Constraint in MYSQL

* [NOT NULL](https://www.w3schools.com/mysql/mysql_notnull.asp) - Ensures that a column cannot have a NULL value
* [UNIQUE](https://www.w3schools.com/mysql/mysql_unique.asp) - Ensures that all values in a column are different
* [PRIMARY KEY](https://www.w3schools.com/mysql/mysql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* [FOREIGN KEY](https://www.w3schools.com/mysql/mysql_foreignkey.asp) - Prevents actions that would destroy links between tables
* [CHECK](https://www.w3schools.com/mysql/mysql_check.asp) - Ensures that the values in a column satisfies a specific condition
* [DEFAULT](https://www.w3schools.com/mysql/mysql_default.asp) - Sets a default value for a column if no value is specified
* [CREATE INDEX](https://www.w3schools.com/mysql/mysql_create_index.asp) - Used to create and retrieve data from the database very quickly

Command To Create DataBase:

1)CREATE DATABASE *databasename*;

Ex:- CREATE DATABASE Ecommerce;

Command To Drop DataBase:

1) DROP DATABASE *databasename*;

Ex:- DROP DATABASE Ecommerce;

Command To Create Table:

1) CREATE TABLE *table\_name*(  
*datatype*,  
*column2 datatype     column1 e*,  
*column3 datatype*,  
   ....  
);

Ex:- CREATE TABLE Users (  
    UserID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);

Ex:- CREATE TABLE customers

(

CustID int8 PRIMARY KEY,

CustName varchar(50) NOT NULL,

Age int NOT NULL,

City char(50),

Salary numeric

);

Create Table Using Another Table

2) CREATE TABLE *new\_table\_name* AS  
    SELECT *column1, column2,...*  
    FROM *existing\_table\_name*  
    WHERE ....;

Ex:- CREATE TABLE TestTable AS  
SELECT FirstName, LastName  
FROM Users;

Create Table With All Constraints

Ex:-

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    UNIQUE (ID)  
);

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (ID)  
);

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CHECK (Age>=18)  
);

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255) DEFAULT 'Sandnes'  
);

CREATE INDEX idx\_pname  
ON Persons (LastName, FirstName);

CREATE TABLE Persons (  
    Personid int NOT NULL AUTO\_INCREMENT,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    PRIMARY KEY (Personid)  
);

Command To Drop Table:

1) DROP TABLE *table\_name*;

Ex:- DROP TABLE TestTable

2) TRUNCATE TABLE *table\_name*;

Ex:- TRUNCATE TABLE TestTable

Command To Alter Table:

1) ALTER TABLE *table\_name*  
 ADD *column\_name datatype*;

Ex:- ALTER TABLE customers

Add Email varchar(255);

2) ALTER TABLE *table\_name*  
 ALTER/MODIFY COLUMN *column\_name datatype*;

Ex:- ALTER TABLE customers

ALTER/MODIFY COLUMN Salary varchar(50)

3) ALTER TABLE *table\_name*  
 DROP *column\_name*;

Ex:- ALTER TABLE customers

DROP COLUMN City

Command To INSERT Into Table:

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

Ex:- INSERT INTO Customer

(CustID, CustName, Age, City, Salary)

VALUES

(1, ‘Sam’, 26, ‘Delhi’, 9000),

(2, ‘Ram’, 19, ‘Bangalore’, 11000),

(3, ‘Pam’, 31, ‘Mumbai’, 6000),

(4, ‘Jam’, 42, ‘Pune’, 10000);

2) INSERT INTO *table\_name*  
VALUES (*value1*,*value2*,*value3*, ...);

Ex:- INSERT INTO Customer

VALUES

(1, ‘Sam’, 26, ‘Delhi’, 9000),

(2, ‘Ram’, 19, ‘Bangalore’, 11000),

(3, ‘Pam’, 31, ‘Mumbai’, 6000),

(4, ‘Jam’, 42, ‘Pune’, 10000);

Command To UPDATE Into Table:

1) UPDATE *table\_name*  
SET *column1*=*value1*,*column2*=*value2*, ...  
WHERE *condition*;

Ex:- UPDATE customer

SET CustName = 'Xam’, Age= 32

WHERE CustID = 4;

Command To DELETE From Table:

1) DELETE FROM *table\_name*WHERE *condition*;

Ex:- DELETE FROM Customers Where CustID = ‘2’;

2) DELETE FROM *table\_name*;

Ex:- DELETE FROM Customers;

Command To Select From Table:

1) SELECT \* FROM *table\_name*;

Ex:- Select \* FROM Customers;

2) SELECT *column1*,*column2, ...*  
 FROM *table\_name*;

Ex:- Select Age,City From Customers;

3) SELECT DISTINCT *column1*,*column2, ...*  
 FROM *table\_name*;

Ex:- SELECT DISTINCT Salary From Customers;

4) SELECT *column1*,*column2, ...*  
FROM *table\_name*  
WHERE *condition*;

Ex:- Select Salary From Customers Where CustID=1;

Operators in where condition

|  |  |  |
| --- | --- | --- |
| = | Equal |  |
| > | Greater than |  |
| < | Less than |  |
| >= | Greater than or equal |  |
| <= | Less than or equal |  |
| <> | Not equal. **Note:** In some versions of SQL this operator may be written as != |  |
| BETWEEN | Between a certain range |  |
| LIKE | Search for a pattern |  |
| IN | To specify multiple possible values for a column |  |

5) SELECT *column1*,*column2, ...*  
FROM *table\_name*  
WHERE *condition1* AND *condition2* AND *condition3 ...*;

Ex:- SELECT \* FROM Customers

WHERE Country = ‘Japan AND City = ‘Tokyo’;

6) SELECT *column1*,*column2, ...*  
FROM *table\_name*  
WHERE *condition1* OR *condition2* OR *condition3 ...*;

Ex:- SELECT \* FROM Customers

WHERE Country = ‘Japan’ OR City = ‘Tokyo’;

7) SELECT *column1*,*column2, ...*  
FROM *table\_name*  
WHERE NOT *condition*;

Ex:- SELECT \* FROM Customers

WHERE NOT Country = ‘Japan’;

Mixing All Arguments

Ex:- SELECT \* FROM Customers

WHERE Country = ‘Japan’ AND (City = ‘Tokyo’ OR City = ‘Nagasaki’

Ex:- SELECT \* FROM Customers

WHERE NOT Country = ‘Japan’ AND NOT Country = ‘USA’;

8) SELECT *column1*,*column2, ...*  
FROM *table\_name*  
ORDER BY *column1, column2, ...*ASC|DESC;

Ex:- SELECT \* FROM Customers

ORDER By Country DESC;

Ex:- SELECT \* FROM Customers

ORDER By Country , City ASC;

9) SELECT *column\_name(s)*  
FROM *table\_name*WHERE *condition*  
LIMIT *number*;

Ex:- SELECT \* FROM Customers

WHERE Country = ‘Tokyo’

LIMIT 3;

10) SELECT column\_name(s)

FROM table\_name

WHERE condition(s)

GROUP BY column\_name(s)

HAVING  condition(s)

ORDER BY *column\_name(s);*

Ex:- SELECT COUNT(CustomerID),Country

FROM customers

GROUP BY country

HAVING COUNT(CustomerID) >= 3

ORDER BY COUNT(CustomerID) DESC;

Ex:- SELECT City, COUNT(City) AS total

FROM Customers

GROUP BY City

HAVING COUNT(City) >= 3

ORDER BY total DESC

Command To Show Time:

• SHOW TIMEZONE

• SELECT NOW()

• SELECT TIMEOFDAY()

• SELECT CURRENT\_TIME

• SELECT CURRENT\_DATE

Command To Extract Specific Thing:

1) SELECT EXTRACT(MONTH FROM date\_field) FROM Table;

Ex:- SELECT EXTRACT (YEAR FROM payment\_date) AS pay\_year,payment\_date FROM payment;

Command To Join Tables:

1) SELECT column\_name(s)

FROM TableA

INNER JOIN TableB

ON TableA.col\_name = TableB.col\_name;

Ex:- SELECT \*

FROM customer AS c

INNER JOIN payment AS p

ON c.customer\_id = p.customer\_id

2) SELECT column\_name(s)

FROM TableA

LEFT JOIN TableB

ON TableA.col\_name = TableB.col\_name;

Ex:- SELECT \*

FROM customer AS c

LEFT JOIN payment AS p

ON c.customer\_id = p.customer\_id

3) SELECT column\_name(s)

FROM TableA

RIGHT JOIN TableB

ON TableA.col\_name = TableB.col\_name;

Ex:- SELECT \*

FROM customer AS c

RIGHT JOIN payment AS p

ON c.customer\_id = p.customer\_id

4) SELECT column\_name(s)

FROM TableA

FULL OUTER JOIN TableB

ON TableA.col\_name = TableB.col\_name;

Ex:- SELECT \*

FROM customer AS c

FULL OUTER JOIN payment AS p

ON c.customer\_id = p.customer\_id

5) SELECT column\_name(s)

FROM Table AS T1

JOIN Table AS T2

ON T1.col\_name = T2.col\_name

Ex:- SELECT T2.empname, T1.empname

FROM emp AS T1

JOIN emp AS T2

ON T1.empid = T2.manager\_id

• INNER JOIN: Returns records that have matching values in both tables

• LEFT JOIN: Returns all records from the left table, and the matched

records from the right table

• RIGHT JOIN: Returns all records from the right table, and the matched

records from the left table

• FULL JOIN: Returns all records when there is a match in either left or

right table

Command To UNION Tables:

1) SELECT column\_name(s) FROM TableA

UNION

SELECT column\_name(s) FROM TableB

Ex:- SELECT cust\_name, cust\_amount from custA

UNION

SELECT cust\_name, cust\_amount from custB

2) SELECT column\_name(s) FROM TableA

UNION ALL

SELECT column\_name(s) FROM TableB

Ex:- SELECT cust\_name, cust\_amount from custA

UNION ALL

SELECT cust\_name, cust\_amount from custB

Command For SubQuery:

1)SELECT column\_name(s)

FROM table\_name

WHERE column\_name operator

( SELECT column\_name FROM table\_name WHERE ... );

Ex:- SQL> SELECT \*

FROM CUSTOMERS

WHERE ID IN (SELECT ID

FROM CUSTOMERS

WHERE SALARY > 4500) ;

Command For In/SELECT:

1) SELECT *column\_name(s)*  
FROM *table\_name*  
WHERE *column\_name* IN (*value1*,*value2*, ...);

Ex:- SELECT \* FROM Customers  
WHERE Country IN ('Germany', 'France', 'UK');

2) SELECT *column\_name(s)*  
FROM *table\_name*  
WHERE *column\_name* IN (*SELECT STATEMENT*);

Ex:- SELECT \* FROM Customers  
WHERE Country IN (SELECT Country FROM Suppliers);

Command For LIKE Query:

1) SELECT *column1, column2, ...*  
FROM *table\_name*  
WHERE *columnN* LIKE *pattern*;

Ex:- SELECT \* FROM Customers  
WHERE CustomerName LIKE '%or%';

Command For OFFSET Query:

1) SELECT column\_name(s)

FROM table\_name

WHERE condition

ORDER BY column\_name

OFFSET rows\_to\_skip ROWS;

Ex:- SELECT Fname, Lname

FROM Customers

ORDER BY Salary

OFFSET 1 ROWS;

2) SELECT column\_name(s)

FROM table\_name

ORDER BY column\_name

OFFSET rows\_to\_skip

FETCH NEXT number\_of\_rows ROWS ONLY;

Ex:- SELECT Fname, Lname

FROM Customers

ORDER BY Salary

OFFSET 2 ROWS

FETCH NEXT 4 ROWS ONLY;

3) LEAD(new\_id, 2) OVER( ORDER BY new\_id) AS "LEAD\_by2",

LAG(new\_id, 2) OVER( ORDER BY new\_id) AS "LAG\_by2"

Command For VIEW Query:

1) CREATE VIEW *view\_name* AS  
SELECT *column1*, *column2*, ...  
FROM *table\_name*  
WHERE *condition*;

EX:- CREATE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'Brazil';

Command For CASE Query:

1) CASE

WHEN condition1 THEN result1

WHEN condition2 THEN result2

WHEN conditionN THEN resultN

ELSE other\_result

END;

Ex:- SELECT customer\_id, amount,

CASE

WHEN amount > 100 THEN 'Expensive product'

WHEN amount = 100 THEN 'Moderate product'

ELSE 'Inexpensive product'

END AS ProductStatus

FROM Customers

2) CASE Expression

WHEN value1 THEN result1

WHEN value2 THEN result2

WHEN valueN THEN resultN

ELSE other\_result

END;

Ex:- SELECT customer\_id,

CASE amount

WHEN 500 THEN 'Prime Customer'

WHEN 100 THEN 'Plus Customer'

ELSE 'Regular Customer'

END AS CustomerStatus

FROM Customers