



Waterford Institute of Technology

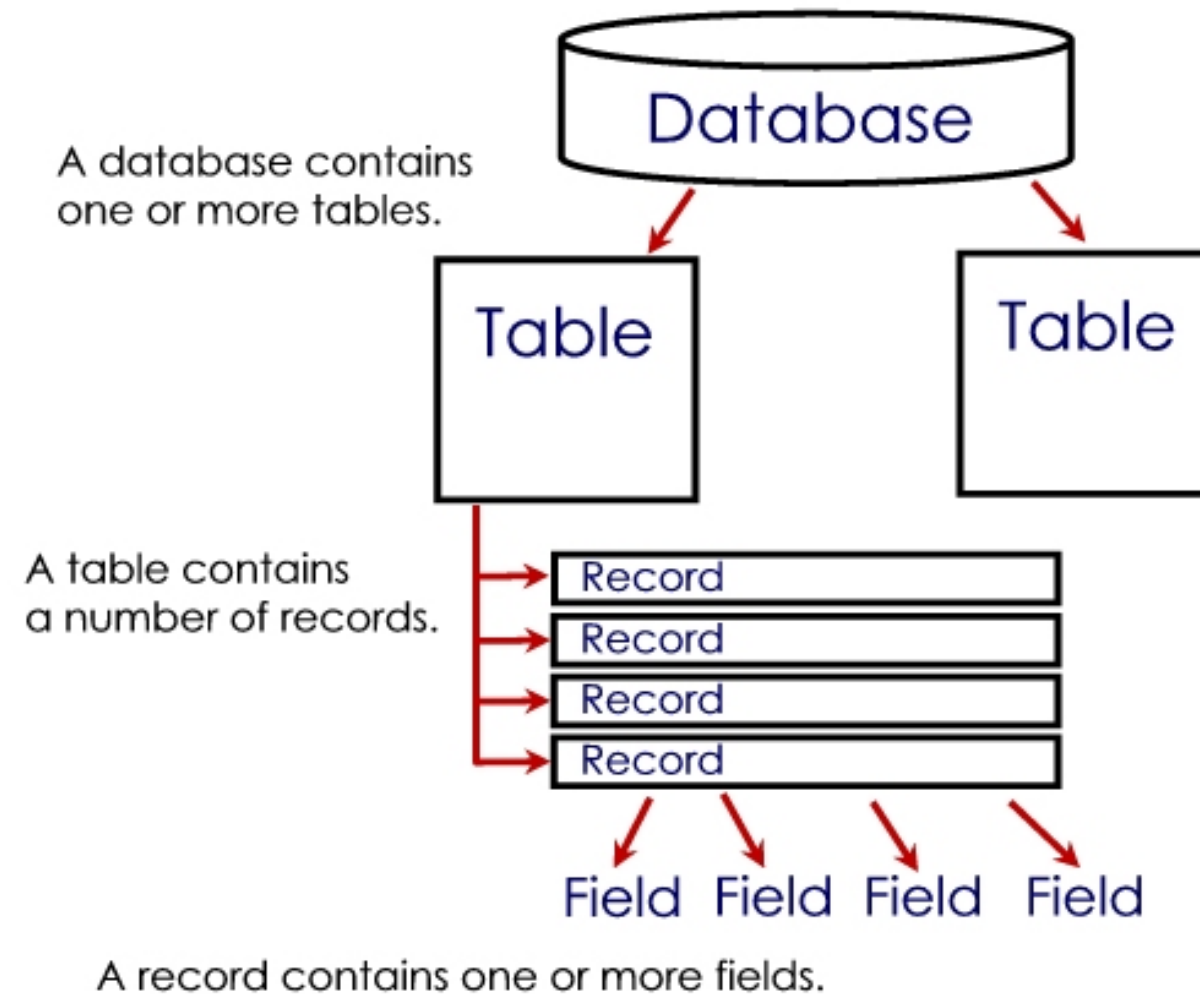


PHP+MYSQL

SAMITHA SOMATHILAKA

Department of Computing & Mathematics, WIT

DATABASE



WHY DATABASES

- Databases can store very large numbers of records efficiently (they take up little space).
- It is very quick and easy to find information.
- It is easy to add new data and to edit or delete old data.
- Data can be searched easily.
- Data can be sorted easily, for example into 'date first registered' order.
- Data can be imported into other applications.
- More than one person can access the same database at the same time - multi-access.
- Security may be better than in flat files.

MYSQL

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

MYSQL

- The data in a MySQL database are stored in tables.
- A table is a collection of related data, and it consists of columns and rows.
- PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)

PHPMYADMIN BRINGING MYSQL TO THE WEB

- phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web.
- phpMyAdmin supports a wide range of operations on MySQL and MariaDB.
- Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.



← → ↻ ⓘ localhost/dashboard/

Apache Friends

Applications

FAQs

HOW-TO Guides

PHPInfo

phpMyAdmin



XAMPP Apache + MariaDB + PHP + Perl







Welcome to XAMPP for Windows 7.1.33

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the [FAQs](#) to learn how to protect your site. Alternatively you can use [WAMP](#), [MAMP](#) or [LAMP](#) which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

phpMyAdmin



Recent Favorites

New

+

information_schema

+

mysql

+

performance_schema

+

phpmyadmin

+

piht

+

test

Server: 127.0.0.1

Databases

SQL

Status

User accounts

Export

Import

Settings

Replication

Variables

Charsets

More







Databases

Create database

Database name

utf8mb4_general_ci

Create

Database	Collation	Action
<input type="checkbox"/> information_schema	utf8_general_ci	 Check privileges
<input type="checkbox"/> mysql	utf8mb4_general_ci	 Check privileges
<input type="checkbox"/> performance_schema	utf8_general_ci	 Check privileges
<input type="checkbox"/> phpmyadmin	utf8_bin	 Check privileges
<input type="checkbox"/> piht	utf8mb4_general_ci	 Check privileges
<input type="checkbox"/> test	latin1_swedish_ci	 Check privileges
Total: 6		utf8mb4_general_ci

↑

☐ Check all


With selected:  Drop

Table name: Add column(s)

Name	Type ?	Length/Values ?	Default ?	Collation	Attributes	Null	Index
<input type="text" value="user_id"/> <small>Pick from Central Columns</small>	INT ▼	<input type="text"/>	None ▼	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--- ▼
<input type="text" value="user_name"/> <small>Pick from Central Columns</small>	VARCHAR ▼	<input type="text"/>	None ▼	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--- ▼
<input type="text" value="password"/> <small>Pick from Central Columns</small>	VARCHAR ▼	<input type="text"/>	None ▼	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--- ▼
<input type="text"/> <small>Pick from Central Columns</small>	INT ▼	<input type="text"/>	None ▼	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--- ▼

Column	Type	Function	Null	Value
user_id	int(11)	<input type="text"/>		<input type="text"/>
user_name	varchar(50)	<input type="text"/>		<input type="text"/>
password	varchar(20)	<input type="text"/>		<input type="text"/>

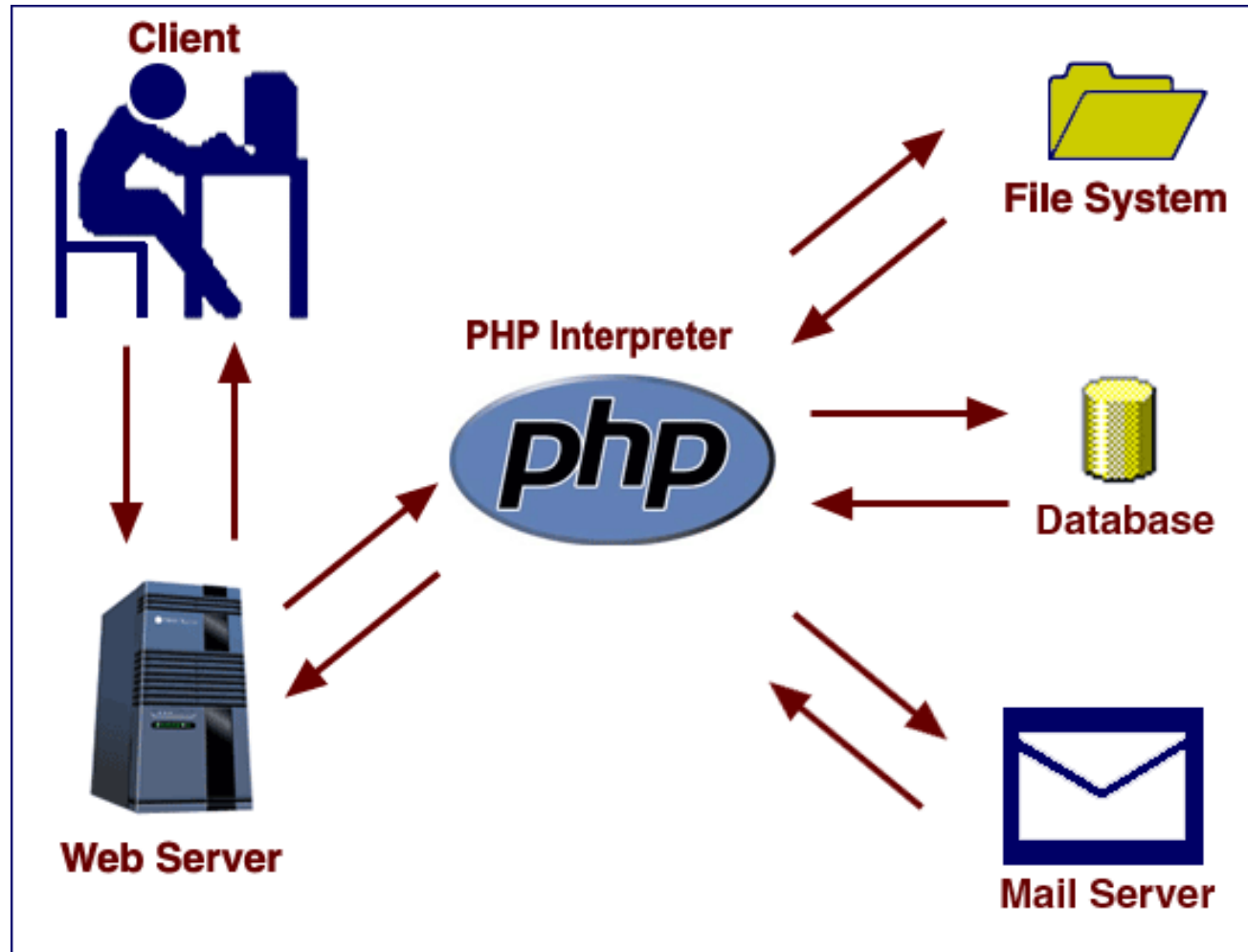
Go

☒ Ignore

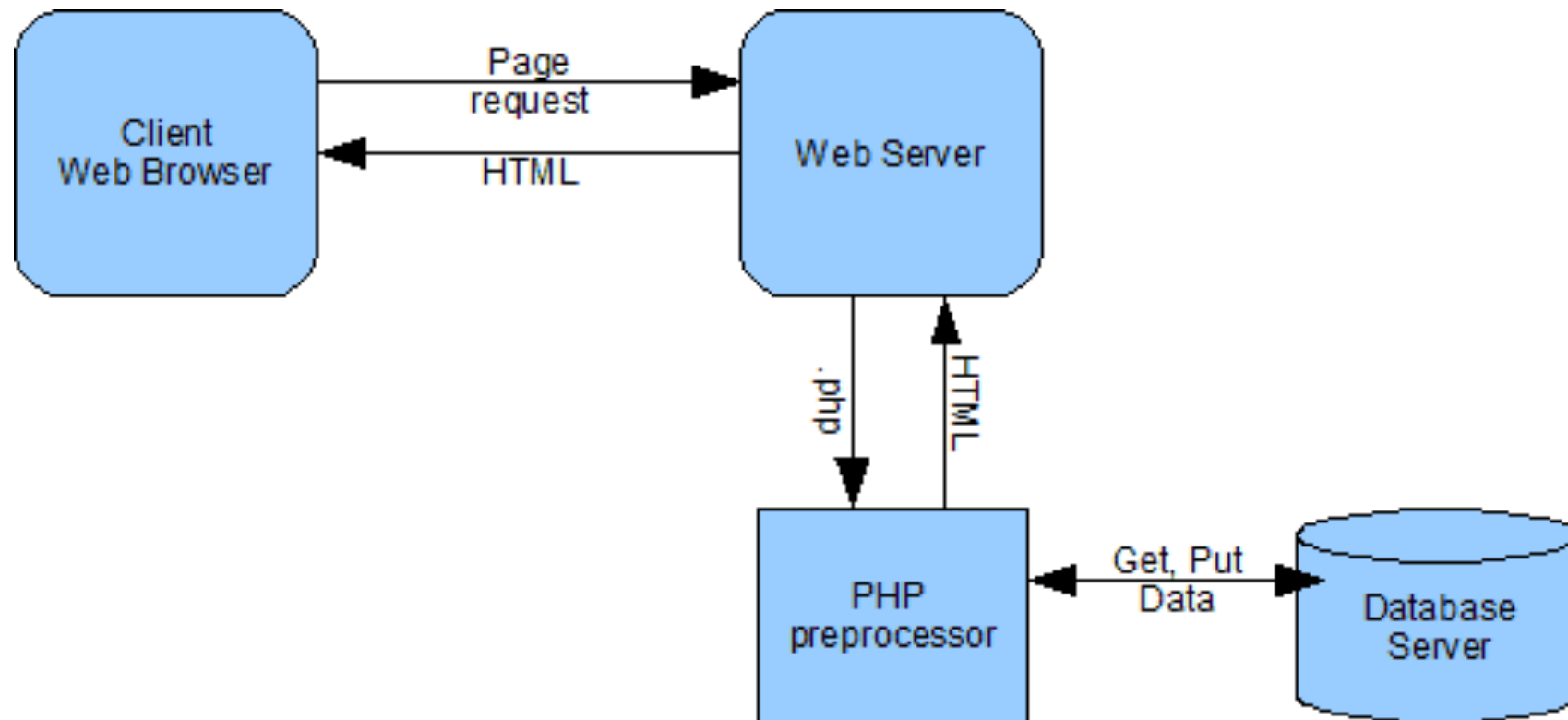
Column	Type	Function	Null	Value
user_id	int(11)	<input type="text"/>		<input type="text"/>
user_name	varchar(50)	<input type="text"/>		<input type="text"/>
password	varchar(20)	<input type="text"/>		<input type="text"/>

Go

PHP WITH OTHER SERVICES



PHP VS MYSQL



PHP FOR DATABASE ACCESS

- Connect to the MySQL server
 - `$connection = mysqli_connect("host","username","password","DB");`
- Perform SQL operations
- Disconnect from the server
 - `mysqli_close($connection);`

ERROR HANDLING

- All `mysql_` functions return NULL (or false) if they fail.
- Several functions are helpful in graceful failure
 - `die(string)` - halts and displays the string
 - `mysqli_connect_error()` - returns number of error

ERROR HANDLING EXAMPLES

```
$conn=mysqli_connect("localhost","root","","db");  
if(!$conn){  
    die(mysqli_connect_error());  
}else {  
    echo "successfull";  
}
```

BUILDING A QUERY

- Directly
 - `$query = 'select * from tablename';`
- Using input information
 - `$input = $_POST['var'];`
 - `$query = "select * from tablename where column=$input";`

RUNNING A QUERY

- `mysql_query` returns a result handle

```
$result = mysqli_query($connection, $query)
```

- `mysql_num_rows` indicates the number of rows returned

```
$num_rows = mysqli_num_rows($result)
```

- `mysql_fetch_array` creates array/hash of result

```
For ($n=0; $n<$num_rows;$n++)
```

```
    $row = mysqli_fetch_array($result)
```

RESULT OF FETCH_ARRAY

- Contains both numeric and index tags
- Values are duplicated
- Example:
 - Query: select surname, city from customers;
 - Row: (0=>'Walker', 'surname'=>'Walker', 1=>'Kent', 'city'=>'Kent');

PRINTING THE COMPLETE ROW

- By number

```
for ($i=0; $i<mysqli_num_fields($result); $i++)  
    echo $row[$i] . " ";
```

- By field

```
echo $row['surname'] . ' ' . $row['city'];
```

INSERTING INTO A DATABASE

- Collect data from a form
- Validate data (JavaScript, PHP or both)
- Create a query

```
$query = "insert into customer set cust_id = NULL, " . "surname =\" .  
    $surname . "\" ...
```

- Run the query

```
mysql_query($query, $db);
```

connection.php X

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $db = "pibt";
6  // Create connection
7  $conn = mysqli_connect($servername, $username, $password,$db);
8  // Check connection
9  if (!$conn) {
10     die("Connection failed: " . mysqli_connect_error());
11 }
12 echo "Connected successfully";
13 ?>
```

login_form.php

```
1 <html>
2 <body>
3 <form action="login.php" method="post">
4   User Name: <input type="text" name="username" id="username" /><br><br>
5   Password: <input type="password" name="password" id="password"/><br><br>
6
7   <input type="submit" name="add" value="LOGIN"/>
8
9 </form>
10 </body>
11 </html>
12
```

login.php

```
1 <?php
2 include "connection.php";
3
4 $user_name = $_POST['username'];
5 $password = $_POST['password'];
6
7 echo "User Name : " . $user_name . "<br>";
8 echo "Password : " . $password;
9 ?>
```