Database concepts (CBD)

Class 7

Linking a database to a PHP document

Of course, MyPHPadmin can be use to modify a database and to make different queries. But the real purpose of a database is to be used online to supply various data to be displayed in web documents. It is then necessary to link the database to a web document so requests can be made from the web document. In order to use a database in a web document, the database must first me connected and the requests allowed. Then requests can be made to finally disconnect the database.

Connecting to a MySQLi database in PHP

MySQLi is an improved version of MySQL declared deprecated in 2011 and removed from PHP 5.5. MySQLi and PDO are object oriented that should now be used to prevent your application from *SQL injection*.

```
<?php
$db = new mysqli('localhost', 'root', 'root', 'My_test_DB');          // Server connection info

if($db->connect_errno > 0){
        die('Error : (. $db->connect_errno .') '. $db->connect_error);   // Error message
}

echo 'Connected successfully.';                                       // Success message
$mysqli->close();                                                     // Closing connection
?>
```

```
<?php
DEFINE('DB_USERNAME', 'root');                  // root or the predefined user name
DEFINE('DB_PASSWORD', 'root');                  // root or the predefined password
DEFINE('DB_HOST', 'localhost');                 // localhost when using MAMP
DEFINE('DB_DATABASE', 'My_test_DB');            // Name of the database

$mysqli = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_DATABASE);

if (mysqli_connect_error()) {
        die('Connect Error ('.mysqli_connect_errno().') '.mysqli_connect_error());
}

echo 'Connected successfully.';
$mysqli->close();
?>
```

Explanation :
In the example above, *mysqli()* function is used to open a connection using the database's host, user name, password and named defined at the very beginning of the script. An error message is defined in a conditional structure using *mysqli_connect_error()* and *die()*. and the connection is closed using the *close()* function after showing the success message.

Retrieving content from a database (object oriented way)

```
<?php
$db = new mysqli('localhost', 'root', 'root', 'My_test_DB');        // Connection stored in variable

if ($db->connect_error) {
        die("Connection failed: " . $conn->connect_error);          // Error message
}

$sql = "SELECT * FROM library";                                     // Prepared statement
$result = $db->query($sql);                                         // Query stored in variable

if ($result->num_rows > 0) {                                        // If there is at least 1 row
        while($row = $result->fetch_assoc()) {                      // Output data of each row
                echo $row["title"]  . "<br>";
        }
} else {
        echo "0 results";                                           // If there is less than 1 row
}

$db->close();                                                       // Closing connection
?>
```

Explanation:
In the example above, after connecting to the database and having defined the error message, the query is set up and stored in the variable ***$sql*** (in this case all the data from the table *library*). The query is then run and the result is stored in the variable ***$result***. Once that is done, the function ***num_rows()*** is used to check if there is at least one row returned. If there are more than zero rows returned, the function ***fetch_assoc()*** is used to store all the results into an *associative array* named ***$row***. Finally, a while() loop goes through the result set and outputs the data from the ***title*** column.

Retrieving content from a database (procedural way)

```
<?php
...

$result = mysqli_query($db, $sql);

if (mysqli_num_rows($result) > 0) {
        while($row = mysqli_fetch_assoc($result)) {
        echo $row["title"] . "<br>";
        }
} else {
        ...
```

Displaying and ordering the data

```
<?php
$db = new mysqli('localhost', 'root', 'root', 'My_test_DB');

if ($db->connect_error) {
        die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT title, author, year FROM library ORDER BY year";
$result = mysqli_query($db, $sql);

if (mysqli_num_rows($result) > 0) {
        while($row = mysqli_fetch_assoc($result)) {
        echo "<b>" . $row["title"] . "</b>, " . $row["author"] . ", (" . $row["year"] . ")<br>";
        }

} else {
        echo "0 results";
}

$db->close();
?>
```

Explanation :
As it is shown in the example above, it is simply needed to concatenate the columns and strings to extract the wanted data. To order the data based on a given filed, simply use the *ORDER BY* command in the query.

Connecting and retrieving data from a database in PHP using PDO

Standing for PHP Data Objects. PDO is another (easier) way to access databases. It consists in a set of PHP extensions making it easier to interact with databases.

```
<?php
$username = root;
$password = root;
$db = new PDO("mysql:host=localhost; dbname=My_test_DB", $username, $password);
$table = 'library';
$stmt = $db->query('SELECT * from '.$table);
$db = NULL;

while($rows = $stmt->fetch()){
        echo $rows['title'];
};
?>
```