Index_Name;

Suppose, you want to delete the index_city from the SQL database. To do this, you have to write the following DDL command:

1. **DROP INDEX** Index_city;

ALTER Command

ALTER is a DDL command which changes or modifies the existing structure of the database, and it also changes the schema of database objects.

We can also add and drop constraints of the table using the ALTER command.

Examples of ALTER Command in SQL

Example 1: This example shows how to add a new field to the existing table.

Syntax to add a newfield in the table:

1. **ALTER TABLE** name_of_table **ADD** column_name column_definition;

Suppose, you want to add the 'Father's_Name' column in the existing Student table. To do this, you have to write the following DDL command:

1. **ALTER TABLE** Student **ADD** Father's_Name **Varchar**(60);

Example 2: This example describes how to remove the existing column from the table.

Syntax to remove a column from the table:

1. **ALTER TABLE** name_of_table **DROP** Column_Name_1 , column_Name_2 ,, column_Name_N;

Suppose, you want to remove the Age and Marks column from the existing Student table. To do this, you have to write the following DDL command:

1. **ALTER TABLE** Student**DROP** Age, Marks;

Example 3: This example describes how to modify the existing column of the existing table.

Syntax to modify the column of the table:

1. **ALTER TABLE** table_name **MODIFY** (column_name column_datatype(**size**));

Suppose, you want to change the character size of the Last_Namefield of the Student table. To do this, you have to write the following DDL command:

1. **ALTER TABLE** table_name **MODIFY** (Last_Name **varchar**(25));

TRUNCATE Command

TRUNCATE is another DDL command which deletes or removes all the records from the table.

This command also removes the space allocated for storing the table records.

Syntax of TRUNCATE command

1. **TRUNCATE TABLE** Table_Name;

Example

Suppose, you want to delete the record of the Student table. To do this, you have to write the following TRUNCATE DDL command:

1. **TRUNCATE TABLE** Student;

The above query successfully removed all the records from the student table. Let's verify it by using the following SELECT statement:

1. **SELECT * FROM** Student;

RENAME Command

RENAME is a DDL command which is used to change the name of the database table.

Syntax of RENAME command

1. RENAME **TABLE** Old_Table_Name **TO** New_Table_Name;

Example

1. RENAME **TABLE** Student **TO** Student_Details ;

This query changes the name of the table from Student to Student_Details.