1. **CREATE DATABASE STUDENT;**
2. **SHOW DATABASES;**
3. **USE STUDENT;**
4. **CREATE TABLE STUDENT:**

CREATE TABLE students (

id INTEGER PRIMARY KEY,

first\_name TEXT,

last\_name TEXT,

age INTEGER,

gender TEXT,

email TEXT,

address TEXT,

phone\_number TEXT

);

**In this example, the students table has eight attributes:**

id - an integer column representing the unique identifier of each student (primary key).

first\_name - a text column representing the first name of the student.

last\_name - a text column representing the last name of the student.

age - an integer column representing the age of the student.

gender - a text column representing the gender of the student.

email - a text column representing the email address of the student.

address - a text column representing the address of the student.

phone\_number - a text column representing the phone number of the student.

1. **INSERT VALUES INTO THE TABLE STUDENTS**

INSERT INTO students (id, first\_name, last\_name, age, gender, email, address, phone\_number)

VALUES

(1, 'John', 'Doe', 20, 'Male', 'john.doe@example.com', '123 Main St', '123-456-7890'),

(2, 'Jane', 'Smith', 22, 'Female', 'jane.smith@example.com', '456 Elm St', '987-654-3210'),

(3, 'Michael', 'Johnson', 19, 'Male', 'michael.johnson@example.com', '789 Oak St', '555-123-4567'),

(4, 'Emily', 'Williams', 21, 'Female', 'emily.williams@example.com', '321 Maple Ave', '111-222-3333'),

(5, 'Daniel', 'Brown', 20, 'Male', 'daniel.brown@example.com', '444 Pine St', '444-555-6666'),

(6, 'Olivia', 'Jones', 23, 'Female', 'olivia.jones@example.com', '567 Cedar Ave', '777-888-9999'),

(7, 'William', 'Taylor', 20, 'Male', 'william.taylor@example.com', '890 Walnut St', '222-333-4444'),

(8, 'Sophia', 'Anderson', 21, 'Female', 'sophia.anderson@example.com', '123 Oakwood Ave', '888-999-0000'),

(9, 'Matthew', 'Thomas', 22, 'Male', 'matthew.thomas@example.com', '456 Maplewood Dr', '555-777-8888'),

(10, 'Ava', 'Jackson', 20, 'Female', 'ava.jackson@example.com', '789 Pinecrest Rd', '777-000-1111'),

(11, 'Jacob', 'White', 23, 'Male', 'jacob.white@example.com', '321 Elmwood Ln', '444-222-3333'),

(12, 'Mia', 'Harris', 21, 'Female', 'mia.harris@example.com', '654 Pinehill Ave', '666-777-8888'),

(13, 'Ethan', 'Martin', 20, 'Male', 'ethan.martin@example.com', '987 Cedarwood Rd', '111-222-3333'),

(14, 'Charlotte', 'Thompson', 22, 'Female', 'charlotte.thompson@example.com', '753 Oakhill Dr', '888-999-1111'),

(15, 'Alexander', 'Garcia', 19, 'Male', 'alexander.garcia@example.com', '369 Maple Ave', '333-444-5555'),

(16, 'Amelia', 'Davis', 21, 'Female', 'amelia.davis@example.com', '852 Willow St', '555-666-7777'),

(17, 'James', 'Rodriguez', 20, 'Male', 'james.rodriguez@example.com', '147 Elmwood Ave', '777-888-9999'),

(18, 'Lily', 'Lopez', 22, 'Female', 'lily.lopez@example.com', '963 Pine St', '222-333-4444'),

(19, 'Benjamin', 'Wilson', 23, 'Male', 'benjamin.wilson@example.com', '258 Oakwood Ln', '666-777-8888'),

(20, 'Grace', 'Lee', 20, 'Female', 'grace.lee@example.com', '753 Maple Ave', '111-222-3333'),

(21, 'Daniel', 'Hall', 22, 'Male', 'daniel.hall@example.com', '369 Cedarwood Dr', '444-555-6666'),

(22, 'Victoria', 'Young', 21, 'Female', 'victoria.young@example.com', '852 Elm St', '777-888-9999'),

(23, 'Henry', 'Gonzalez', 20, 'Male', 'henry.gonzalez@example.com', '456 Pinecrest Ave', '222-333-4444'),

(24, 'Sofia', 'Clark', 23, 'Female', 'sofia.clark@example.com', '963 Oakwood Rd', '555-666-7777'),

(25, 'Christopher', 'Lewis', 21, 'Male', 'christopher.lewis@example.com', '147 Maplewood Ave', '888-999-1111');

1. **DESC Students;**

The DELETE statement is used to delete rows from a table. If you want to remove a specific row from a table you should use WHERE condition.

**DELETE** **FROM** table\_name [**WHERE** condition];

But if you do not specify the WHERE condition it will remove all the rows from the table.

**DELETE** **FROM** table\_name;

There are some more terms similar to DELETE statement like as DROP statement and TRUNCATE statement, but they are not exactly same there are some differences between them.

1. **To find how much space is taken by the student table in SQL:**

SELECT

TABLE\_NAME AS `students`,

ROUND((DATA\_LENGTH + INDEX\_LENGTH) / 1024 / 1024) AS `Size (MB)`

FROM

INFORMATION\_SCHEMA.TABLES

WHERE

TABLE\_SCHEMA = 'student'

AND TABLE\_NAME = 'students'

ORDER BY

(DATA\_LENGTH + INDEX\_LENGTH) DESC;

1. **Difference between DELETE and TRUNCATE statements**

There is a slight difference b/w delete and truncate statement. The DELETE statement only deletes the rows from the table based on the condition defined by WHERE clause or delete all the rows from the table when condition is not specified.

But it does not free the space containing by the table.

The TRUNCATE statement: it is used to delete all the rows from the table and free the containing space.

**TRUNCATE** **TABLE** students;

1. **Difference b/w DROP and TRUNCATE statements**

When you use the drop statement it deletes the table's row together with the table's definition so all the relationships of that table with other tables will no longer be valid.

When you drop a table:

Table structure will be dropped

Relationship will be dropped

Integrity constraints will be dropped

Access privileges will also be dropped

On the other hand, when we TRUNCATE a table, the table structure remains the same, so you will not face any of the above problems.

1. **DROP Table:**

DROP TABLE STUDENTS;

1. **Syntax of RENAME statement in SQL**

RENAME old\_table \_name To new\_table\_name ;

RENAME TABLE STUDENTS TO STUDENT;

1. **Copy table into another table Syntax:**

CREATE TABLE New\_table\_name AS

SELECT \* FROM old\_table\_name;

CREATE TABLE STUDENTS AS SELECT \* FROM STUDENT;

1. **SQL ALTER TABLE**

## Syntax of ALTER TABLE ADD Column statement in SQL

ALTER TABLE table\_name ADD column\_name column-definition;

ALTER TABLE Students ADD `Father's\_Name` TEXT;

1. **UPDATE Father’s Name Column only**

UPDATE Students

SET `Father's\_Name` = (

CASE

WHEN id = 1 THEN 'John Doe Sr.'

WHEN id = 2 THEN 'Michael Smith Sr.'

WHEN id = 3 THEN 'William Johnson Sr.'

WHEN id = 4 THEN 'David Brown Sr.'

WHEN id = 5 THEN 'Robert Davis Sr.'

WHEN id = 6 THEN 'James Miller Sr.'

WHEN id = 7 THEN 'Daniel Wilson Sr.'

WHEN id = 8 THEN 'Thomas Taylor Sr.'

WHEN id = 9 THEN 'Joseph Anderson Sr.'

WHEN id = 10 THEN 'Charles Clark Sr.'

WHEN id = 11 THEN 'Christopher Martinez Sr.'

WHEN id = 12 THEN 'Matthew Jones Sr.'

WHEN id = 13 THEN 'Anthony Rodriguez Sr.'

WHEN id = 14 THEN 'Mark Thompson Sr.'

WHEN id = 15 THEN 'Paul Garcia Sr.'

WHEN id = 16 THEN 'Donald Martinez Sr.'

WHEN id = 17 THEN 'George Harris Sr.'

WHEN id = 18 THEN 'Kenneth Wilson Sr.'

WHEN id = 19 THEN 'Steven Thomas Sr.'

WHEN id = 20 THEN 'Edward Martinez Sr.'

WHEN id = 21 THEN 'Brian Davis Sr.'

WHEN id = 22 THEN 'Ronald Clark Sr.'

WHEN id = 23 THEN 'Timothy Young Sr.'

WHEN id = 24 THEN 'Jason Harris Sr.'

WHEN id = 25 THEN 'Jeffrey Lewis Sr.'

END

)

WHERE id BETWEEN 1 AND 25;

1. **Modify the column definition**

ALTER TABLE student.students

MODIFY COLUMN `Father's\_Name` VARCHAR(100);

1. **Rename Column:**

ALTER TABLE students

RENAME COLUMN `father's\_name` TO `father's name`;

1. **Drop Column:**

ALTER TABLE STUDENTS

DROP COLUMN `father's name`;

ALTER TABLE STUDENTS

DROP COLUMN `father's name`;

1. **SELECT Statement**

**CREATE** **TABLE** Student\_Records

(

Student\_Id **Int** **PRIMARY** **KEY**,

First\_Name **VARCHAR** (20),

Address **VARCHAR** (20),

Age **Int** NOT NULL,

Percentage **Int** NOT NULL,

Grade **VARCHAR** (10)

) ;

INSERT INTO Student\_Records

VALUES (201, 'Akash', 'Delhi', 18, 89, 'A2'),

(202, 'Bhavesh', 'Kanpur', 19, 93, 'A1'),

(203, 'Yash', 'Delhi', 20, 89, 'A2'),

(204, 'Bhavna', 'Delhi', 19, 78, 'B1'),

(205, 'Yatin', 'Lucknow', 20, 75, 'B1'),

(206, 'Ishika', 'Ghaziabad', 19, 51, 'C1'),

(207, 'Vivek', 'Goa', 20, 62, 'B2');

**SELECT** Student\_Id, Age, Percentage, Grade **FROM** Student\_Records;

1. **SELECT Statement with WHERE clause**

The WHERE clause is used with SELECT statement to return only those rows from the table, which satisfy the specified condition in the query.

In SQL, the WHERE clause is not only used with SELECT, but it is also used with other SQL statements such as UPDATE, ALTER, and DELETE statements.

**Syntax of SELECT Statement with WHERE clause**

**SELECT** \* **FROM** Name\_of\_Table **WHERE** [condition];

**CREATE** **TABLE** Employee\_Details

(

Employee\_ID **INT** AUTO\_INCREMENT **PRIMARY** **KEY**,

Emp\_Name **VARCHAR** (50),

Emp\_City **VARCHAR** (20),

Emp\_Salary **INT** NOT NULL,

Emp\_Panelty **INT** NOT NULL

) ;

**INSERT INTO Employee\_Details (Employee\_ID, Emp\_Name, Emp\_City, Emp\_Salary, Emp\_Panelty)**

**VALUES (101, 'Anuj', 'Ghaziabad', 25000, 500),**

**(102, 'Tushar', 'Lucknow', 29000, 1000),**

**(103, 'Vivek', 'Kolkata', 35000, 500),**

**(104, 'Shivam', 'Goa', 22000, 500);**

**SELECT** \* **FROM** Employee\_Details **WHERE** Emp\_Panelty = 500;

1. **SQL SELECT Statement with GROUP BY clause**

The GROUP BY clause is used with the SELECT statement to show the common data of the column from the table:

**Syntax of SELECT Statement with GROUP BY clause**

**SELECT** column\_Name\_1, column\_Name\_2, ....., column\_Name\_N aggregate\_function\_name(column\_Name2) **FROM** table\_name **GROUP** **BY** column\_Name1;

**Example of SELECT Statement with GROUP BY clause**

Use the following query to create the **Cars\_Details** table:

CREATE TABLE Cars\_Details

(

Car\_Number INT PRIMARY KEY,

Car\_Name VARCHAR(50),

Car\_Price INT NOT NULL,

Car\_Amount INT NOT NULL

);

The following INSERT query inserts the record of cars into the **Cars\_Details** table:

INSERT INTO Cars\_Details (Car\_Number, Car\_Name, Car\_Amount, Car\_Price)

VALUES (2578, 'Creta', 3, 1500000),

(9258, 'Audi', 2, 3000000),

(8233, 'Venue', 6, 900000),

(6214, 'Nexon', 7, 1000000);

**SELECT COUNT(Car\_Name), Car\_Price**

**FROM Cars\_Details**

**GROUP BY Car\_Price;**

**Update Car Price:**

UPDATE Cars\_Details

SET Car\_Price =

CASE Car\_Number

WHEN 2578 THEN 1000000

WHEN 9258 THEN 900000

WHEN 8233 THEN 900000

WHEN 6214 THEN 1000000

END

WHERE Car\_Number IN (2578, 9258, 8233, 6214);

1. **SQL SELECT Statement with HAVING clause**

The HAVING clause in the SELECT statement creates a selection in those groups which are defined by the GROUP BY clause.

**Example of SELECT Statement with HAVING clause**

CREATE TABLE Orders (

Order\_ID INT,

Customer\_ID INT,

Order\_Date DATE,

Total\_Amount DECIMAL(10, 2)

);

INSERT INTO Orders (Order\_ID, Customer\_ID, Order\_Date, Total\_Amount)

VALUES

(1, 101, '2023-01-01', 500),

(2, 102, '2023-02-01', 1000),

(3, 103, '2023-03-01', 1500),

(4, 104, '2023-04-01', 2000),

(5, 105, '2023-05-01', 2500);

**SELECT Order\_ID, Customer\_ID, Order\_Date, Total\_Amount**

**FROM Orders**

**HAVING Total\_Amount > 1000;**

1. **SELECT Statement with ORDER BY clause**

**SELECT** \* **FROM** Orders **ORDER** **BY** Total\_Amount **DESC**;

**SELECT** \* **FROM** Orders **ORDER** **BY** Total\_Amount **ASC**;

1. **SELECT DISTNICT**

**SELECT** **DISTINCT** Address

**FROM** student\_records;

1. **SELECT COUNT**

**CREATE TABLE Bike\_Details (**

**Bike\_Name VARCHAR(50),**

**Bike\_Color VARCHAR(50),**

**Bike\_Cost DECIMAL(10, 2)**

**);**

**INSERT INTO Bike\_Details (Bike\_Name, Bike\_Color, Bike\_Cost)**

**VALUES**

**('Pulsar', 'Black', 185000),**

**('Apache', 'Black', NULL),**

**('KTM RC', 'Red', 900000),**

**('Royal Enfield', 'White', NULL),**

**('Livo', 'Black', 80000),**

**('KTM DUKE', 'Red', 195000);**

SELECT COUNT(DISTINCT Bike\_Color) AS Distinct\_Color\_Count

FROM Bike\_Details;

SELECT COUNT(Bike\_Color) AS TotalBikeColor FROM Bike\_Details;

SELECT COUNT(Bike\_Name) AS TotalBikeBlackColor FROM Bike\_Details WHERE Bike\_Color = 'Black';