Manipulate a MySQL database

Open a connection to the MySQL server

First, we will see how to connect to MySQL.
For this, we will use the function. **mysqli_connect ()**

There is another way to connect to a database that we will discuss later.

Here is an example of using this function:

```
<? Php
        $ server = 'localhost';
    $ user = 'root';
        $ password = 'password';

   mysqli_connect ($ server, $ user, $ password);

?>
```

The function expects 3 arguments:

- the address of the server on which your database is located
- the user to access this database
- his password

Require & Define

The function **define ()** is used to define a constant.
A constant is similar to a variable except that it can not change value after initializing it. Constants can be passed anywhere and regardless of the scope.

Instead of having our credentials to our server directly in each PHP file, we will define them in a separate file.

We will create a 'configure.php' file with this content:

```
<? Php
        define ('DB_USER', 'root');
        define ('DB_PASS', 'password');
        define ('DB_SERVER', 'localhost');
?>
```

Then we will include this configuration file in our original file.
For this, we will use **require.**

Require is seems to **include** because it allows you to include a file in the code. The difference is when an error occurs, include will continue to execute the script while require will stop the execution of this script.

We will now use require and call our constants this way:

```
<? Php
        require 'configure.php';

        mysqli_connect (DB_SERVER, DB_USER, DB_PASS);
?>
```

Selecting a database

Now that we are connected to the server, we will have to choose which table to work on using the **mysqli_select_db () function**.

Here is a piece of functional code:

```
<? Php
            // I get the info to login
      require 'configure.php';

            // I connect to my server
            $ db_handle = mysqli_connect (DB_SERVER,
                              DB_USER, DB_PASS);

            echo "Connection to server open! <br>";

            // I chose the base I'm working on
            $ db_name = "ifa";
            $ db_found = mysqli_select_db ($ db_handle,
                              $ db_name);

            if ($ db_found) {
            echo $ db_name. 'found!' ;
            } else {
            echo $ db_name. 'not found☹ !' ;
            }
      ?>
```

Let's analyze this code. At first we have this line:

```
      $ db_handle = mysqli_connect (DB_SERVER,
DB_USER, DB_PASS);
```

We save the connection to our server in a variable.
Then, we choose on which basis to work:

```
      $ db_found = mysqli_select_db ($ db_handle,
                              $ db_name);
```

This function will return true / false. True if she finds the base, false if she did not find it.

Closing the connection

Now that we have seen how to open a MySQL connection, we have to see how to close this connection. It is important to close the connection because it uses the memory resource.

Here is the syntax for closing a connection:

```
mysqli_close ($ db_handle);
```

Reading the result of a query

We will now see how to read the result of a SELECT query.
For that, it is necessary to use the functions **mysqli_query ()** and **mysqli_fetch_assoc ()**

The function **mysqli_query ()** is used to send an SQL request to our DB. This function will return a result of type handle, either true or false. When using a SELECT query, it returns a result of type handle (it saves all the results returned by the query).

Then we use thefunction **mysqli_fetch_assoc ()** to retrieve the results as a table. Whenever this function is called, the next record is retrieved.

Here is an example of a functional code:

```
<? Php
        // I get the informations to log in
require 'configure.php';

        // I connect to my server
        $ db_handle = mysqli_connect (DB_SERVER,
                        DB_USER, DB_PASS);

        echo "Connection to server open! <br>";

        // I chose the base I'm working on
        $ db_name = "ifa";
        $ db_found = mysqli_select_db ($ db_handle,
                        $ db_name);
```

```
if ($ db_found) {
// I prepare my query
$ sql_query = "SELECT * FROM movies";

// I run my query
$ result_query = mysqli_query ($ db_handle, $ sql_query);

/ *
        I retrieve the result as a
        table anddisplay
II use a loop to get
        ALL records.
        * /

        while ($ db_field = mysqli_fetch_assoc ($ result_query))
{
        echo "<hr>";
        echo $ db_field ['movie_id'] "<br>";
        echo $ db_field ['title']. "<br>";
        echo $ db_field ['release_year']
                . "<br>";
        echo $ db_field ['views']. "<br>";
}
} else {
        echo $ db_name. 'not found☹ !' ;
}
?>
```

Add a record (insert)

To add a record, the way is similar to a ready difference.

Thefunction **mysqli_query ()** will return a true or false result. It returns true when the query has run, it returns false when it failed.

Here is an example of a functional code:

```
<? Php
        // I get the informations to log in
require 'configure.php';

        // I connect to my server
        $ db_handle = mysqli_connect (DB_SERVER,
                        DB_USER, DB_PASS);

        echo "Connection to server open! <br>";

        // I chose the base I'm working on
        $ db_name = "ifa";
        $ db_found = mysqli_select_db ($ db_handle,
                        $ db_name);

        if ($ db_found) {
        // I prepare my query
        $ sql_query = " INSERT INTO actors (name,
                    date_of_birth, nationality)
                    VALUES ( 'Tom Cruise', '1962-08-03',
                    'USA') ";

        // I run my query
        = $ result_query mysqli_query ($ db_handle, $ sql_query);

        if ($ result_query) {
                echo "Record added to the database";
                }

        }
        } else {
                echo $ db_name. 'not found☹ !' ;
        }
?>
```

Edit or delete a record

To modify or delete a record (UPDATE or DELETE), this is done in the same way as an INSERT statement.