GIFT School of Engineering and Applied Sciences

Fall 2022

CS-244: Database Systems-Lab

Lab-3 Manual

Introduction to MySQL and SQL

Introduction to Lab

This lab introduces students to the MySQL database management system. The lab discusses the basic operations of working with the MySQL using PhpMyAdmin. The fundamental concepts of using Structured Query Language (SQL) are also introduced.

The main topics of this lab include:

1. Introduction to MySQL
2. System Development Life Cycle
3. Data Storage on Different Media
4. Definition of a Relational Database
5. Relational Database Terminology
6. Starting a Database Session
7. Communicating with an RDBMS Using SQL
8. SQL Statements
9. Configuring and connecting to MySQL
10. Creating& Finding Tables in a Database
11. Displaying the Table Data
12. Displaying the Structure of Tables
13. Practice SQL Statements

This lab resembles a walkthrough and guides students by giving them hands-on practice of applying the various operations of working with the database. This lab also contains various exercises of using the SQLSELECT statement for fetching data from tables.

Objectives of this Lab

At the end of this lab, students should be able to:

a) Understand the usage MySQL and SQL
b) Understand how to import a simple database schema.
c) Understand the purpose and the various types of SQL languages
d) Understand how to use the SQLSELECT statement for listing tables and simple data fetching from tables
e) Understand how to list the structure of tables

1. Introduction to the MySQL

MySQL is a Relational Database Management System (RDBMS).

RDBMS means R--DB--MS.

- DB stands for Database, a repository for the information store.

i. The data in a database is organized into tables, and each table is organized into *rows* and *columns*.

ii. Each row in a table is called a record. A record may contain several pieces (called *fields*) of information, and each column in a table is known as a *field*.

-MS stands for Management System, the software that allows you to insert, retrieve, modify, or delete records.

-R stands for Relational, indicates a particular kind of DBMS that is good at relating information stored in one table to information stored in another table by looking for elements common to each of them. Relational DBMS has the advantage of efficient storage, and retrieval mechanisms for data, and uses normalization process during design of RDBMS. Database normalization process is beyond the scope of this article, and several references are available.

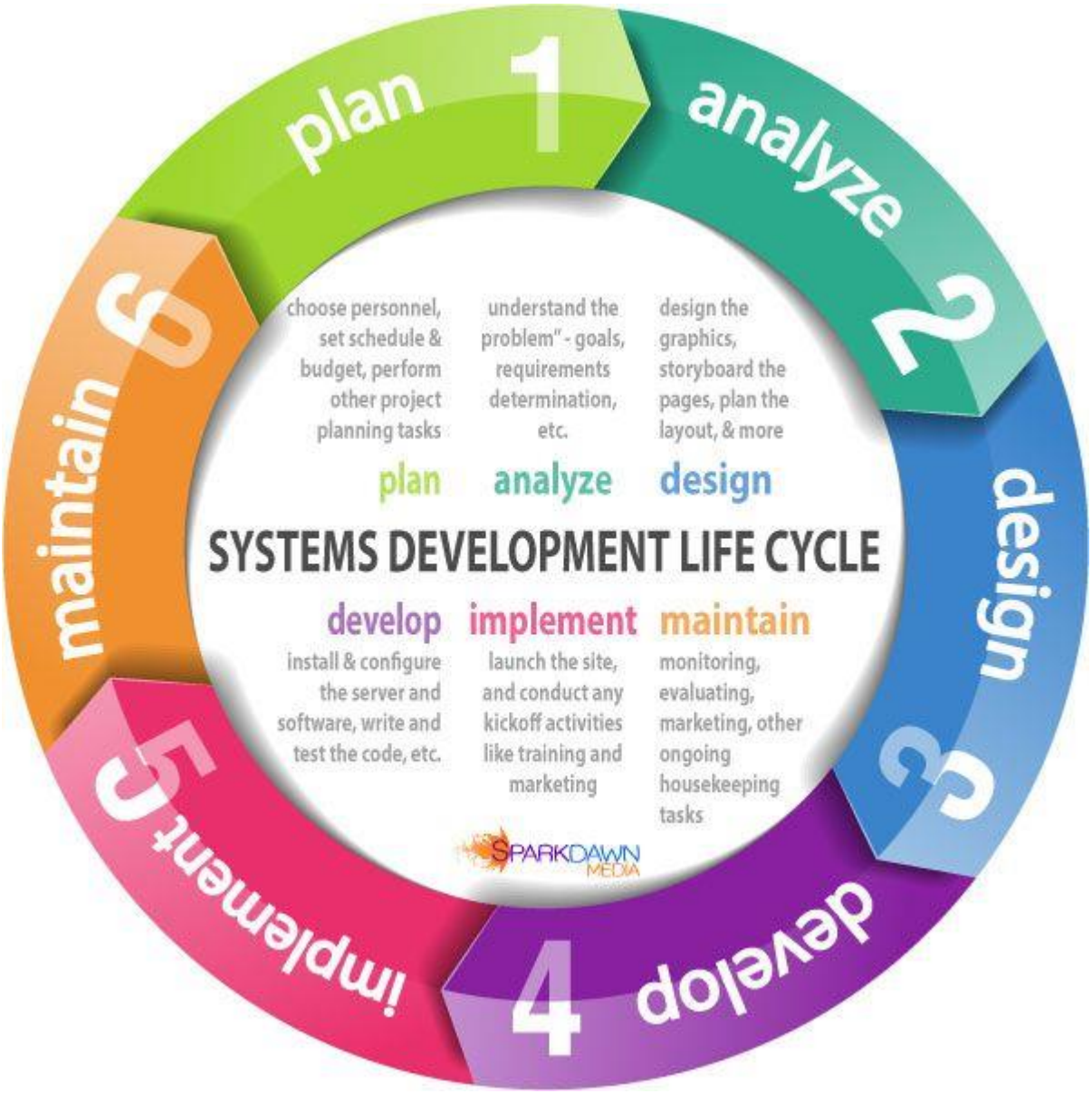
MySQL operates using client/server architecture in which the server runs on the machine containing the databases and clients connect to the server over a network. The server operating systems is usually a Linux or Windows 2000 operating system. Typically, MySQL is supported on Windows XP, Windows Server 2003, Red Hat Fedora Linux, and Debian Linux, and others. As with any other client/server application, MySQL is a multi-user database system, meaning several users can access the database simultaneously. Here:

-The server (MySQL server) listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients.

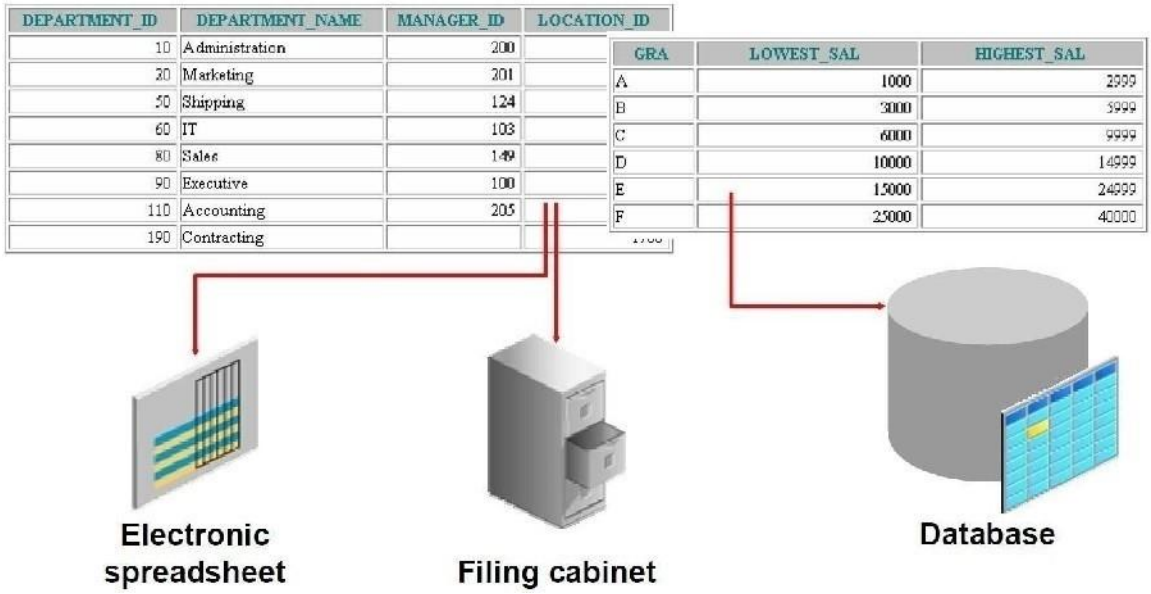
- Clients are programs that connect to the database server and issue queries in a pre-specified format. MySQL is compatible with the standards-based SQL (SQL stands for Structured Query Language) language. The client program may contact the server programmatically (meaning a program call the server during execution) or manually. For example, when you are issuing commands over a telnet session to a MySQL server, you are issuing the requests to the server by typing commands at your command prompt manually. On the other hand, if you have input some data (say your credit card information on the Internet towards purchase of some goods) in a form, and the form is processed by using a server-side program, then the MySQL server is contacted programmatically. This is often the case in credit card approvals, member subscriptions etc.

2. System Development Life Cycle

From concept to production, you can develop a database by using the system-development life cycle, which contains multiple stages of development. This top-down, systematic approach to database development transforms business information requirements into an operational database.


plan
1
analyze
2
design
3
develop
4
implement
5
maintain
6
choose personnel, set schedule & budget, perform other project planning tasks
understand the problem" - goals, requirements determination, etc.
design the graphics, storyboard the pages, plan the layout, & more
plan
analyze
design
SYSTEMS DEVELOPMENT LIFE CYCLE
develop
install & configure the server and software, write and test the code, etc.
implement
launch the site, and conduct any kickoff activities like training and marketing
maintain
monitoring, evaluating, marketing, other ongoing housekeeping tasks
SPARKDAWN MEDIA

3. Data Storage on Different Media



Every organization has some information needs.

For Example: A library keeps a list of members, books, due dates, and fines. A company needs to save information about employees, departments and salaries. These pieces of information are called *data*.

Organizations can store data on various media and in different formats, such as a hard-copy document in a filing cabinet or data stored in electronic spreadsheets or in databases.

Database: A *database* is an organized collection of information.

To manage databases, you need a database management system (DBMS).

DBMS: A DBMS is a program that stores, retrieves and modifies data in databases on request.

Types of databases: There are four main types of databases:

a. *Hierarchical*
b. *Network*
c. *Relational*
d. *Object relational*.

4. Definition of a Relational Database

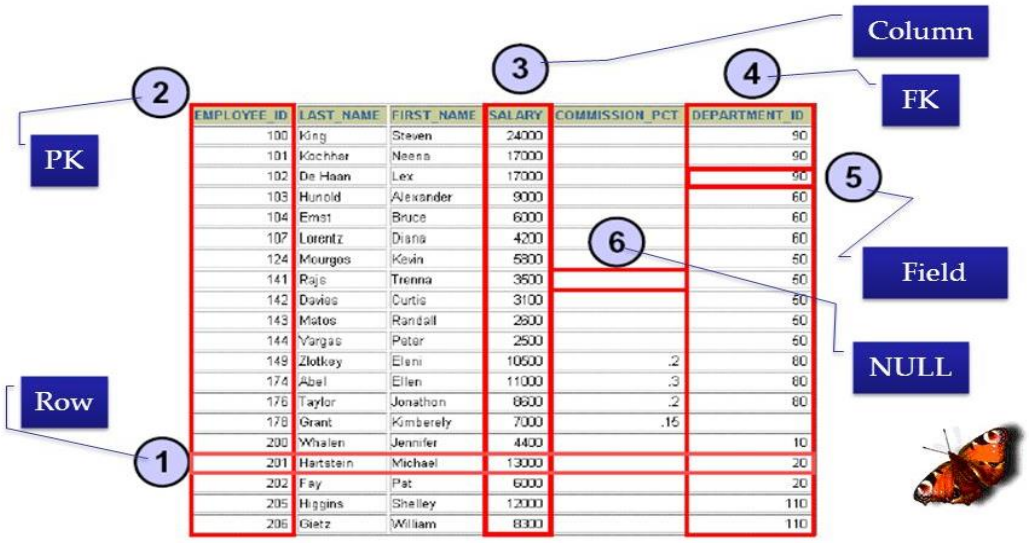
"A relational database uses relations or two-dimensional tables to store information."

For example, you might want to store information about all the employees in your company. In a relational database, you create several tables to store different pieces of information about your employees, such as an employee table, a department table, and a salary table.

5. Relational Database Terminology

A relational database can contain one or many tables. A table is the basic storage structure of an RDBMS.
A table holds all the data necessary about something in the real world, such as employees, invoices, or customers.
The diagram shows the contents of the EMPLOYEES *table* or *relation*.

The numbers indicate the following:



EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	King	Steven	24000		90
101	Kochhar	Neena	17000		90
102	De Haan	Lex	17000		90
103	Hunold	Alexander	9000		60
104	Ernst	Bruce	6000		60
107	Lorentz	Diana	4200		60
124	Mourgos	Kevin	5800		50
141	Rajs	Trenna	3500		50
142	Davies	Curtis	3100		50
143	Matos	Randall	2600		50
144	Vargas	Peter	2500		50
149	Zlotkey	Eleni	10500	.2	80
174	Abel	Ellen	11000	.3	80
176	Taylor	Jonathon	8600	.2	80
178	Grant	Kimberely	7000	.15	
200	Whalen	Jennifer	4400		10
201	Hartstein	Michael	13000		20
202	Fay	Pat	6000		20
205	Higgins	Shelley	12000		110
206	Gietz	William	8300		110

1. A single *row* (or *tuple*) representing all data required for a particular employee.
2. Each row in a table should be identified by a primary key, which permits no duplicate rows. The order of rows is insignificant; specify the row order when the data is retrieved.
3. A *column* or attribute containing the employee number. The employee number identifies a *unique* employee in the EMPLOYEES table. In this example, the employee number column is designated as the *primary key*. A primary key must contain a value, and the value must be unique. A column that is not a key value. A column represents one kind of data in a table; in

this example, the data is the salaries of all the employees. Column order is insignificant when storing data specify the column order when the data is retrieved.

4. A column containing the department number, which is also a *foreign key*. A foreign key is a column that defines how tables relate to each other. A foreign key refers to a primary key or a unique key in the same table or in another table. In the example, DEPARTMENT_ID *uniquely* identifies a department in the DEPARTMENTS table.
5. A *field* can be found at the intersection of a row and a column. There can be only one value in it.
6. A field may have no value in it. This is called a *null value*. In the EMPLOYEES table, only those employees who have the role of sales representative have a value in the COMMISSION_PCT (commission) field

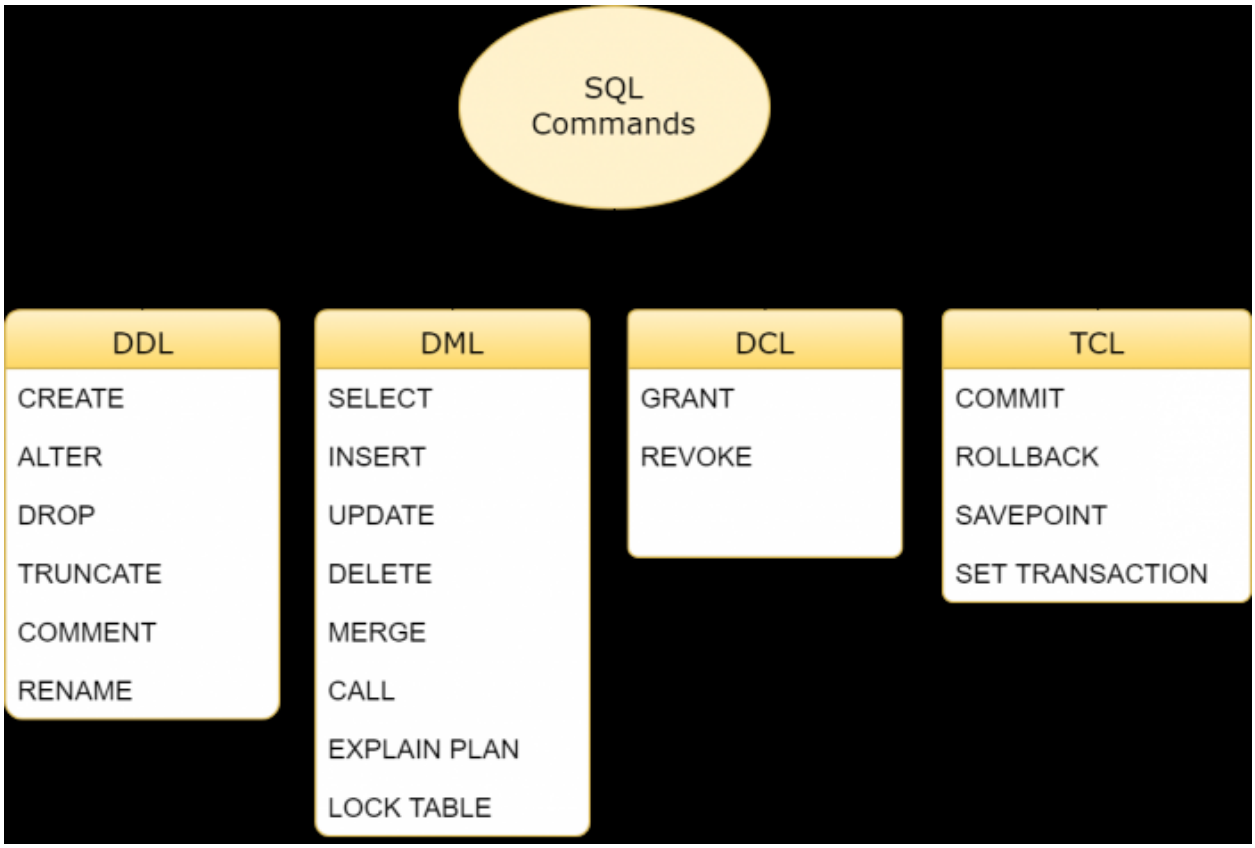
6. Communicating with an RDBMS Using SQL

Using SQL, you can communicate with the MySQL.

SQL has the following advantages:

A. Efficient
B. Easy to learn and use
C. Functionally complete (With SQL, you can define, retrieve, and manipulate data in the tables.)

7. SQL Statements



Command	Description
insert	to insert a new row
update	to update existing row
delete	to delete a row
merge	merging two rows or two tables

Command	Description
create	to create new table or database
alter	for alteration
truncate	delete data from table
drop	to drop a table
rename	to rename a table

Command	Description
select	retrieve records from one or more table

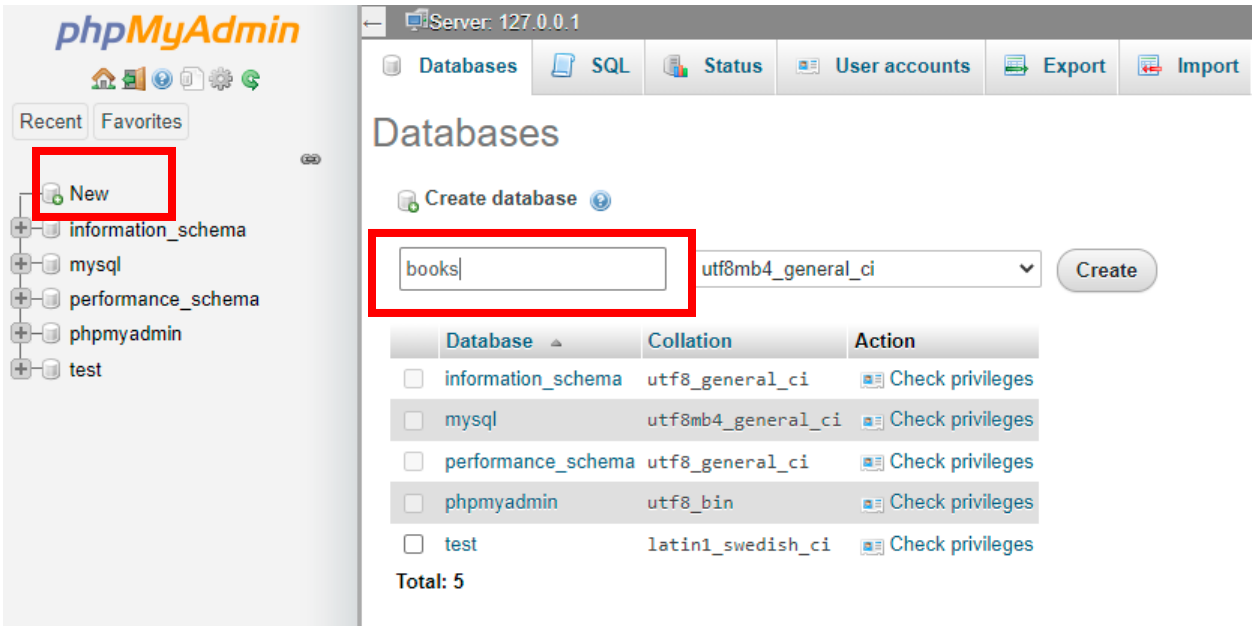
Command	Description
commit	to permanently save
rollback	to undo change
savepoint	to save temporarily

Command	Description
grant	grant permission of right
revoke	take back permission.

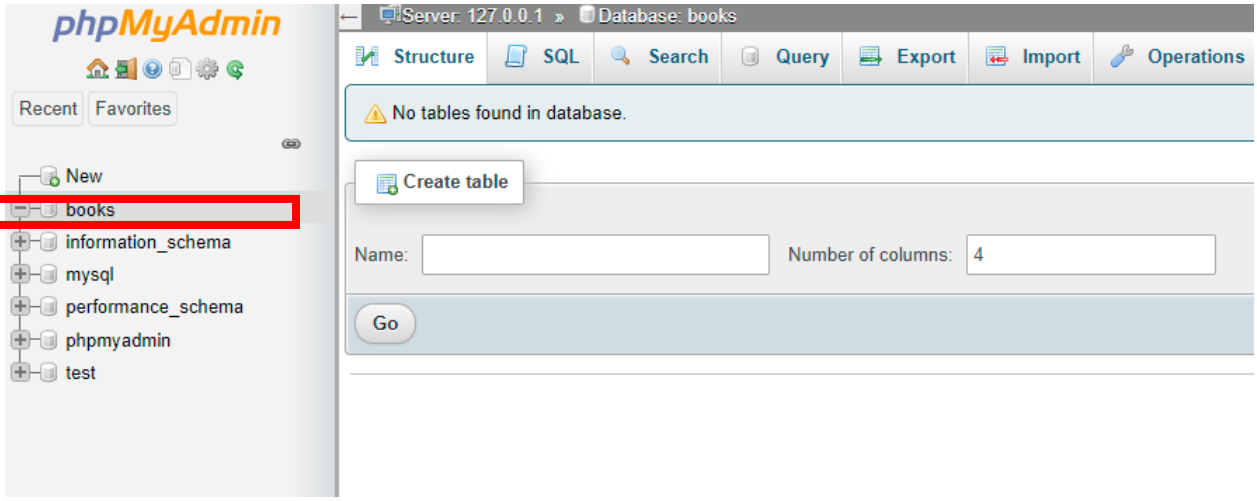
10. Creating database

First of all open phpMyAdmin perform the following tasks.

1. Create database with the name “books”.



No click on creates button and your “books” database will look like this.



1. PHP MySQL Connection

1.1. Ways of Connecting to MySQL through PHP

In order to store or access the data inside a MySQL database, you first need to connect to the MySQL database server. PHP offers two different ways to connect to MySQL server: **MySQLi** (Improved MySQL) and **PDO** (PHP Data Objects) extensions.

While the PDO extension is more portable and supports more than twelve different databases, MySQLi extension as the name suggests supports MySQL database only. MySQLi extension however provides an easier way to connect to, and execute queries on, a MySQL database server. Both PDO and MySQLi offer an object-oriented API, but MySQLi also offers a procedural API which is relatively easy for beginners to understand.

Tip: The PHP's MySQLi extension provides both speed and feature benefits over the PDO extension, so it could be a better choice for MySQL-specific projects.

1.2. Connecting to MySQL Database Server

In PHP you can easily do this using the mysqli_connect() function. All communication between PHP and the MySQL database server takes place through this connection. Here're the basic syntaxes for connecting to MySQL using MySQLi.

```
$link = mysqli_connect("hostname", "username", "password", "database");
```

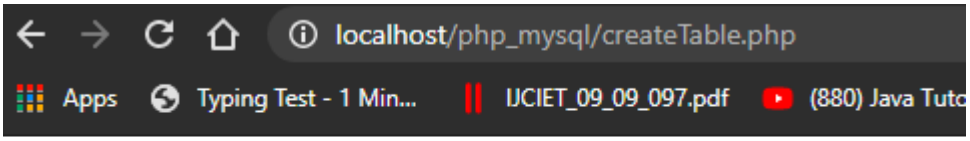
The *hostname* parameter in the above syntax specifies the host name (e.g. localhost), or IP address of the MySQL server, whereas the *username* and *password* parameters specifies the credentials to access MySQL server, and the *database* parameter, if provided will specify the default MySQL database to be used when performing queries.

The following example shows how to connect to MySQL database server using MySQLi

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "books";
//create connection to MySQL
$conn = mysqli_connect($servername , $username , $password ,
$database) or die("Cannot connect to
Database".mysqli_connect_error());
echo "Connected to database succesfully"."<br>";
 ?>
```

Note: The default username for MySQL database server is root and there is no password. However to prevent your databases from intrusion and unauthorized access you should set password for MySQL accounts.

Output of above Example in Web Browser:



Connected to database succesfully

The End