Database management

Contents

- MySQL
- Open and close a connection to a MySQL server
- Update data in a table
- Delete a record

MySQL

- A database is a structured collection of records or data stored in a computer system and organized in such a way that it can be quickly searched, and information can be rapidly retrieved.

- A MySQL database contains one or more tables, each of which contains records or rows. Within these rows are various columns or fields that contain the data itself.

MySQL

- This table shows the contents of an example database of five publications detailing the author, title, type, and year of publication.

Author	Title	Type	Year
Mark Twain	The Adventures of Tom Sawyer	Fiction	1876
Jane Austen	Pride and Prejudice	Fiction	1811
Charles Darwin	The Origin of Species	Non-Fiction	1856
Charles Dickens	The Old Curiosity Shop	Fiction	1841
William Shakespeare	Romeo and Juliet	Play	1594

- Each row in the table is the same as a row in a MySQL table, and each element within a row is the same as a MySQL field.

Open and close a connection to a MySQL server

In this chapter, we are working with the database and there are also <u>3 main steps</u>…

1. Open a connection to a MySQL server and select the database

2. Work with data in the database

3. Close the connection

```php
<?php
$server = "127.0.0.1";
$user = "root";
$password = "1234";
$dbname = "camt_db";

$connect = mysqli_connect($server, $user, $password, $dbname);

if (!$connect)
{
    die ("ERROR\: Cannot connect to the database $dbname on server $server using username $User ("
        .mysqli_connect_errno(). ", ".mysqli_connect_error(). ")");
}

mysqli_query($connect, "SET NAMES utf8");
?>
```
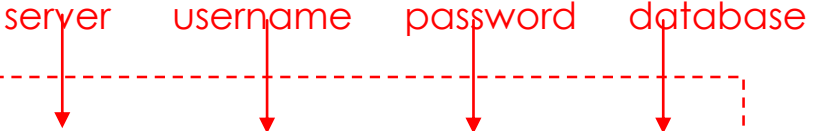
Open and close a connection to a MySQL server

Open a connection to a MySQL server and select the database

- We can use function **mysqli_connect()**

- This function returns a reference to the   connection   which is assigned to $connect

server   username   password   database

$connect = mysqli_connect('localhost', 'whip', 'whip123', 'test');

Open and close a connection to a MySQL server

- Besides, we can use **die()** to <u>terminate the PHP script</u> if the connection fails.

- We can also use **mysqli_connect_errno()** to <u>display the error number</u> and **mysqli_connect_error()** to <u>display the description of that error</u> (for ex. if user ID or password is incorrect, it will display "Access denied")

Open and close a connection to a MySQL server

Close the connection from MySQL server

- We can use function **mysqli_close()**

- This function is used with the $connect variable that we used in mysqli_connect()

```
mysqli_close($connect);
```

```php
<?php
require_once "config.php";

$userQuery = "SELECT * FROM `product`";
$result = mysqli_query($connect, $userQuery);

if (!$result) {
    die ("Could not successfully run the query $userQuery ".mysqli_error($connect));
}
if (mysqli_num_rows($result) == 0) {
    echo "No records were found with query $userQuery";
}
else {
    echo "<a href='add_product.html'>Add more product</a><br><br>";
    echo "<table border = \"1\">";
    echo "<tr><th>Product Name</th><th>Price</th><th>Quality</th></tr>";
    while ($row = mysqli_fetch_assoc($result)) {
        echo "<tr><td>".$row['productName']."</td><td>".$row['price']."</td><td>".$row['qty']."</td></tr>";
    }
    echo "</table>";
}
mysqli_close($connect);
?>
```

Using the MySQL SELECT query

Recall how to use 'select'

- Select * from stu_test

- Select firstName, lastName from stu_test

- Select score from stu_test where firstname ='John'

- Select score from stu_test where firstname = **'$name'**

$name = $_POST['name'];

Table stu_test

ID	firstName	lastName	score
0450	John	Willey	70
0451	Rose	Malone	82

Viewing the query results

- MySQL queries are submitted using the **mysqli_query()** function

- You have to open a connection with mysqli_connect() **before** you can submit a query

- mysqli_query() returns the **result** of the query, so you have to create a variable to receive it

```
$result = mysqli_query($connect, "SELECT * from stu_test");
```

Viewing the query results

- Besides, we can use **mysqli_error()** to <u>display the error message</u> from processing that query.

- **_Note_** _that the $connect variable is required to use with this function._

Viewing the query results

- If the query was successfully processed but no records met the requirements.

  We need to notify the users by using the **mysqli_num_rows()** function to

  count the number of records that were returned from the query

  ```
  mysqli_num_rows($result);
  ```

Viewing the query results

- If the query returned a result set (1 or more records), we can process $result

  with a **loop** and **mysqli_fetch_assoc()** function to <u>extract the next record</u> from

  the result set, then we can put them in an **associative array.**

```
$row = mysqli_fetch_assoc($result);
```

```html
<!DOCTYPE html>
<html>
<head>
    <title>Add product</title>
</head>
<body>
    <form method="POST" action="add_product.php">
        <label>Product name:</label>
        <input type="text" name="productName"><br>
        <label>Price:</label>
        <input type="text" name="price"><br>
        <label>Qulity:</label>
        <input type="text" name="qty"><br>
        <input type="submit" name="" value="submit">
    </form>
</body>
</html>
```

add_product.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>Add product</title>
</head>
<body>
    <form method="POST" action="add_product.php">
        <label>Product name:</label>
        <input type="text" name="productName"><br>
        <label>Price:</label>
        <input type="text" name="price"><br>
        <label>Qulity:</label>
        <input type="text" name="qty"><br>
        <input type="submit" name="" value="submit">
    </form>
</body>
</html>
```

add_product.php

```php
<?php

require_once "config.php";

$productName = $_POST["productName"];
$price = $_POST["price"];
$qty = $_POST["qty"];

...

?>
```

```php
<?php

require_once "config.php";

$productName = $_POST["productName"];
$price = $_POST["price"];
$qty = $_POST["qty"];

$userQuery = "INSERT INTO `product` (`productID`, `productName`, `price`, `qty`) VALUES (NULL, '$productName', $price, $qty)";

$result = mysqli_query($connect, $userQuery);

if (!$result) {
    die ("Could not successfully run the query $userQuery ".mysqli_error($connect));
}
else {
    echo "Insert successfully.<br><b>";
    echo "<a href='display_product-2.php'>Display product</a><br><br>";
}
mysqli_close($connect);
?>
```

display_product.php

```php
<?php
...

    echo "<table border = \"1\">";
    echo "<tr><th>Product Name</th><th>Price</th><th>Quality</th></tr>";
    while ($row = mysqli_fetch_assoc($result)) {
        echo "<tr><td>".$row['productName']."</td><td>".$row['price']."</td><td>".$row['qty']."</td></tr>";
    }
    echo "</table>";

...

?>
```

```php
<table border="1">
    <tr>
        <th>Product Name</th><th>Price</th><th>Quality</th>
        <th>Update</th><th>Deleate</th>
    </tr>
    <?php while ($row = mysqli_fetch_assoc($result)) { ?>
    <tr>
        <td><?=$row['productName'];?></td><td><?=$row['price'];?></td><td><?=$row['qty'];?></td>
        <td>
            <form action="update_form.php" method="post">
            <input type="hidden" name="productID" value="<?=$row['productID'];?>">

            <input type="submit" name="submit" value="Update">
            </form>
        </td>
        <td>
            <form action="delete.php" method="post">
            <input type="hidden" name="productID" value="<?=$row['productID'];?>">
            <input type="submit" name="submit" value="Delete">
            </form>
        </td>
    </tr>
    <?php } ?>
</table>
```

display_product.php

```php
<table border="1">
    <tr>
        <th>Product Name</th><th>Price</th><th>Quality</th>
        <th>Update</th><th>Deleate</th>
    </tr>
    <?php while ($row = mysqli_fetch_assoc($result)) { ?>
    <tr>
        <td><?=$row['productName'];?></td><td><?=$row['price'];?></td><td><?=$row['qty'];?></td>
        <td>
            <form action="update_form.php" method="post">
                <input type="hidden" name="productID" value="<?=$row['productID'];?>">
                <input type="hidden" name="productName" value="<?=$row['productName'];?>">
                <input type="hidden" name="price" value="<?=$row['price'];?>">
                <input type="hidden" name="qty" value="<?=$row['qty'];?>">
                <input type="submit" name="submit" value="Update">
            </form>
        </td>
        ...
    </tr>
    <?php } ?>
</table>
```

update_form.php

```php
<?php

require_once "config.php";

$productID = $_POST["productID"];
$productName = $_POST["productName"];
$price = $_POST["price"];
$qty = $_POST["qty"];

?>

...
```

```
...
<form method="POST" action="update.php">
    <input type="hidden" name="productID" value="<?=$productID; ?>">
    <label>Product name:</label>
    <input type="text" name="productName" value="<?=$productName; ?>"><br>
    <label>Price:</label>
    <input type="text" name="price" value="<?=$price; ?>"><br>
    <label>Qulity:</label>
    <input type="text" name="qty" value="<?=$qty; ?>"><br>
    <input type="submit" name="" value="submit">
</form>
...
```

Updating data in a table

☐ Changing data that you have already inserted is also quite simple, as in the example below:

UPDATE product SET productName ='Iphone' WHERE productName='Smartphone';

UPDATE travel SET destination ='Osaka' WHERE destination='Tokyo';

UPDATE travel SET destination ='Osaka' WHERE productID=1;

```php
<?php
require_once "config.php";

$productID = $_POST["productID"];
$productName = $_POST["productName"];
$price = $_POST["price"];
$qty = $_POST["qty"];

$userQuery = "UPDATE `product` SET productName='$productName',price='$price',qty='$qty'  WHERE `product`.`productID`=$productID";

$result = mysqli_query($connect, $userQuery);

if (!$result) {
      die ("Could not successfully run the query $userQuery ".mysqli_error($connect));
}
else {
      echo "Update successfully.<br><b>";
      echo "<a href='display_product.php'>Display product</a><br><br>";
}
mysqli_close($connect);
?>
```

Deleting a record

☐ When you need to remove a row from a table, use the DELETE command. This command allows you to narrow down the exact row or rows to delete using qualifiers such as WHERE and LIMIT.

☐ The first example issues a DELETE command for all rows whose title column contains the string Little Dorrit.

☐ The second example issues a DELETE command for all rows whose productid column contains the $productID

```
DELETE FROM classics WHERE title='Little Dorrit';
```

```
DELETE FROM product WHERE productID='$productID';
```

```php
<table border="1">
    <tr>
        <th>Product Name</th><th>Price</th><th>Quality</th>
        <th>Update</th><th>Deleate</th>
    </tr>
    <?php while ($row = mysqli_fetch_assoc($result)) { ?>
    <tr>
        <td><?=$row['productName'];?></td><td><?=$row['price'];?></td><td><?=$row['qty'];?></td>
        …

        <td>
            <form action="delete.php" method="post">
                <input type="hidden" name="productID" value="<?=$row['productID'];?>">
                <input type="submit" name="submit" value="Delete">
            </form>
        </td>
    </tr>
    <?php } ?>
</table>
```

```php
<?php
require_once "config.php";

$productID = $_POST["productID"];

$userQuery = "DELETE FROM `product` WHERE `product`.`productID`=$productID";

$result = mysqli_query($connect, $userQuery);

if (!$result) {
    die ("Could not successfully run the query $userQuery ".mysqli_error($connect));
}
else {
    echo "Delete successfully.<br><b>";
    echo "<a href='display_product-2.php'>Display product</a><br><br>";
}

mysqli_close($connect);
?>
```