

PDO - PHP Data Object

When working with PHP and databases, the PHP provides several functions within the *mysqli* class for MySQL Database and *pg_** function for PostgreSQL Databases. The functions aren't interchangeable. So, if there is a need to change the database, we have to change every function in all the code. To prevent that and get ourself a cleaner, correct and safety code, we have to use the *PDO* class.

To connect to a database, we have to define which DBMS to use, port, username, password and name of the database. Which leads us to:

```
$dsn = "mysql:host=127.0.0.1;port=3307;dbname=test";
$username = "root";
$password = "";
try {
    $conn = new PDO($dsn, $username, $password);
    echo "Connected!";
} catch (PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
```

Gives the following result:

Connected!

The table to use as example in MySQL DBMS is:

```
CREATE TABLE `test`.`client` (
  `id_client` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `address` VARCHAR(100) NOT NULL,
  `age` INT NOT NULL,
  `gender` INT NOT NULL COMMENT '(0 - Male, 1 - Female)',
  `document_id` INT NOT NULL,
  `telephone` VARCHAR(15) NOT NULL,
  `mobile_fone` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`id_client`));
```

And add values to the table as:

```
INSERT INTO client (name, address, age, gender, document_id, telephone, mobile_fone) VALUES
("<name>", "<address>", <age>, <0 or 1 for gender>, "<document identificator>", "<telephone
number>", "<mobile number>");
```

To print all the data in this table, we can use PDO as following:

```
try {
    $results = $conn->query("SELECT * FROM client");
    foreach ($results as $row) {
        print_r($row);
        print_r("<br>");
    }
} catch (PDOException $e) {
    echo "Something gone wrong: " . $e->getMessage();
}
```

Gives the result below:

```
Array ( [id_client] => 1 [0] => 1 [name] => Danielson [1] => Danielson [address] => 455 E DRAGAM
[2] => 455 E DRAGAM [age] => 30 [3] => 30 [gender] => 0 [4] => 0 [document_id] => 8874432 [5] =>
8874432 [telephone] => +558420201521 [6] => +558420201521 [mobile_fone] => +5584988662397 [7]
=> +5584988662397 )
Array ( [id_client] => 2 [0] => 2 [name] => Angela [1] => Angela [address] => 887 W MAIN ST [2] =>
887 W MAIN ST [age] => 18 [3] => 18 [gender] => 1 [4] => 1 [document_id] => 112345 [5] => 112345
[telephone] => +3557499881124 [6] => +3557499881124 [mobile_fone] => +3557499881124 [7] =>
+3557499881124 )
Array ( [id_client] => 3 [0] => 3 [name] => Jonh [1] => Jonh [address] => 4211 N BOLYSTON AVE [2]
=> 4211 N BOLYSTON AVE [age] => 23 [3] => 23 [gender] => 0 [4] => 0 [document_id] => 556123 [5]
=> 556123 [telephone] => +672449987751 [6] => +672449987751 [mobile_fone] => +6724420203535
[7] => +6724420203535 )
Array ( [id_client] => 4 [0] => 4 [name] => Jamela [1] => Jamela [address] => 766 E MAIN ST [2] =>
766 E MAIN ST [age] => 35 [3] => 35 [gender] => 1 [4] => 1 [document_id] => 9941154 [5] =>
9941154 [telephone] => +614499876547 [6] => +614499876547 [mobile_fone] => +614432320122 [7]
=> +614432320122 )
```

To use dynamic values into a query, the correct way to do it is using statements. We can use statements in three different ways.

(1) - Ways using an array of parameters:

```
try {
    $id = 1;
    $statement = $conn->prepare("SELECT * FROM client WHERE id_client = ? AND age = ?");
    $statement->execute(array($id, 30));
    $results = $statement->fetchAll();
    foreach ($results as $row) {
        print_r($row);
        print_r("<br>");
    }
} catch (PDOException $e) {
    echo "Something gone wrong: " . $e->getMessage();
}
```

Results:

```
Array ( [id_client] => 1 [0] => 1 [name] => Danielson [1] => Danielson [address] => 455 E DRAGAM
[2] => 455 E DRAGAM [age] => 30 [3] => 30 [gender] => 0 [4] => 0 [document_id] => 8874432 [5] =>
8874432 [telephone] => +558420201521 [6] => +558420201521 [mobile_fone] => +5584988662397 [7]
=> +5584988662397 )
```

(2) - Ways using an array of named parameters:

```
try {
    $id = 1;
    $statement = $conn->prepare("SELECT * FROM client WHERE id_client = :id AND age = :age");
    $statement->execute(array(
        ":id" => $id,
```

```

        ":age" => 30
    ));
    $results = $statement->fetchAll();
    foreach ($results as $row) {
        print_r($row);
        print_r("<br>");
    }
} catch (PDOException $e) {
    echo "Something gone wrong: " . $e->getMessage();
}

```

Results:

```

Array ( [id_client] => 1 [0] => 1 [name] => Danielson [1] => Danielson [address] => 455 E DRAGAM
[2] => 455 E DRAGAM [age] => 30 [3] => 30 [gender] => 0 [4] => 0 [document_id] => 8874432 [5] =>
8874432 [telephone] => +558420201521 [6] => +558420201521 [mobile_fone] => +5584988662397 [7]
=> +5584988662397 )

```

(3) - Ways using **bind** methods:

```

try {
    $id = 1;
    $statement = $conn->prepare("SELECT * FROM client WHERE id_client = :id AND age = :age");
    $statement->bindParam(":id", $id);
    $statement->bindValue(":age", 30);
    $statement->execute();
    $results = $statement->fetchAll();
    foreach ($results as $row) {
        print_r($row);
        print_r("<br>");
    }
} catch (PDOException $e) {
    echo "Something gone wrong: " . $e->getMessage();
}

```

Results:

```

Array ( [id_client] => 1 [0] => 1 [name] => Danielson [1] => Danielson [address] => 455 E DRAGAM
[2] => 455 E DRAGAM [age] => 30 [3] => 30 [gender] => 0 [4] => 0 [document_id] => 8874432 [5] =>
8874432 [telephone] => +558420201521 [6] => +558420201521 [mobile_fone] => +5584988662397 [7]
=> +5584988662397 )

```