Using PHP with MYSQL

PHP & MYSQL

So far you've learned the theory behind relational databases and worked directly with MySQL through the mysql command-line tool.

Now it's time to get your PHP scripts talking to MySQL.

One of the ways PHP can connect to MYSQL is through PHP Data Objects(PDO)

PDO sits between the MySQL server and the PHP engine.

It gives you a nice, simple, clean set of classes and methods that you can use to work with MySQL databases.

Making a Connection

To make a connection to a MySQL database in your PHP script, all you need to do is create a new PDO object.

When you create the object, you pass in three arguments:

- The DSN, which describes the database to connect to;
- The username of the user you want to connect as;
- The user's password.

The returned PDO object serves as your script's connection to the database

Making a Connection

So, putting it all together, if you want to connect to a database called *mydatabase* on a server named *localhost* with your username being *myuser* and password being *mypass* :

```
$dsn = “mysql:host=localhost;dbname=mydatabase”;
$username = “myusername”;
$password = “mypass”;
$conn = new PDO( $dsn, $username, $password );
```

The returned PDO object serves as your script's connection to the database ($conn)

Reading and Manipulating Data

Now that you've connected to your database in your PHP script, you can run queries on the given database.

To send SQL statements to the MySQL server, you use the query method of the PDO object:

```
$sql = "SELECT * FROM fruit";
$conn -> query ( $sql );
```

This will run the specified SELECT statement against the database that you are connected to.

Reading and Manipulating Data

If your SQL statement returns rows of data as a result set, you can capture the data by assigning the result of `$conn->query()` to a variable:

```
$sql = "SELECT * FROM fruit";
$rows = $conn- > query ( $sql );
```

The result returned by `$conn->query()` is a set of records (rows).

Each row is an associative array containing all the field names and values for that row in the table.

You can use foreach loop to move through all the rows in the result set.

Reading Data

```
$sql = “SELECT * FROM fruit”;
$rows = $conn -> query( $sql );
foreach ( $rows as $row ) {
 echo “name = “ . $row[“name”] . “ < br / > ”;
 echo “color = “ . $row[“color”] . “ < br / > ”;
}
```

id	name	color
1	banana	yellow
2	tangerine	orange
3	plum	purple

output

banana
yellow
tangerine
orange
plum
purple

Let's see this in action

First, we will see how to perform different sql operations by using php.

Then we will see an example where we combine html, css, php and mysql to read field values from a form and insert them into a mysql database.

You can download all files for this example from here.

Make sure to update the mysql_connect.php file inside the folder, with your database, user and password information.

Another Example

This time we want to build a website that tracks our spending.

Basically, the website needs to provide 2 functions:

- a page which enables me to enter my expenses
- a page to show me a report of my expenses

expense table

First, we need to create a table to store the expenses.

The table needs to have the following fields:

- id : unique identifier for each expense
- description : a short description of the money we have spent
- date : When did we spend this money
- amount : how much did we spend

expense table

based on the mentioned criteria, we can create the table using the following SQL statement.

Notice the data types used for each field.

```
CREATE TABLE expense (
id INT UNSIGNED AUTO_INCREMENT,
description TEXT
edate DATE,
amount INT UNSIGNED,
PRIMARY KEY(id));
```

Inserting records into our expense table

Now that we have created the expense table, let's populate a couple of records for it.

Note how we specify the date.

```
INSERT INTO expense(description,edate,amount) VALUES('dinner at
subway','2016-12-01',11);

INSERT INTO expense(description,edate,amount) VALUES('coffee at
starbucks','2017-01-01',3);

INSERT INTO expense(description,edate,amount) VALUES('1 night hotel
in new york','2017-05-14',97);
```

Input form

So far we have created the table to store our information, and we have also inserted some records into our table.

Now it's time to create a simple page to let the user enter their expenses to be stored in this table.

This will be a form with three fields:

- description
- Transaction date
- Amount paid

Input form

We implement the form in a php page, that is also responsible for processing the user input.

The page should do the following:

- show the form to the user
- read input data when the form is submitted
- check if the user has provided values for all fields
- insert the input data into the database

Let's take a look at the code for this page (addexpense.php)

Expense Report

Now that we are done with adding expenses to our table, we should also create a page to show previous expenses to the user.

But instead of simply showing all records in the expense table, we want to let the user filter the specific records he wants.

For this purpose we will let the user filter expense report by date and amount

This means that, the user should be able to select a start and end date for the report as well as a minimum and maximum value for the amount field

Expense Report

Once the user gives us this information, we will only select and show the records that match the specified criteria

So we have 4 fields to filter the results:

- start date
- end date
- minimum amount
- maximum amount

Our query should result in selecting transactions that were made on a date between the start and end dates, and also the transaction amount needs to be between the minimum and maximum amount specified by the user.

Expense Report

But what if the user does not specify some of those fields?

We can assign default values to each field based on the data types we used for each of them:

Field	Default Value
Start Date	1000-1-1
End Date	9999-12-31
Minimum Amount	0
Maximum Amount	4,000,000,000

Expense Report

So far we know that our report page should:

- show a couple of fields to the user to let them filter the results
- show the records that match the user filters

For this, we would need a form to get the user input

when the user submits the form, we will read the input and customize our SELECT statement to fetch records that match the user requirements from the table.

we implement all of this in a php page.

Let's take a look at the code for this page (report.php)

Done!

We saw how to create a website that lets the user to keep track of their expenses.

You can download the code for this example from here.

Don't forget to change database connection properties in mysql_connect.php and also to create the table in MYSQL.