How to Start MySql Server

Download latest version of XAMPP From https://www.apachefriends.org/download.html

Install XAMPP and open XAMPP Control Panel

From Control panel you need to start two services apache, mysql

Now open browser and enter URL http://localhost/phpmyadmin/ or press admin button from XAMPP control panel.

Now you are connected with MySql server home page

MySQL Tutorial

MySQL is a widely used relational database management system (RDBMS).

MySQL is free and open-source.

MySQL is ideal for both small and large applications.

Introduction to MySQL

MySQL is a very popular open-source relational database management system (RDBMS).

What is MySQL?

- MySQL is a relational database management system
- MySQL is open-source
- MySQL is free
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, scalable, and easy to use
- MySQL is cross-platform
- MySQL is compliant with the ANSI SQL standard
- MySQL was first released in 1995
- MySQL is developed, distributed, and supported by **Oracle Corporation**
- MySQL is named after co-founder Monty Widenius's daughter: My

---

Who Uses MySQL?

- Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
- Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
- A very large number of web developers around the world

Show Data On Your Web Site

To build a web site that shows data from a database, you will need:

- An RDBMS database program (like MySQL)
- A server-side scripting language, like PHP
- To use SQL to get the data you want
- To use HTML / CSS to style the page

MySQL RDBMS

What is RDBMS?

RDBMS stands for Relational Database Management System.

RDBMS is a program used to maintain a relational database.

RDBMS is the basis for all modern database systems such as MySQL, Microsoft SQL Server, Oracle, and Microsoft Access.

RDBMS uses SQL queries to access the data in the database.

What is a Database Table?

A table is a collection of related data entries, and it consists of columns and rows.

A column holds specific information about every record in the table.

A record (or row) is each individual entry that exists in a table.

Look at a selection from the Northwind "Customers" table:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

The columns in the "Customers" table above are: CustomerID, CustomerName, ContactName, Address, City, PostalCode and Country. The table has 5 records (rows).

What is a Relational Database?

A relational database defines database relationships in the form of tables. The tables are related to each other - based on data common to each.

Look at the following three tables "Customers", "Orders", and "Shippers" from the Northwind database:

Customers Table

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

The relationship between the "Customers" table and the "Orders" table is the CustomerID column:

Orders Table

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10278	5	8	1996-08-12	2
10280	5	2	1996-08-14	1
10308	2	7	1996-09-18	3
10355	4	6	1996-11-15	1
10365	3	3	1996-11-27	2
10383	4	8	1996-12-16	3
10384	5	3	1996-12-16	3

The relationship between the "Orders" table and the "Shippers" table is the ShipperID column:

Shippers Table

ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shipping	(503) 555-9931

MySQL SQL

What is SQL? (Structured Query Language)

SQL is the standard language for dealing with Relational Databases.

SQL is used to insert, search, update, and delete database records.

How to Use SQL

The following SQL statement selects all the records in the "Customers" table:

```
SELECT * FROM Customers;
```

Keep in Mind That...

- SQL keywords are NOT case sensitive: `select` is the same as `SELECT`

In this tutorial we will write all SQL keywords in upper-case.

Semicolon after SQL Statements?

Some database systems require a semicolon at the end of each SQL statement.

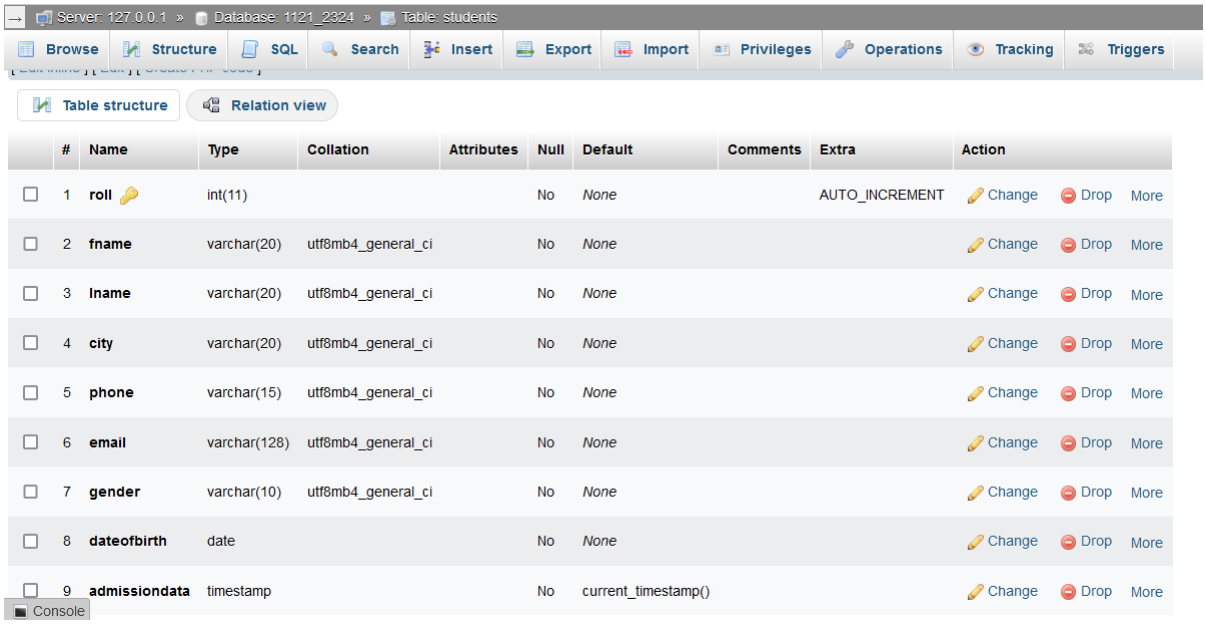
Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.

In this tutorial, we will use semicolon at the end of each SQL statement.

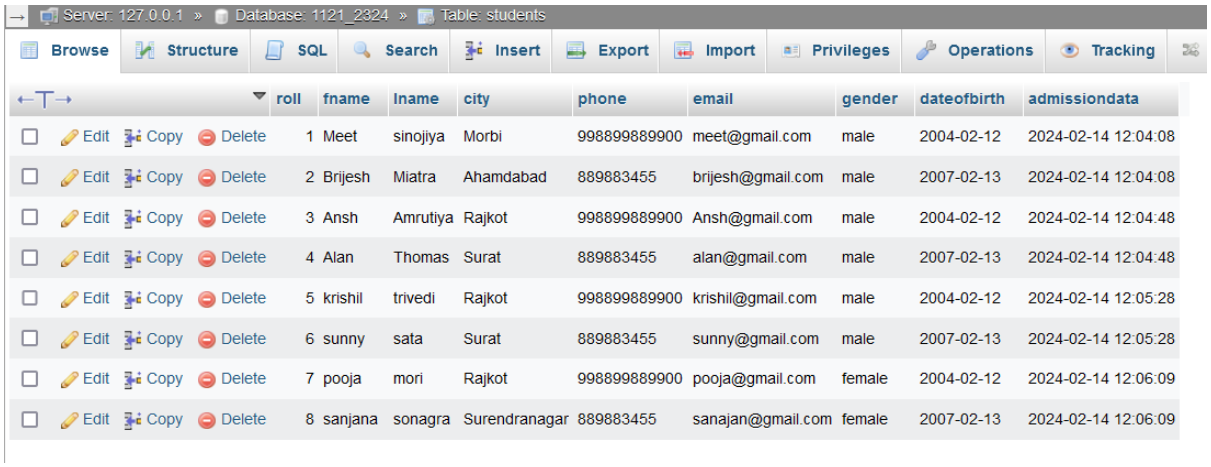
Some of The Most Important SQL Commands

- `SELECT` - extracts data from a database
- `UPDATE` - updates data in a database
- `DELETE` - deletes data from a database
- `INSERT INTO` - inserts new data into a database
- `CREATE DATABASE` - creates a new database
- `ALTER DATABASE` - modifies a database
- `CREATE TABLE` - creates a new table
- `ALTER TABLE` - modifies a table
- `DROP TABLE` - deletes a table
- `CREATE INDEX` - creates an index (search key)
- `DROP INDEX` - deletes an index

Before continue with MySql you need to create sample database and add some data in it.



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	roll	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	fname	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	lname	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
☐	4	city	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
☐	5	phone	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
☐	6	email	varchar(128)	utf8mb4_general_ci		No	None			Change Drop More
☐	7	gender	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
☐	8	dateofbirth	date			No	None			Change Drop More
☐	9	admissiondata	timestamp			No	current_timestamp()			Change Drop More



	roll	fname	lname	city	phone	email	gender	dateofbirth	admissiondata
☐ Edit Copy Delete	1	Meet	sinojiya	Morbi	998899889900	meet@gmail.com	male	2004-02-12	2024-02-14 12:04:08
☐ Edit Copy Delete	2	Brijesh	Miatra	Ahamdabad	889883455	brijesh@gmail.com	male	2007-02-13	2024-02-14 12:04:08
☐ Edit Copy Delete	3	Ansh	Amrutiya	Rajkot	998899889900	Ansh@gmail.com	male	2004-02-12	2024-02-14 12:04:48
☐ Edit Copy Delete	4	Alan	Thomas	Surat	889883455	alan@gmail.com	male	2007-02-13	2024-02-14 12:04:48
☐ Edit Copy Delete	5	krishil	trivedi	Rajkot	998899889900	krishil@gmail.com	male	2004-02-12	2024-02-14 12:05:28
☐ Edit Copy Delete	6	sunny	sata	Surat	889883455	sunny@gmail.com	male	2007-02-13	2024-02-14 12:05:28
☐ Edit Copy Delete	7	pooja	mori	Rajkot	998899889900	pooja@gmail.com	female	2004-02-12	2024-02-14 12:06:09
☐ Edit Copy Delete	8	sanjana	sonagra	Surendranagar	889883455	sanajan@gmail.com	female	2007-02-13	2024-02-14 12:06:09

MySQL SELECT Statement

The MySQL SELECT Statement

The `SELECT` statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

SELECT Syntax

```
SELECT column1, column2, ... FROM table_name;
```

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

```
SELECT * FROM table_name;
```

```
SELECT roll, fname, lname, city from students
```

```
SELECT *  from students;
```

The MySQL SELECT DISTINCT Statement

The `SELECT DISTINCT` statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

SELECT DISTINCT Syntax

```
SELECT DISTINCT column1, column2, ... FROM table_name;
```

```
SELECT city  from students;

SELECT DISTINCT city  from students;

SELECT count(DISTINCT city)  from students;

SELECT roll, fname, lname, dateofbirth from students
```

MySQL WHERE Clause

The MySQL WHERE Clause

The `WHERE` clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

WHERE Syntax

SELECT *column1*, *column2, ...* FROM *table_name*WHERE *condition*;

SELECT roll, fname, lname, dateofbirth from students WHERE roll = 1;

SELECT roll, fname, lname, dateofbirth from students WHERE roll > 5;

SELECT roll, fname, lname, dateofbirth from students WHERE not roll > 5;

Note: The `WHERE` clause is not only used in `SELECT` statements, it is also used in `UPDATE`, `DELETE`, etc.!

SELECT * from students WHERE city = 'rajkot'

SELECT * from students WHERE not city = 'rajkot';

SELECT * from students WHERE city <> "rajkot";

Text Fields vs. Numeric Fields

SQL requires single quotes around text values (most database systems will also allow double quotes).

However, numeric fields should not be enclosed in quotes:

SELECT * from students WHERE not city = rajkot;

SELECT * from students WHERE roll BETWEEN 1 and 5

MySQL AND, OR and NOT Operators

The MySQL AND, OR and NOT Operators

The `WHERE` clause can be combined with `AND`, `OR`, and `NOT` operators.

The `AND` and `OR` operators are used to filter records based on more than one condition:

- The `AND` operator displays a record if all the conditions separated by `AND` are TRUE.
- The `OR` operator displays a record if any of the conditions separated by `OR` is TRUE.
- The `NOT` operator displays a record if the condition(s) is NOT TRUE.

AND Syntax

```
SELECT column1, column2, ... FROM table_name WHERE condition1 AND condition2 AND condition3 ...;
```

```
SELECT * from students WHERE roll = 1
```

```
SELECT * from students WHERE roll = 1 and city = 'Rajkot';
```

OR Syntax

```
SELECT column1, column2, ... FROM table_name WHERE condition1 OR condition2 OR condition3 ...;
```

```
SELECT * from students WHERE city = 'surat' or city = 'Rajkot';
```

NOT Syntax

```
SELECT column1, column2, ... FROM table_name WHERE NOT condition;
```

```
SELECT * from students WHERE not (city = 'surat' or  city = 'Rajkot');
```

Combining AND, OR and NOT

You can also combine the `AND`, `OR` and `NOT` operators.

```
SELECT * from students WHERE roll = 1 and (city = 'surat' or  city = 'Rajkot' or city = 'morbi');
```

```
SELECT * from students WHERE roll = 1 or roll = 5 and city = 'Rajkot';s
```