**DDL Commands**

**DDL (Data Definition Language)**

DDL, or Data Definition Language, consists of SQL commands that can be used to define a database schema. It simply deals with database schema descriptions and is used to create and modify the structure of database objects in the database. DDL is a collection of SQL commands that are used to create, modify, and delete database structures but not data. A general user should access the database through an application, not through these commands.

Let us look into the DDL commands.

**1. CREATE DATABASE**

The CREATE SCHEMA statement is used to create a new SQL database.

Syntax

CREATE SCHEMA *database name*;

Example



**2. CREATE TABLE**

A table is comprised of rows and columns. To create a table, we must define its structure by assigning names to columns and specifying the type and size of data to be stored in columns.

*Syntax*

CREATE TABLE *table\_name* (

*column1 datatype*,

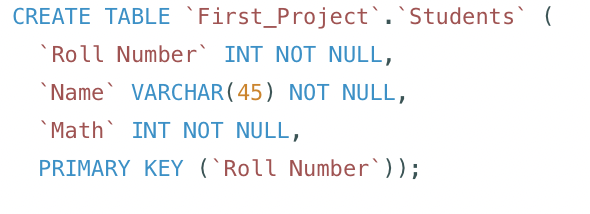
*column2 datatype*,

*column3 datatype*,

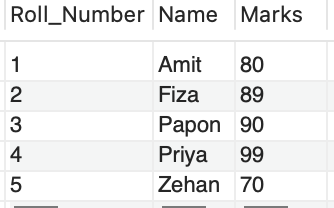
....

);

Example



The table that we created.



**SQL Constraints**

Constraints can be specified when the table is created using the CREATE TABLE statement or afterwards using the ALTER TABLE statement.

Syntax

CREATE TABLE *table* (

*column\_1 datatype* *constraint*,

*column\_2 datatype* *constraint*,

*column\_3 datatype* *constraint*,

....

);

The following SQL constraints are frequently used:

**1. NOT NULL** - Prevents a column from having a NULL value.

**2. UNIQUE** - Ensures that all values in a column are distinct.

Example

CREATE TABLE `STUDENTS\_TABLE` (

`ROLL\_NUMBER` INT NOT NULL,

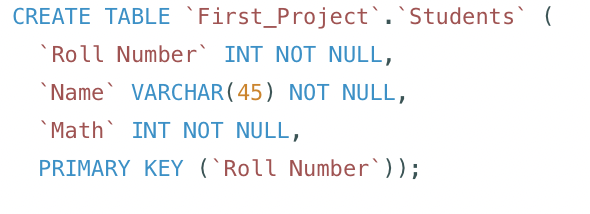
PRIMARY KEY (`ROLL\_NUMBER`),

UNIQUE INDEX `ROLL\_NUMBER\_UNIQUE` (`ROLL\_NUMBER` ASC) VISIBLE);

**3. PRIMARY KEY**

The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must have UNIQUE values and may not have NULL values. A table can only have ONE primary key, which can be made up of single or multiple columns (fields).

When the “Students” table is created, the following SQL creates a PRIMARY KEY on the "Roll Number" column.



**4. FOREIGN KEY**

The FOREIGN KEY constraint prevents actions that would break links between tables.A FOREIGN KEY is a field (or group of fields) in one table that refers to the PRIMARY KEY in another. The table containing the foreign key is referred to as the child table, while the table containing the primary key is referred to as the referenced or parent table.

Example

CREATE TABLE English (

Serial\_No int NOT NULL,

Name int NOT NULL,

Marks int,

PRIMARY KEY (Serial\_No),

FOREIGN KEY (Roll\_Number) REFERENCES Students(Roll\_Number)

);

**2. INSERT INTO**

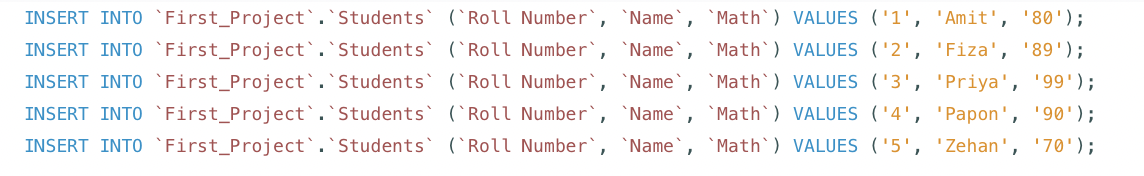
We use INSERT INTO to insert data into a table.

Syntax

INSERT INTO *table\_name*

VALUES (*value1*, *value2*, *value3*, ...);

Example



**3. ALTER TABLE**

The ALTER TABLE statement adds, deletes, or modifies columns in an existing table.

The ALTER TABLE statement can also be used to add or remove constraints from an existing table.

**a. ALTER TABLE - ADD COLUMN**

Use the following syntax to add a column to a table:

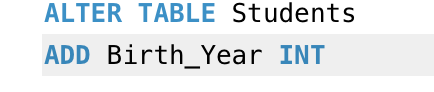
Syntax

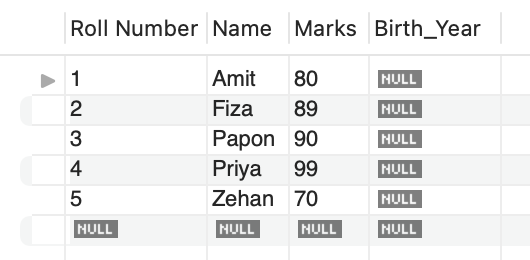
ALTER TABLE *table\_name*

ADD *column\_name datatype*;

Example

We will add a new column, BIRTH\_YEAR, to the existing table Students.

****

****

**b. ALTER TABLE - MODIFY COLUMN**

To change the data type of a table column, use the following syntax.

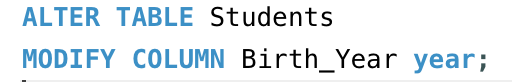
Syntax

ALTER TABLE *table\_name*

MODIFY COLUMN *column\_name datatype*;

Example

Change the datatype of Birth\_Year to the year datatype.



**c. ALTER TABLE - DROP COLUMN**

Use the following syntax to delete a column from a table.

Syntax

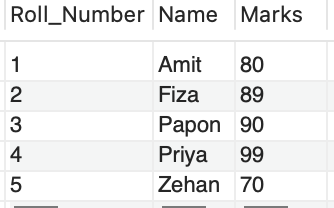
ALTER TABLE *table\_name*

DROP COLUMN *column\_name*;

Example

We will delete the Birth\_Year column.

****



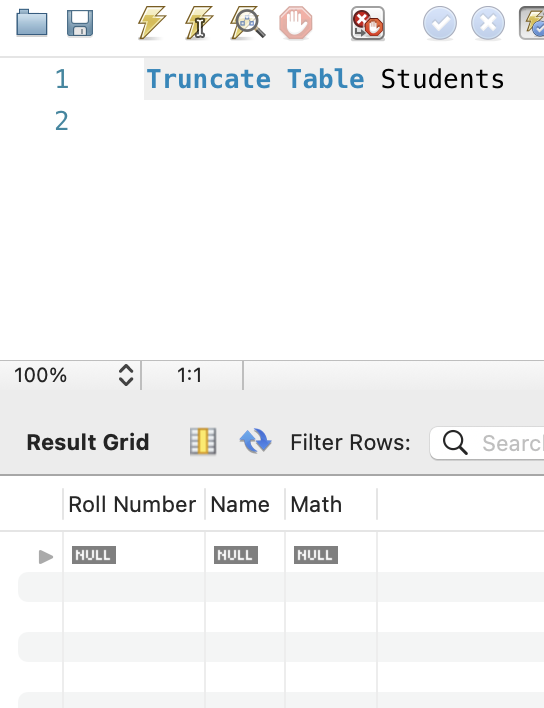
**4. TRUNCATE TABLE**

The TRUNCATE TABLE statement deletes only the data within a table, not the table itself.

Syntax

TRUNCATE TABLE *table\_name*;

Example



**5. DROP TABLE**

To remove an existing table from a database, use the DROP TABLE statement.

Syntax

DROP TABLE *table\_name*;

Example



**6. TRIGGER**

A trigger is a storing procedure of a database that is automatically invoked whenever a special event occurs in the database. A trigger, for example, can be triggered when a row is inserted into a specified table or when certain table columns are updated.

Syntax

create trigger [trigger\_name]

[before | after]

{insert | update | delete}

on [table\_name]

[for each row]

[trigger\_body]

Example

Let us create a table along with Trigger.

CREATE TABLE Citizens (

PersonID int,

Name varchar(255),

City varchar(255)

);

INSERT INTO Citizens VALUES (001, 'Tom B. Erichsen', 'Norway');

INSERT INTO Citizens VALUES (002, 'Jerry Thompson', 'Texas');

INSERT INTO Citizens VALUES (003, 'Michael Depp', 'California');

CREATE TRIGGER ID

BEFORE INSERT

ON Citizens

FOR EACH ROW

SET NEW.PersonID = NEW.PERSONID + 1000000;

INSERT INTO Citizens VALUES (003, 'Jennifer Anniston', 'NY');

