

# PHP-DB

A. Atlas

# Introduction

PHP permet de manipuler les bases de données MySQL en utilisant:

- L'extension `mysql_` : Fonctions permettant d'accéder et manipuler une base de données MySQL. Leur nom commence par `mysql_`.
  - Vieilles fonctions « obsolètes »
  - Utilisation déconseillée
- L'extension `mysqli_` (le "i" improved) : Fonctions améliorées d'accès à MySQL. Elles proposent plus de fonctionnalités.
- L'extension PDO (PHP Data Objects). un outil complet qui permet d'accéder à n'importe quel type de base de données.

# Créer une Base de données

The screenshot shows the phpMyAdmin interface in a browser window. The address bar shows the URL: localhost/phpmyadmin/index.php?route=/server/databases&server=1. The interface includes a navigation menu on the left with options like 'Recent' and 'Favorites', and a main content area titled 'Databases'. In the 'Databases' section, there is a 'Create database' form with a text input field containing 'gidb' and a dropdown menu set to 'utf8mb4\_general\_ci'. A 'Create' button is next to the form. Below the form is a table listing existing databases with their collations and actions.

Database	Collation	Action
<input type="checkbox"/> information_schema	utf8_general_ci	<a href="#">Check privileges</a>
<input type="checkbox"/> mysql	utf8mb4_general_ci	<a href="#">Check privileges</a>
<input type="checkbox"/> performance_schema	utf8_general_ci	<a href="#">Check privileges</a>
<input type="checkbox"/> phpmyadmin	utf8_bin	<a href="#">Check privileges</a>
<input type="checkbox"/> test	utf8mb4_general_ci	<a href="#">Check privileges</a>

**Total: 5**

Check all    With selected: [Drop](#)

**Note:** Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

- **Enable statistics**

CREATE DATABASE gidb



Recent Favorites

- New
- [-] gidb
  - New
  - + user
- + information\_schema
- + mysql
- + performance\_schema
- + phpmyadmin
- + test

Server: localhost » Database: gidb » Table: user

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#)

[Table structure](#) [Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 <b>username</b>	varchar(20)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	2 <b>email</b>	varchar(20)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 <b>password</b>	varchar(40)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 <b>score</b>	int(11)			No	None			Change  Drop  More

Check all    With selected: [Browse](#) Change Drop Primary Unique Index Spatial  
[Add to central columns](#) [Remove from central columns](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

Add  column(s)

[Indexes](#)

# PDO: Se connecter à une base de données

- Pour Se connecter à une base de données avec PDO nous allons avoir besoin de quatre renseignements :
  - Le nom de l'hôte : c'est l'adresse du serveur de BD MySQL
  - La base : c'est le nom de la base de données
  - Le login : il permet de vous identifier
  - Le mot de passe :
- Exemple :

```
try {  
    $bdd = new PDO('mysql:host=localhost;dbname=gi_db', 'root', '');  
} catch(PDOException $e) {  
    echo $e->getMessage();  
}
```

# PDO: Executer une requete

L'objectif consiste à récupérer la liste des enregistrements sous forme d'un tableau associatif.

```
<?php
try {
    $db = new PDO('mysql:host=localhost;dbname=gidb', 'root', '');
    $req = $db->prepare('SELECT * FROM user');
    $req->execute();
    $res = $req->fetchAll();
    foreach ($res as $r) {
        print_r(r);
    }
}
```

## PDO: Executer une requete

- Pour construire une requête en fonction de la valeur d'une variable, on passe par des *marqueurs* qui permettent d'éviter les dangereuses failles d'injection SQL.
- Les marqueurs sont des identifiants reconnus par PDO pour être remplacés lors de la préparation de la requête par les variables PHP
- On ne concatène JAMAIS une requête SQL pour passer des variables, au risque de créer des injections SQL !

```
<?php
    try {
        $db = new PDO('mysql:host=localhost;dbname=gidb', 'root', '');
        $req = $db->prepare('SELECT * FROM user WHERE username=:u');
        $req->execute(['u'=>'mohammed']);
        $res = $req->fetchAll();
        foreach ($res as $r) {
            ...
        }
    }
?>
```

## Insertion

```
<?php
    try {
        $db = new PDO('mysql:host=localhost;dbname=gidb', 'root', '');
        $sql = 'INSERT INTO user(username, password, email) VALUES (:u, :p,
:e)';

        $req = $db->prepare($sql);
        $req->execute(['u'=>'amale', 'p'=>'1222', 'e'=>'a@a.a']);
    } catch(PDOException $e) {
        echo $e->getMessage();
    }
?>
```



# Update

```
<?php
    try {
        $db = new PDO('mysql:host=localhost;dbname=gidb', 'root', '');
        $sql = 'UPDATE user SET password= :p, email= :e WHERE username = :u';
        $req = $db->prepare($sql);
        $req->execute(['u'=>'amale','p'=>'0000','e'=>'a@a.a']);
    }catch(PDOException $e){
        echo $e->getMessage();
    }
?>
```

# DELETE

```
<?php
    try {
        $db = new PDO('mysql:host=localhost;dbname=gidb', 'root',
            '');

        $sql = 'DELETE FROM user WHERE username = :u';
        $req = $db->prepare($sql);
        $req->execute(['u'=>'amale']);
        $res = $req->fetchAll();

    } catch(PDOException $e) {
        echo $e->getMessage();
    }
}
```