

# Database Connectivity Using PHP & MySQL

Prepared By: Bhavana Hotchandani, DCS, INDUS University

# Introduction



- **Why Database?**
  - Databases have become an essential component of every web-based project. From a simple blog to a corporate portal, database engines drive the application and are the central location where all processes and logics come together.
- **What is a Database?**
  - A database is a collection of data and information that is stored in an organized manner that can be retrieved, updated and deleted in the future. Database can be stored in files or in different tables in a system.
  - A table consists of rows and columns. Columns are different entities or characteristics of that table, for example, if we create a table of teachers, its entities would be ID, Name, Age, and Qualification.
- **What is Database Management System?**
  - Database management helps you handle your databases and assists you to create functions to insert, update, delete, and view your databases.

# List of Software for Database Software

- There are many software which can help you create and manage your databases. Some are paid, while some are free. Have a look at some of them below:
  - SQL Server (paid)
  - Oracle (paid+free)
  - MySQL (free)
  - MariaDB (free)
  - FireBase (paid+free)
  - Percona (free)
- We will be using MySQL with PHP as it is free and open source, and is used by a major portion of database users.

# PHP- Supported Databases

- Popular Databases for PHP Web Application Development
  - *MySQL*
  - *PostgreSQL*
  - *SYBASE*
  - *IBM-DB2*
  - *Oracle Database*

First, MySQL is free (as in price), although the licensing has changed.

Second, MySQL is also stable, meaning that it's not prone to crashing even under load.

Third, MySQL is lightweight, meaning that it doesn't require many resources to install or run.

Fourth, MySQL is fast and easy to use.

Finally, MySQL is powerful, with all of the features required for web applications.

# Why MySQL is so popular?

# What is MySQL?

- MySQL is a database management system developed by Oracle Corporation used mostly for web servers. MySQL can be deployed and run on any operating system. MySQL is easy, fast and can be used for any type of database whether it is relational or simple, large or small. Since it uses standard SQL for queries, what this means is that if you learn MySQL you will be able to make queries on different Database management software also.
- When PHP developers usually say database, they are referring to MySQL, a highly popular DBMS that can power projects of all sizes. MySQL's ability to handle huge volumes of data without breaking a sweat is one of its biggest selling points.

# PHP methods to connect MySQL

- There are three types of methods in PHP to connect MySQL database through backend:
  - MySQL
  - MySQLi
  - PDO
- My SQL: mysql() is now obsolete because of security issues like SQL injection etc, but the other two are being actively used.
- MySQLi: It is an API that uses a connector function to link the backend of the PHP app to the MySQL database. It provides a better set of functions and extensions. It works just like the previous version, but it is more safe and faster. **We will use this.**
- PDO: PHP Data Objects (PDO) extension is a Database Abstraction Layer. It is like an interface for the backend to interact with the MySQL database and make changes without making any change in the PHP code. It also gives you the freedom to work with multiple databases. The major advantage of using PDO is that your code stays simple and portable.

# Connect to MySQL with MySQL Improved extension

- **MySQLi** is an extension that only supports MySQL databases. It allows access to new functionalities found in MySQL systems (version 4.1. and above), providing both an object-oriented and procedural interface.
- The MySQLi extension is included PHP version 5 and newer.



# Integrating PHP and MySQLi

- 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.
  - `$link = mysqli_connect("hostname", "username", "password", "database");`
  - `$mysqli = new mysqli("hostname", "username", "password", "database");`
- The connection to the MySQL database server will be closed automatically as soon as the execution of the script ends. However, if you want to close it earlier you can do this by simply calling the PHP `mysqli_close()` function.
- `mysqli_close($link);`

# Making MySQL Queries



A database query from PHP is basically a MySQL command wrapped up in a tiny PHP function called `mysqli_query()`.



This is where you use the basic SQL workhorses of `SELECT`, `INSERT`, `UPDATE`, and `DELETE`.



The function `mysqli_query` takes as arguments a link identifier and the query string.



It returns a `TRUE` (nonzero) integer value if the query was executed successfully *even if no rows were affected*.



It returns a `FALSE` integer if the query was illegal or not properly executed for some other reason.

# Handling Results

- If your query is an INSERT, UPDATE, DELETE, CREATE TABLE, or DROP TABLE and returned TRUE, you can now use `mysql_affected_rows` to see how many rows were changed by the query.
  - This function optionally takes a link identifier, which is only necessary if you are using multiple connections.
  - It *does not* take the result handle as an argument! You call the function like this, without a result handle:
  - `$affected_rows = mysqli_affected_rows();`
- If your query was a SELECT statement, you can use `mysqli_num_rows($result)` to find out how many rows were returned by a successful SELECT.

# Examples

Create DB & Table

Insert Data

Update & Delete