1. CREATE DATABASE

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

Syntax:

CREATE DATABASE databasename;

Example:

CREATE DATABASE MCABatches;

1. CREATE TABLE

The create command is used to create a new table.

Syntax:

CREATE TABLE TableName (

Column1 datatype,

Column2 datatype,

. . . .

);

Example:

CREATE TABLE MCA23(

Roll\_No int,

FirstName varchar(50),

LastName varchar (50),

Email\_Id varchar (50),

Batch varchar (50)

);

1. Insert Into

The Insert Into query is used to insert new records in a table.

Syntax:

Insert Into TableName VALUES (value1, value2, . . .)

Example:

INSERT INTO MCA23 VALUES (11,'Atul','Kohli’, ‘atulkohli31@gmail.com', 'MCA-23');

INSERT INTO MCA23 VALUES (13,'Kapil','Sadotra’, ‘SadotraKapil@gmail.com', 'MCA-23');

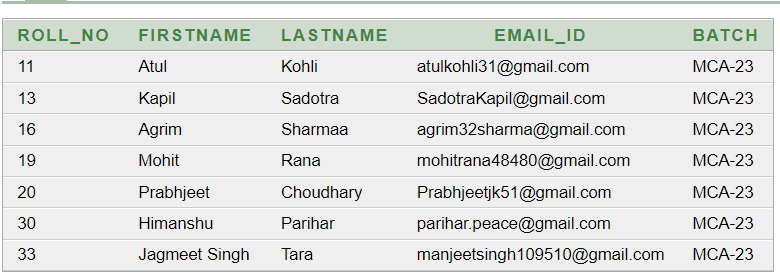
INSERT INTO MCA23 VALUES (16,'Agrim','Sharmaa’, ‘agrim32sharma@gmail.com', 'MCA-23');

INSERT INTO MCA23 VALUES (19,'Mohit','Rana’, ‘mohitrana48480@gmail.com', 'MCA-23');

INSERT INTO MCA23 VALUES (20,'Prabhjeet','Choudhary’, ‘Prabhjeetjk51@gmail.com', 'MCA-23');

INSERT INTO MCA23 VALUES (30,'Himanshu','Parihar’, ‘parihar.peace@gmail.com', 'MCA-23');

INSERT INTO MCA23 VALUES (33,'Jagmeet Singh’, ‘Tara’, ‘manjeetsingh109510@gmail.com', 'MCA-23');



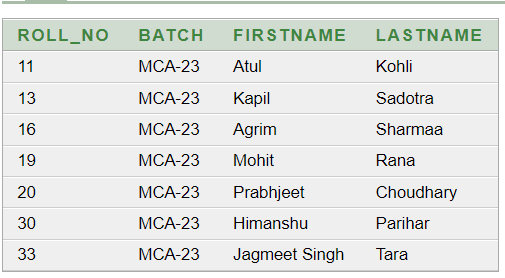
1. Select

The Select query is used to select data from a database.

Syntax:

Select \* From TableName;

Example:

Select Roll\_no, Batch, FirstName, LastName from mca23;

1. DROP

The Drop query is used to drop an existing table in a database.

Syntax:

DROP Table TableName;

Example:

DROP TABLE RegisteredStudents;

1. Truncate

The Truncate query deletes the data inside a table, but not the table itself.

Syntax:

TRUNCATE TABLE TableName;

Example:

TRUNCATE TABLE Users;

1. Order By

The Order By keyword sorts the result-set in ascending or descending order.

Syntax:

Select column1, column2, . . .

From TableName

Order By column1, column2, . . .

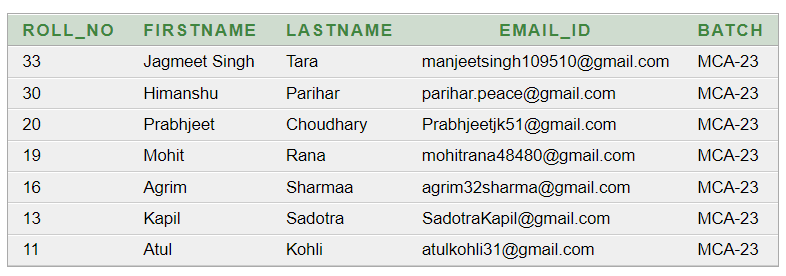
ASC | DESC;

Example:

SELECT \* FROM MCA23

ORDER BY Roll\_No

DESC;



1. WHERE

The Where clause is used to filter records.

Syntax:

Select column1, column2, . . .

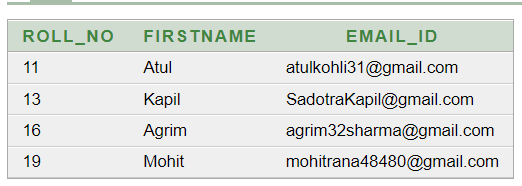
From TableName

Where condition;

Example:

SELECT Roll\_No, FirstName, Email\_Id FROM MCA23

WHERE Roll\_No < 20;



1. LIKE

The Like Operator is used in a WHERE clause to search for a specific pattern in a column.

Syntax:

SELECT column1, column2, . . .

FROM TableName;

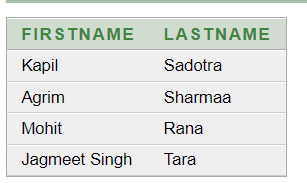
WHERE ColumnN LIKE pattern;

Example:

SELECT FirstName, LastName

FROM MCA23

WHERE LastName LIKE ‘%a’ ;



1. AND

The AND Operator is used to filter records based on one or more than one condition.

Syntax:

Select column1, column2, . . .

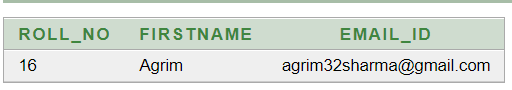
From TableName

Where condition1 AND condition2 AND condition3 . . .;

Example:

SELECT Roll\_No, FirstName, Email\_Id FROM MCA23

WHERE Roll\_No > 15 AND Roll\_No < 18;



1. OR

The OR command is used with WHERE to include rows where either condition is true.

Syntax:

Select column1, column2, . . .

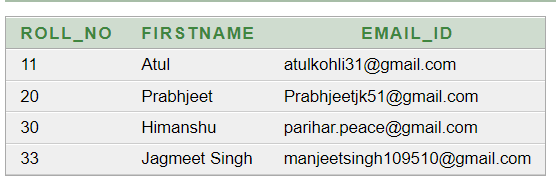
From TableName

Where condition1 OR condition2 OR condition3 . . .;

Example:

SELECT Roll\_No, FirstName, Email\_Id FROM MCA23

WHERE Roll\_No > 19 OR Roll\_No = 11;



1. NOT

The NOT command is used with Where to only include rows where a condition is not true.

Syntax:

Select column1, column2, . . .

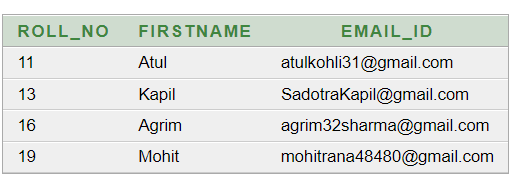
From TableName

Where NOT condition;

Example:

SELECT Roll\_No, FirstName, Email\_Id FROM MCA23

WHERE NOT Roll\_No > 19 AND NOT Roll\_No = 30;



1. BETWEEN

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

Syntax:

Select column1, column2, . . .

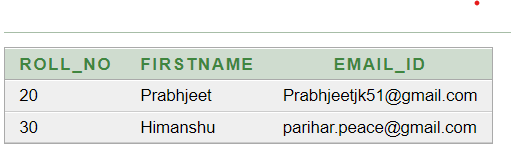
From TableName

Where condition1 BETWEEN condition2 AND condition3 . . .;

Example:

SELECT Roll\_No, FirstName, Email\_Id FROM MCA23

WHERE Roll\_No BETWEEN 20 AND 30;



1. DISTINCT

The Distinct statement is used to return only distinct values.

Syntax:

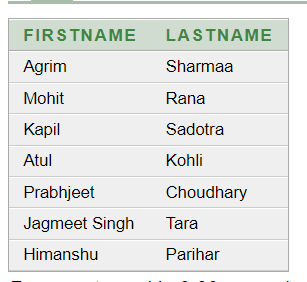
SELECT DISTINCT column1, column2, . . .

FROM table\_name;

Example:

SELECT DISTINCT FirstName, LastName

FROM MCA23;



1. GROUP BY

The GROUP BY Statement groups rows that have the same values.

This statement is often used with aggregate functions to group the result-set by one or more columns.

Syntax:

SELECT column\_name

FROM table\_name

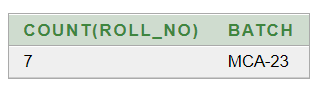
WHERE condition

GROUP BY column\_name;

Example:

SELECT COUNT(Roll\_No), Batch FROM MCA23

GROUP BY Batch;



1. MIN ()

The Min () Function returns the smallest value of the selected column.

Syntax:

SELECT MIN (column\_name)

FROM table\_name

WHERE condition;

Example:

SELECT MIN(Roll\_No) FROM MCA23;



1. MAX ()

The Max () Function returns the largest value of the selected column.

Syntax:

SELECT MAX (column\_name)

FROM table\_name

WHERE condition;

Example:

SELECT MAX(Roll\_No) FROM MCA23;



1. COUNT ()

The Count () Function returns the number of rows that matches a specified criterion.

Syntax:

SELECT COUNT (column\_name)

FROM table\_name

WHERE condition;

Example:

SELECT COUNT(Email\_Id) FROM MCA23;

1. AVG ()

The Avg () Function returns the average value of a numeric function.

Syntax:

SELECT AVG (column\_name)

FROM table\_name

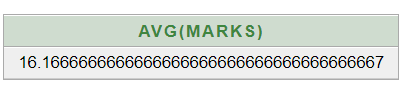
WHERE condition;

Example:

SELECT AVG(Marks)

FROM StudentMarks

WHERE Marks > 10;



1. SUM ()

The Sum () Function returns the total sum of a numeric column.

Syntax:

SELECT SUM (column\_name)

FROM table\_name

WHERE condition;

Example:

SELECT SUM(Marks)

FROM StudentMarks;



1. Alias

SQL Aliases are used to give a table, or a column in a table, a temporary name.

An Alias is created with the AS keyword.

Syntax:

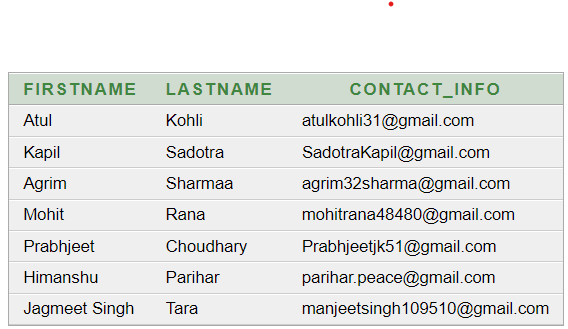
SELECT column\_name

FROM table\_name AS alias\_name;

Example:

SELECT FirstName, LastName, Email\_Id AS Contact\_Info FROM MCA23

where Roll\_No > 0;



1. Constraints

SQL Constraints are used to specify rules for data in a table.

1. NOT NULL

The NOT NULL constraint enforces a column to NOT accept NULL values.

Syntax:

Create Table TableName (

Column1 NOT NULL,

Column2, . . .);

Example:

CREATE TABLE MCA23(

Roll\_No int,

FirstName varchar (50) NOT NULL,

LastName varchar (50),

Email\_Id varchar (50),

Batch varchar (50)

);

1. UNIQUE

The UNIQUE constraint ensures that all values in a column are different.

Syntax:

Create Table TableName (

Column1 UNIQUE,

Column2, . . .);

Example:

CREATE TABLE MCA23(

Roll\_No int unique,

FirstName varchar(50),

LastName varchar (50),

Email\_Id varchar (50),

Batch varchar (50)

);

1. PRIMARY KEY

The PRIMARY KEY constraint uniquely identifies each record in a table.

Syntax:

Create Table TableName (

Column1 PRIMARY KEY,

Column2, . . .);

Example:

CREATE TABLE MCA23(

Roll\_No int PRIMARY KEY,

FirstName varchar (50),

LastName varchar (50),

Email\_Id varchar (50),

Batch varchar (50)

);

1. FOREIGN KEY

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

A FOREIGN KEY is a field in one table, that refers to the PRIMARY KEY in another table.

Syntax:

Create Table TableName (

Column1 NOT NULL,

Column2, . . .

FOREIN KEY (column\_name) REFERENCES Tablename (column\_name);

Example:

CREATE TABLE MCA-New-Batch (

Registration\_No int;

RNo int,

FirstName varchar (50),

LastName varchar (50),

Email\_Id varchar (50),

FOREIGN KEY (R\_No) REFERENCES MCA23 (Roll\_No)

);

1. CHECK

The CHECK constraint is used to limit the value range that can be placed in a column.

Syntax:

Create Table TableName (

Column1 NOT NULL,

Column2,

. . . . .

Column\_name CHECK (condition));

Example:

CREATE TABLE MCA-Admission (

FirstName varchar (50),

LastName varchar (50),

Email\_Id varchar (50),

Address varchar (50),

Age int,

CHECK (Age >= 18)

);

1. DEFAULT

The DEFAULT constraint is used to set a default value for a column.

Syntax:

Create Table TableName (

Column1 NOT NULL,

Column2. . .. .,

Column\_name Default value.);

Example:

CREATE TABLE MCA-Admission (

FirstName varchar (50),

LastName varchar (50),

Email\_Id varchar (50),

Address varchar (50),

State varchar (50) DEFAULT ‘Jammu & Kashmir’,

Age int,

CHECK (Age >= 18),

);

1. HAVING

The having clause is used with Aggregate Functions

Syntax:

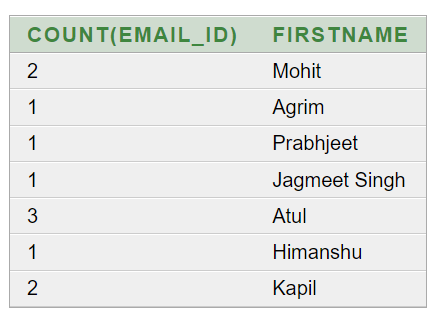
SELECT *column\_name(s)*  
FROM *table\_name*  
WHERE *condition*  
GROUP BY *column\_name(s)*HAVING *condition*ORDER BY *column\_name(s)*

Example:

SELECT COUNT(Email\_Id), FirstName FROM MCA23

GROUP BY FirstName

HAVING COUNT(Email\_Id) > 0;



1. VIEWS

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

Syntax:

CREATE VIEW view\_name AS

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

Example:

CREATE VIEW Batch2k23 AS

SELECT FirstName, LastName, Email\_Id

FROM MCA23;

1. SQL JOINS

A JOIN Clause is used to combine rows from two or more tables, based on a related column between them.

INNER JOIN

The INNER JOIN keyword selects records that have matching values in both tables.

Syntax:

SELECT column\_name

FROM Table1

IINER JOIN table2

ON table1.column\_name = table2.column\_name;

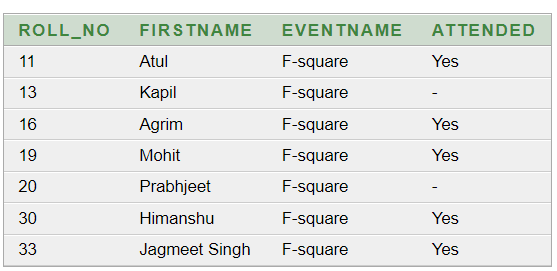
Example:

SELECT MCA23.Roll\_No, MCA23.FirstName, Event.EventName, Event. Attended

FROM MCA23

INNER JOIN EVENT

ON MCA23.Roll\_No = Event. Roll\_No;



LEFT OUTER JOIN

The LEFT OUTER JOIN keyword returns all records from the left table(table1) and the matching records from the right table(table2). The result is 0 records from the right side, if there is no match.

Syntax:

SELECT column\_name

FROM Table1

LEFT OUTER JOIN table2

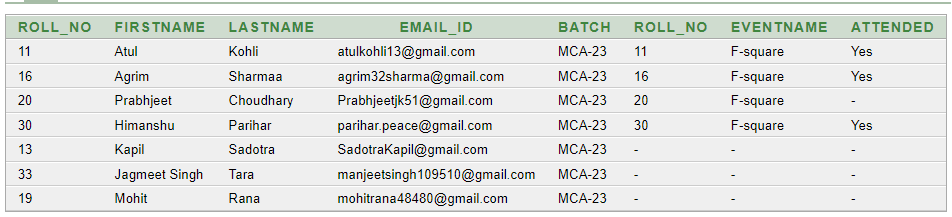
ON table1.column\_name = table2.column\_name;

Example:

SELECT \* FROM MCA23

LEFT OUTER JOIN EVENT

ON MCA23.Roll\_No = Event. Roll\_No;

 RIGHT OUTER JOIN

The RIGHT OUTER JOIN keyword returns all records from the right table(table2) and the matching records from the left table(table1). The result is 0 records from the left side, if there is no match.

Syntax:

SELECT column\_name

FROM Table1

RIGHT OUTER JOIN table2

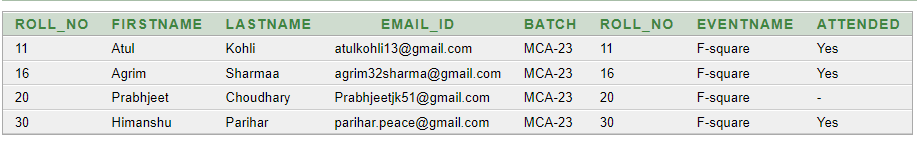
ON table1.column\_name = table2.column\_name;

Example:

SELECT \* FROM MCA23

LEFT OUTER JOIN EVENT

ON MCA23.Roll\_No = Event. Roll\_No;

FULL OUTER JOIN

The INNER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

Syntax:

SELECT column\_name

FROM Table1

FULL OUTER JOIN table2

ON table1.column\_name = table2.column\_name

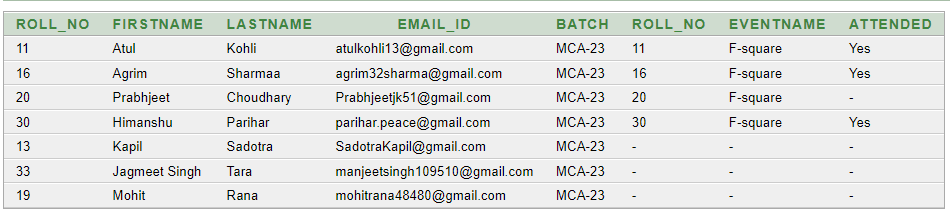
WHERE Condition;

Example:

SELECT \* FROM MCA23

FULL OUTER JOIN EVENT

ON MCA23.Roll\_No = Event. Roll\_No;



1. PRIVELEGES

A privilege is permission to execute one particular type of SQL statement or access a second person’s object. Database privilege controls the use of computing resources.

Syntax:

CREATE USER Username identified by “pwd”;

Example:

CREATE USER User1 identified by “password”;

1. Roles

Role is a named group of related privileges that can be granted to the user.

Syntax:

CREATE ROLE rolename;

Example:

CREATE ROLE students;