Mysql documentation:

* What is RDBMS ?
* RDBMS stands for Relational Database Management System.
* RDBMS uses SQL queries to access the data in the database

What is database table ?

* A table is a collection of related data entities and it’s consists of columns & rows.
* A column hold specific information about every record in the table.
* A row is each individual entry that exists in a table.

What is relational database ?

* Relational database defines database relationships in the form of tables.

What is SQL ?

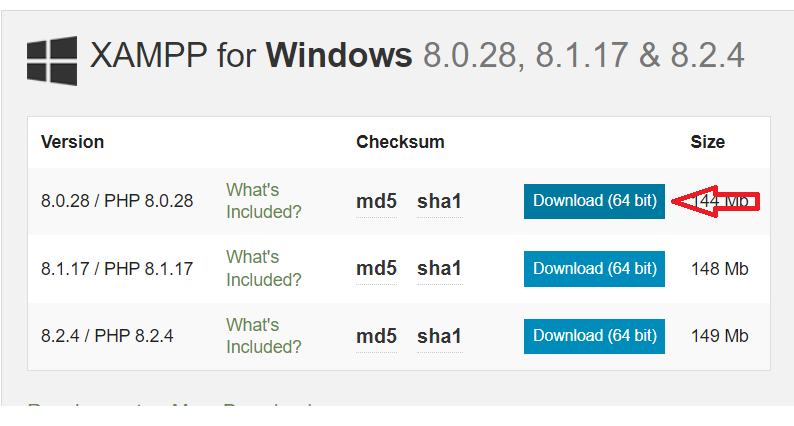
SQL is standard query language to insert , update , select & delete the data from the tables.

Ex:

Select \* from users;

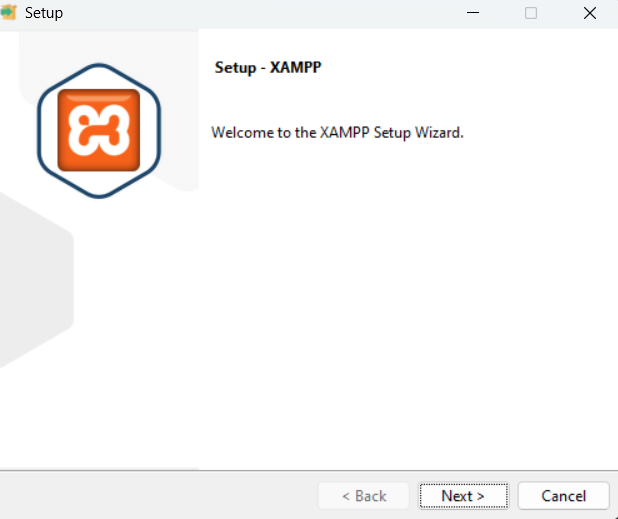
Installation of XAMPP.

url : <https://www.apachefriends.org/download.html>

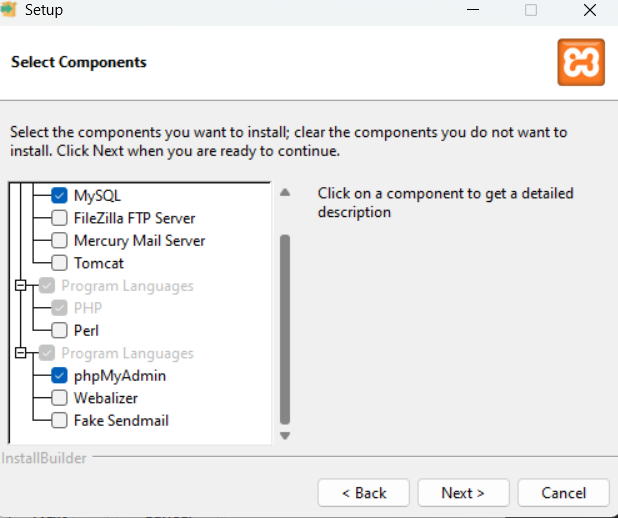


Click to download.

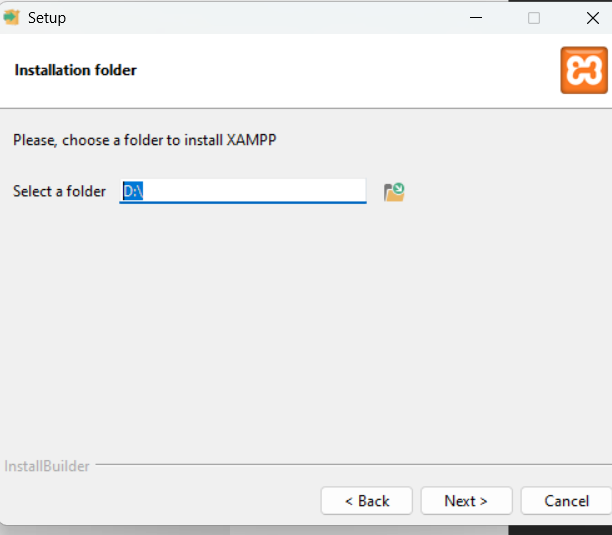
Double click the software and then click on “Yes”.



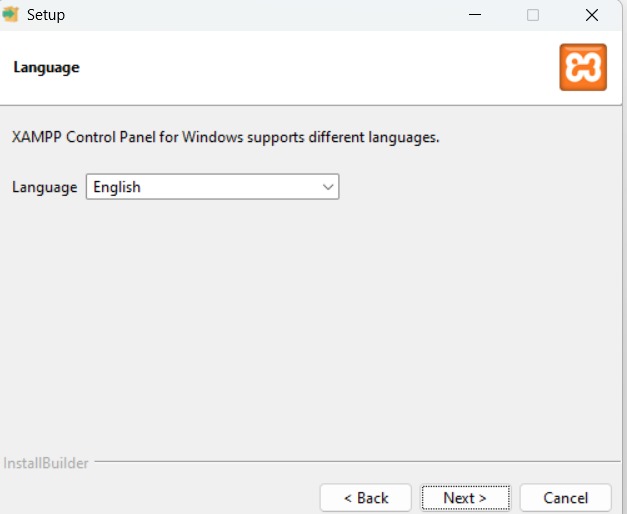
Click “Next”



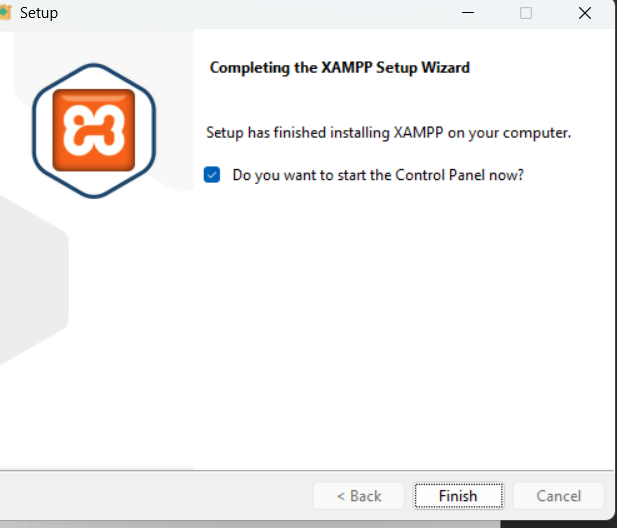
Select options as shown in the above picture. Click Next



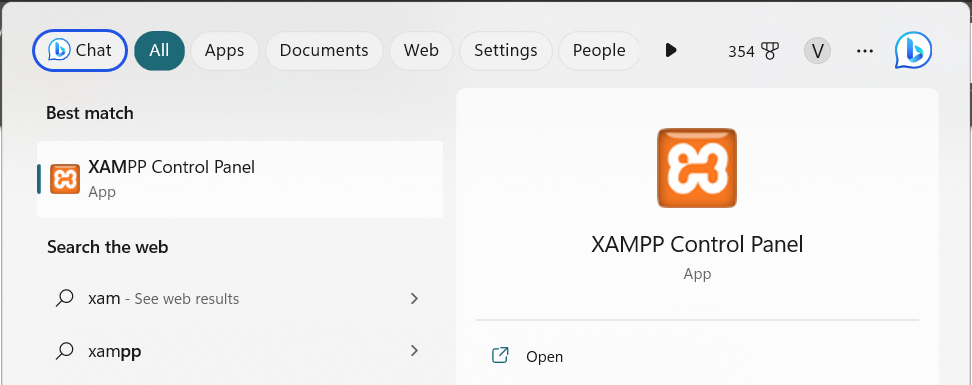
Select the folder path and click on “Next”



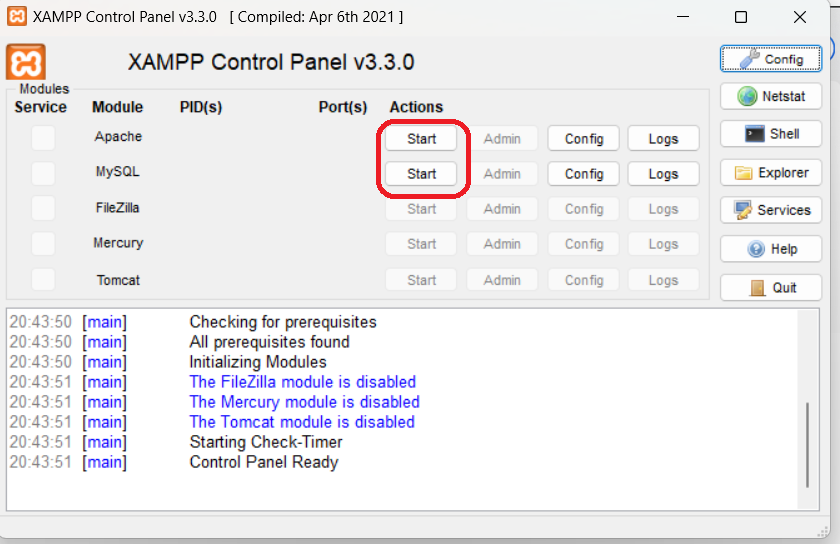
Choose English Language, click Next.



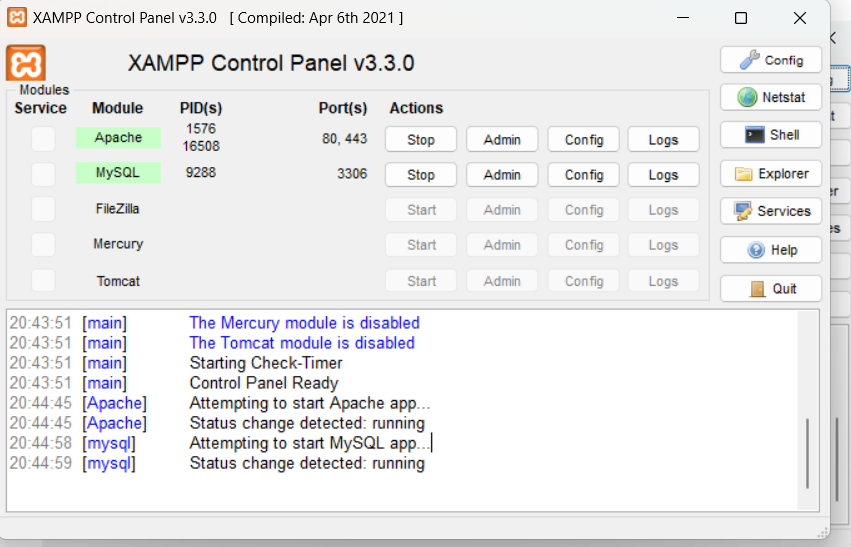
How to start Apache & Mysql server.



Click on XAMPP Control Panel



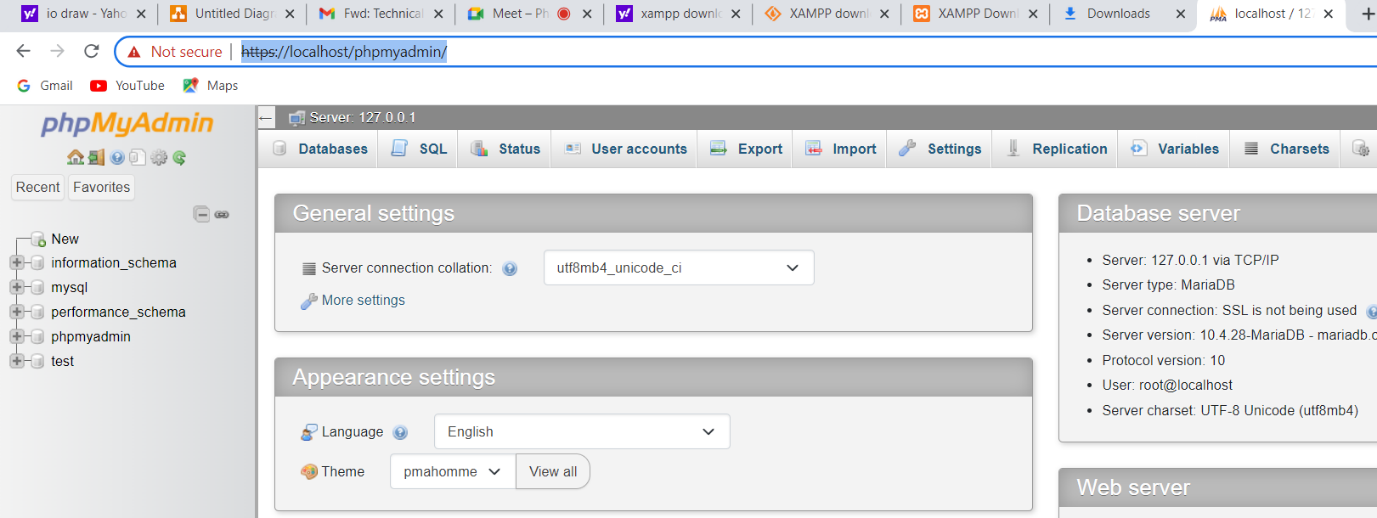
Click on both Start buttons. Finally you should get the below confirmation messages.



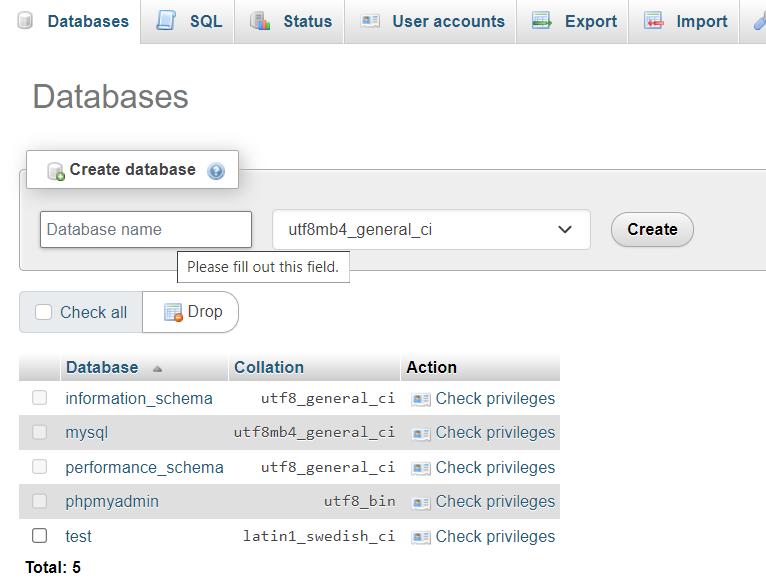
How to access MySQL ( Phpmyadmin ).

<https://localhost/phpmyadmin/>

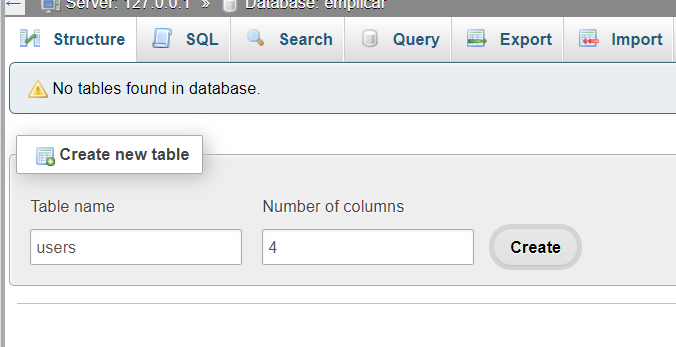
Paste the above url in your web browser, then you will get the below page.



How to create a database ?



How to create a table ?



As per the above screenshot, we are going to create Users table with 4 columns i.e,,

Id, username, created, modified.

How to fetch the data from the table ?

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users`

INSERT INTO `users` (`id`, `username`, `created`, `modified`) VALUES (NULL, 'Supriya', current\_timestamp(), current\_timestamp());

[UPDATE](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) `users` [SET](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) `username` = 'Supriyaedited' WHERE `users`.`id` = 3;

DELETE FROM users WHERE `users`.`id` = 2"

The above statements are called SQL queries.

Advantages of MySQL.

* It is RDBMS
* It is opensource
* It is computability for both small & larger applications
* Easy to learn.
* User friendly.

My SQL datatypes.

INT : It supports only numbers with 11 digits.

VARCHAR : It supports only strings up to 255 characters

TEXT : It supports only strings up to 8000 characters

DATE : It supports only Date.

DATETIME: It supports Date & Time.

BIGINT: It support more than 11 digit number.

TINYINT : It supports only 0 or 1 ( True / False ).

BOOLEAN: It supports True / False and it is deprecated. Use TINYINT instead.

DOUBLE: It supports float values with pre-defined decimal points. ( 10.258974)

FLOAT : It support float with 2 decimal points ( 10.25).

CHAR : It supports only strings.

BLOB: It supports to store base64 encoded data.

ENUM: It supports only pre-defined key value pair.

Char vs Varchar difference:

Ex: comments field

Comments char(150).

The value is “Test comment” - > Complete 150 character memory allocated.

Ex:

Comments filed.

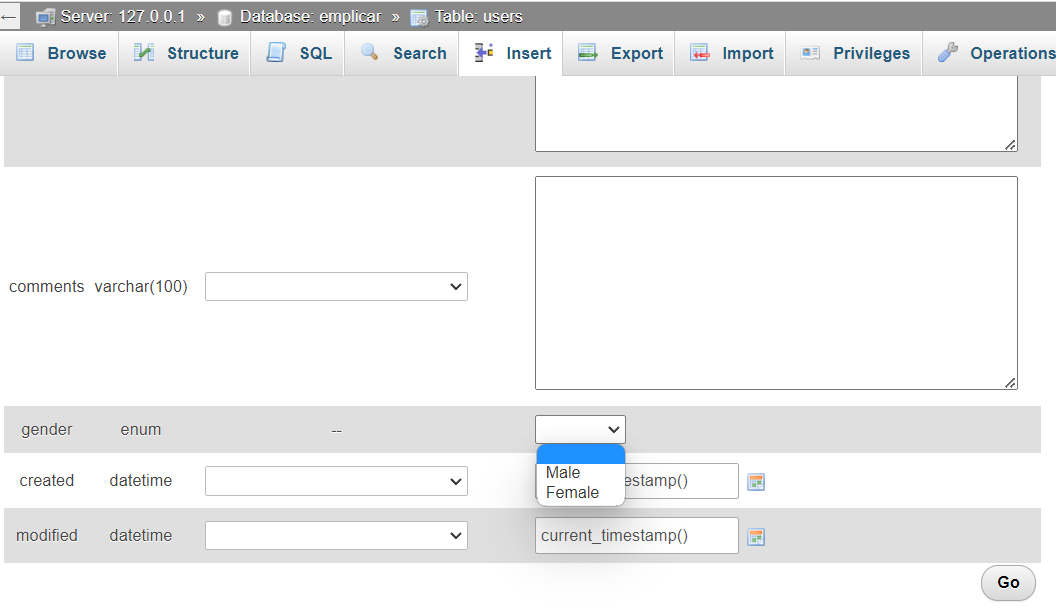
Comment VARCHAR(150)

The value is “Test”, this time memory allocated only 4 characters and remaining will be released.

BLOB:

To store any data into binary format you can use BLOB data type.

**Enum Screenshot.**



# Where Condition:

SELECT column1 , column2 …….

FROM table

WHERE condition.

Ex:

SELECT \* FROM `users` where username = 'Vasu'

Operator Description

= Equal

>= Greater than or equal

> Greater than

<= Less than or equal

< Less than

BETWEEN Range

EX: [SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM users where sal BETWEEN 100 [AND](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) 200;

LIKE Search for a pattern.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` where username [like](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html%23operator_like) ('%dev');

IN Specify multiple values.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` where id [in](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) (1,6);

AND condition.

SELECT column1 , column2 …….

FROM table

WHERE condition

And condition.

EX: [SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` where id [in](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) (1,6) [AND](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) sal >= 500;

1 OR condition.

SELECT column1 , column2 …….

FROM table

WHERE condition

OR condition

SELECT \* FROM `users` WHERE username like ('%dev%') OR ( sal BETWEEN 100 AND 150 );

ORDER BY : The order by keyword is used to sort the results either ascending or descending. By default order by always ascending order.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` order by username;

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` order by username desc;

# How to insert data into the table.

Syntax:

INSERT INTO <table Name> ( column1 , column2 ….)

Values ( column1 , column2 ).

INSERT INTO `users` (`id`, `username`, `created`, `modified`) VALUES (NULL, 'Supriya', current\_timestamp(), current\_timestamp());

# Handle NULL values

How to filter all NULL value records.

To fetch all the NULL value records , we must use IS NULL instead of = , >= <=.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` WHERE grade [IS](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23operator_is) NULL;

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` Where grade [IS](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23operator_is) [NOT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_not) NULL;

# How to use UPDATE statement.

Update < Table Name >

SET

<column1 name> = Value,

<column2 name> = Value

WHERE

<column name> = Value

Ex:

[UPDATE](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) `users` [SET](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/set.html) username = 'Vasudev\_edited', sal = 1000 where id = 1;

# How to use DELETE statement.

Syntax:

DELETE FROM <table name> WHERE condition.

Ex:

DELETE FROM users where id = 1;

It will delete the record which is belongs to id = 1;

If we did not provide where condition, entire table data will be deleted.

# How to user LIMIT statement

To fetch limited number of records instead of all records.

Syntax:

select \* from <TABLE NAME) LIMIT 5;

To fetch next 5 records.

Select \* from <table name> limit 6,5;

Means , from 6th record to next 5 records.

# How to use Min , Max methods in mySQL

Ex:

SELECT max(sal) FROM users;

SELECT min(sal) FROM users;

# How to use COUNT , AVG & SUM in mySQL

Count:

Syntax : SELECT COUNT(\*) FROM <table> WHERE condition.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [count](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(\*) FROM `users`;

It will return count of all the records.

Avg:

To find the user salary average.

Syntax : SELECT AVG(<column>) RROM <table>

Avg means sum / no of records.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [AVG](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_avg)(sal) FROM `users`;

Sum:

To find the sum of the values.

Syntax : SELECT sum(<column>) FROM <table>

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [sum](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_sum)(sal) FROM `users`;

# How to use LIKE operator in MySQL

To find the records which are matched with search pattern.

Syntax:

SELECT \* FROM <tablename> WHERE <column> LIKE <value>

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` WHERE username [like](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html%23operator_like) ('%Vb%');

# How to user IN operator in MySQL

This operator allows to specify multiple values in the WHERE clause.

Syntax :

SELECT \* FROM <table> WHERE <column> IN (<values>)

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` WHERE id [IN](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) (16,21);

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` WHERE username [IN](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_in) ('Suresh' , 'Supriya');

# How to use BETWEEN operator.

Between operator can user to find the records of a particular range.

Syntax:

SELECT \* FROM <tablename> BETWEEN <range1> AND <range2>

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` WHERE sal BETWEEN 100 [AND](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) 200;

# **JOINS**

There are 4 type of Joins.

1. Inner Join ( Join )
2. Left Join
3. Right Join
4. Cross Join
5. Self Join.

## Inner Join:

Inner join to retrieve the similar values from both the tables.

Syntax:

SELECT

<columns>

FROM

<table1>

JOIN

<table2>

ON

<table1.column> = <table2.column>

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM users JOIN user\_grades ON users.id = user\_grades.user\_id;

## LEFT JOIN

The left join returns all the records from the left table and matched records from the right table ( if anything matches).

Syntax:

SELECT

<columns>

FROM

<table1>

LEFT JOIN

<table2>

ON

<table1.column> = <table2.column>

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM users [left](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_right) JOIN user\_grades ON users.id = user\_grades.user\_id;

## RIGHT JOIN

The right join returns all the records from the right table and matched records from the left table ( if anything matches).

Syntax:

SELECT

<columns>

FROM

<table1>

RIGHT JOIN

<table2>

ON

<table1.column> = <table2.column>

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM users [RIGHT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_right) JOIN user\_grades ON users.id = user\_grades.user\_id;

Cross Join:

It will return all the records from table1 and table2.

Syntax:

SELECT <columns>

FROM <table1>

CROSS JOIN <table2>

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users` cross join documents;

Self Join:

The table join with itself.

Syntax:

SELECT <column>

From <table1> t1 , <table1> t2

WHERE condition.

EX:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) cat1.categoryName, cat2.categoryName FROM `categories` cat1, `categories` cat2 WHERE cat1.id = cat2.parent\_id;

# UNION OPERATOR :

Used to combine the results set of two or more.

Syntax:

SELECT <column names> FROM table1

UNION ALL

SELECT <column names> FROM table2.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `andhra` UNION ALL [SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM telangana;

# GROUP BY

Group by provides group of rows like., find the number of customer in each country or find the sum of the salaries for the employees.

SELECT

<aggregate function> <column name > FROM TABLE GROUP BY <column name>

aggregate function -> Sum , Avg & Count.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) employeeId , emploeename , [SUM](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_sum)(salary) FROM `salaries` GROUP BY employeeId , emploeename;

# HAVING CLAUSE

The having used when SQL query contains any aggregate function in place of WHERE clause.

Syntax

SELECT

<aggregate function> <column name > FROM TABLE GROUP BY <column name>

HAVING <column name> condition.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) emploeename , [SUM](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_sum)(salary) FROM `salaries` GROUP BY emploeename HAVING emploeename = 'Firu';

# EXISTS OPERATOR

This operator to check whether record is exists or not in the sub-query.

Syntax:

SELECT <column name> FROM <table> WHERE EXISTS ( SELECT <columnname> FROM <tablename> WHERE <condition> );

Ex:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) product\_name FROM products WHERE EXISTS ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) categoryName FROM categories WHERE categories.id = products.category\_id [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) categories.id = 1);

# INSERT INTO SELECT

This will useful to copy the data from one table another table.

Syntax:

INSERT into <tablename>

SELECT <columns> FROM tablename.

Ex:

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html) INTO `salaries\_march` (employeeId , emploeename , month, salary) [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) employeeId , emploeename , month, salary FROM salaries WHERE month = 'Mar';

# CASE WHEN

Syntax:

CASE

WHEN condition1 THEN result1

WHEN condition2 THEN result2

ELSE result

END as ``

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [CASE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/control-flow-functions.html%23operator_case) WHEN (employeeId = 1001) THEN salary \* 2 WHEN (employeeId = 1002) THEN salary \* 3 END as 'revised\_salary' FROM salaries\_march;

# How to create a database from the Query.

Syntax : CREATE DATABASE <databaseName>

Ex:

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-database.html) [DATABASE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-database.html) emplicar2;

# How to drop a database from the Query.

Syntax : DROP DATABASE <databaeName>

EX:

DROP DATABASE emplicar2;

# How to create a table from the Query

Syntax:

CREATE TABLE <tableName> ( <columns> ).

Ex:

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) salesforce ( id int );

# How to drop a table from the query

# Syntax:

# DROP TABLE <tableName>

# Ex:

[DROP](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/drop-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/drop-table.html) salesforce;

# How to alter a table from the query

Syntax:

ALTER TABLE <tablename> ADD <column> datatype.

Ex:

[ALTER](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) `salaries` ADD years int;

# Constraints

Constraints are used to defined a specific rules for the data of the table.

* NOT NULL
  + Make sure that column data can not have NULL value.

Ex:

INSERT INTO `notnull` (`username`) VALUES (NULL);

# UNIQUE Constraint.

Column contains all values are different.

How to set unique key.

[ALTER](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) `notnull` ADD UNIQUE(`grade`);

# Difference between Primary Key & Unique Key

Both keys are not allowed duplicate values but unique key will accept NULL values.

# Primary Key

Combination of Unique value and Not null. Uniquely identify each row of the table.

# Foreign Key

To build the relations between two ore more tables.

[ALTER](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) [TABLE](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) `products` ADD FOREIGN KEY (`category\_id`) REFERENCES `products`(`id`) ON [DELETE](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) RESTRICT ON [UPDATE](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html) RESTRICT;

# Define Variables in MySQL

Syntax : SET @<variableName> = ‘’;

Ex:

SET @a = 10;

SET @b = 20;

SELECT @a+@b;

# MySQL Concat:

SET @firstName = 'Vasudev';

SET @lastName = 'Sarvepalli';

SELECT CONCAT(@firstName , " ", @lastName);

# Mysql If Expression.

SET @a = 10;

SET @b = 20;

SELECT IF(@a > @b , 'a is greater' , 'b is greater');

# IFNULL Function

This function can be used to replace the NULL values in the select statement.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) id , ifnull(grade, '1') FROM `notnull`;

# CHAR\_LENGTH FUNCTION

To find the character length of a string.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) CHAR\_LENGTH("Emplicar");

# CONCAT-WS

This function is used to concatenating the strings with a specified delimiter.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) CONCAT\_WS("==", "Welcome" , "To", "Emplicar");

SELECT CONCAT\_WS("\*\*", "Welcome" , "To", "Emplicar");

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) CONCAT("Welcome", "==", "to" , "==", "Emplicar");

# LENGTH FUNCTION

To find the length of the string. User this method instead of Char\_length.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) LENGTH("Welcome to emplicar");

# LOCATE:

It returns the position of the first occurrence of the substring in the main string.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) LOCATE("to" , "Welcome to emplicar");

# LTRIM

This method is used to remove the left side white spaces of string.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) LTRIM(" Welcome");

# RTRIM

This method is used to remove the right side white spaces of a string.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) RTRIM("Welcome ");

# TRIM

This method is used to remove white spaces from both the sides of the string.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) TRIM(" Welcome ");

# REPEAT

This method is used to display the string repeatedly with a given count.

Ex:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [REPEAT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_repeat)('Welcome', '5');

# REPLACE

This method is used to replace the string in the main string.

Ex:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [REPLACE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/replace.html)('Welcome to Ifyn', 'Ifyn', 'Emplicar');

# REVERSE

This method is used to reverse the main string.

Ex:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) REVERSE('BYE');

# RIGHT

This method will returns right most length characters from the string.

Ex:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [RIGHT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_right)('Bye to Nellore', '5');

# LEFT

This method will returns left most length characters from the string.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [LEFT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_left)(created, 4) as created\_year FROM `users`;

# SUBSTR

This method will return sub string from main string.

Ex:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) SUBSTR('Welcome', 2,3);

# STRCMP

This method used to compare two given strings. It will return ‘0’ if both the strings are equal and return -1 if string1 is less than string2 and return 1 if string1 greater than string2.

Ex:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) STRCMP('abc', 'ABCD');

# CURRENT DATE

This method will return current date in ‘YYYY-mm-dd’ format.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) CURDATE();

# NOW

This method will return current date and time in ‘YYYY-mm-dd h:i:s’

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) NOW();

# DATE

This method is used to extract the date from datetime string.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE('2023-07-03 20:58:55');

# DATEDIFF

This method will return the days between the two dates.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATEDIFF('2023-07-05', '2023-07-01');

1st Parameter date should be greater than 2nd Parameter date.

# DATE\_ADD

This method will be used to add one or more Day, Hour , Year, Month.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_ADD('2023-07-05', [INTERVAL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_interval) 1 DAY);

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_ADD('2023-07-05', [INTERVAL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_interval) 1 MONTH);

SELECT DATE\_ADD('2023-07-05', INTERVAL 1 YEAR);

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_ADD('2023-07-05 20:30:00', [INTERVAL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_interval) 1 MINUTE);

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_ADD('2023-07-05 20:30:00', [INTERVAL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_interval) 1 HOUR);

# DATE\_SUB

This method will be used to deduct one or more Days , Hours, Years, Months, Minutes.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_SUB('2023-07-05', [INTERVAL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_interval) 1 DAY);

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_SUB('2023-07-05', [INTERVAL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_interval) 1 MONTH);

SELECT DATE\_SUB('2023-07-05', INTERVAL 1 YEAR);

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_SUB('2023-07-05 20:30:00', [INTERVAL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_interval) 1 MINUTE);

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_SUB('2023-07-05 20:30:00', [INTERVAL](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html%23function_interval) 1 HOUR);

# DATE FORMAT

|  |  |
| --- | --- |
| Format Type | Format Value |
| %a | It will return week day name.  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%a');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%a'); |
| %M | It will return month name  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%M');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%M'); |
| %c | It will return month name in numeric format  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%c');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%c'); |
| %d | It will return day of the month in numeric format. ( 00 – 30)  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%d');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%d'); |
| %D | It will return day of the month in numeric format with English suffix  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%D');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%D'); |
| %e | It will return month in numeric format. ( 1-30)  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%e');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%e'); |
| %h | It will return hours in numeric format. ( 12 hour format)  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%h');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%h'); |
| %H | It will return hours in numeric format. ( 24 hour format)  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%H');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%H'); |
| %i | It will return minutes in numeric format.  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%i');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%i'); |
| %j | It will return day of year in 3 letter character ( 001 – 365)  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%i');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%i'); |
| %m | It will return month in numeric format ( 1 – 12)  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%m');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%m'); |
| %p | It will return AM / PM  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%p');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%p'); |
| %s | It will return seconds in the double letter format ( 00 -59)  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%s');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%s'); |
| %W | It will return Week Name  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%W');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%W'); |
| %y | It will return year in the two digit format  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%y');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%y'); |
| %Y | It will return year in the 4 digit format  Ex: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%Y');  SELECT DATE\_FORMAT('2023-07-05 20:45:00', '%Y'); |

Ex of formats using : [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DATE\_FORMAT(NOW(), '%D %M, %Y ');

# DAYNAME

This method returns the name of the weekday.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DAYNAME('2023-07-05');

# DAYOFWEEK

This method returns week day in a number format.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DAYOFWEEK('2023-07-05');

# DAYOFTHEYEAR

This method returns the day of the year for a specific date.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DAYOFYEAR('2023-07-05');

# Hour

This method will return hours from a given time.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) hour('25:12:00');

# Minute

This method will return minutes from a given time.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) minute('25:12:00');

# Second

This method will return seconds from a given time.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) second('25:12:59');

# Month

This method will return month in numeric format for a specific date.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) month('2023-12-31');

# Monthname

This method will return month name for a specific date.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) monthname('2023-12-31');

# SYSDATE

This method will return current date and time ( same like now()).

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) SYSDATE();

# Timediff

This method will return time difference between two date&times.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) timediff(SYSDATE(), '2023-07-05 20:30:00');

# To\_days

This method will return number of days between **01-01-1582 to given date.**

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) to\_days('2023-06-07');

# Weekday

This method will return week day in a number format for specific day.

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) weekday('2023-06-07');

# WeekofYear

This method will returns the calendar week of the day as a number in a range of 1 to 53.

Ex:

[SELECT](https://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) WEEKOFYEAR('2023-07-06');

# Sub – Queries

Ex:

A query inside a query, means one outer query and one inner query.

select \* from products where category\_id = (select id from categories where category\_name = 'Cat4') ;

# Indexing:

Preparing logical groups of rows in a b-tree format.

Ex:

CREATE INDEX ix\_cate ON Products(category\_id);

CREATE INDEX < index\_name> ON <table\_name> ( < columnname>, <columnname);

# My SQL Storage Engines.

* MyISAM
* Memory
* Merge
* InnoDB.

Advantages of InnoDB.

* InnoDB supports Cache & Indexing in very efficient manner.
* InnoDB supports lock the tables in a row level.
* InnoDB accepts transactions based mysql table.
* InnoDB achieves concurrent access by multiple users with high performance.
* InnoDB engine supports the usage of foreign keys that helps consistent update, insert and delete operations.
* Implemented ONDELETE CASCADE and ONUPDATE CASCADE.

DELIMITER //

CREATE PROCEDURE GetSalaries()

BEGIN

SELECT \* FROM `salaries`;

UPDATE `salaries` SET emploeename = 'vasuedited' where id = 1;

END //

DELIMITER ;