

# Base de données sur lequel on travail

← → ↻  ☆

**phpMyAdmin**

Récentes Préférées

- Nouvelle base de données
- information\_schema
- mydatabase
- mysql
- performance\_schema
- phpmyadmin
- store
  - Nouvelle table
  - customers
  - customer\_addresses
  - orders
  - products
  - product\_categories
  - reviews
- test

Serveur: localhost » Base de données: store

Structure SQL Rechercher Requête Exporter Importer Opérations Privilèges Procédures stockées

Filtres

Contenant le mot :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> customers	★ Parcourir Structure Rechercher Insérer Vider Supprimer	24	InnoDB	latin1_swedish_ci	16 kio	-
<input type="checkbox"/> customer_addresses	★ Parcourir Structure Rechercher Insérer Vider Supprimer	19	InnoDB	latin1_swedish_ci	32 kio	-
<input type="checkbox"/> orders	★ Parcourir Structure Rechercher Insérer Vider Supprimer	29	InnoDB	latin1_swedish_ci	48 kio	-
<input type="checkbox"/> products	★ Parcourir Structure Rechercher Insérer Vider Supprimer	12	InnoDB	latin1_swedish_ci	32 kio	-
<input type="checkbox"/> product_categories	★ Parcourir Structure Rechercher Insérer Vider Supprimer	6	InnoDB	latin1_swedish_ci	16 kio	-
<input type="checkbox"/> reviews	★ Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	latin1_swedish_ci	48 kio	-
<b>6 tables</b>	<b>Somme</b>	<b>94</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>192 kio</b>	<b>0 o</b>

↑ ☐ Tout cocher Avec la sélection :

Imprimer Dictionnaire de données

## Configuration des droits utilisateur dans la bd.

← Serveur: localhost

Bases de données

SQL

État

Comptes utilisateurs

Exporter

Importer

Paramètres

Réplication

Aperçu des comptes utilisateurs

Groupes d'utilisateurs

### Aperçu des comptes utilisateurs

⚠

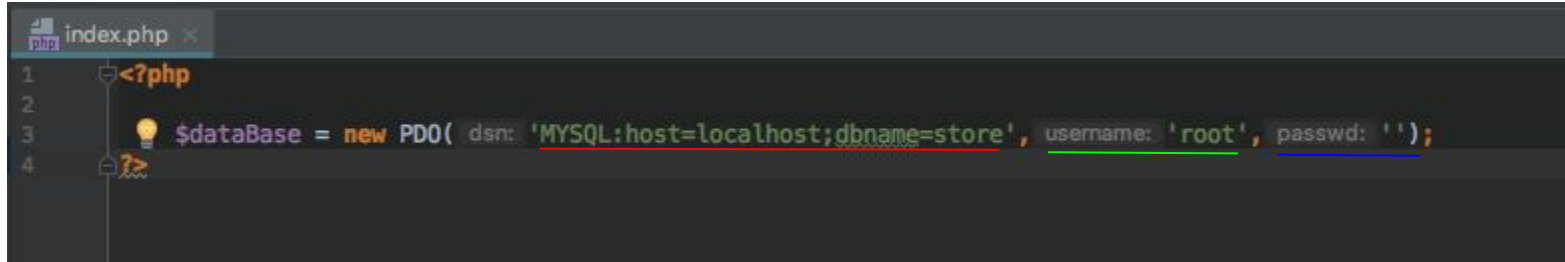
Un compte d'utilisateur permettant à tout utilisateur de se connecter depuis « localhost » est présent. Cela empêchera les autres utilisateurs de se connecter. Ce compte permet une connexion de n'importe quel hôte (%).

?

	Nom d'utilisateur	Nom d'hôte	Mot de passe	Privilèges globaux	Groupe d'utilisateurs	« Grant »	Action
<input type="checkbox"/>	N'importe quel	%	Non	USAGE		Non	<div>Éditer les privilèges</div> <div>Exporter</div>
<input type="checkbox"/>	N'importe quel	localhost	Non	USAGE		Non	<div>Éditer les privilèges</div> <div>Exporter</div>
<input type="checkbox"/>	pma	localhost	Non	USAGE		Non	<div>Éditer les privilèges</div> <div>Exporter</div>
<input type="checkbox"/>	root	127.0.0.1	Non	ALL PRIVILEGES		Oui	<div>Éditer les privilèges</div> <div>Exporter</div>
<input type="checkbox"/>	root	::1	Non	ALL PRIVILEGES		Oui	<div>Éditer les privilèges</div> <div>Exporter</div>
<input type="checkbox"/>	root	localhost	Non	ALL PRIVILEGES		Oui	<div>Éditer les privilèges</div> <div>Exporter</div>

## Création de l'objet de connexion PDO

On doit fournir 3 arguments minimum:



```
index.php x
1 <?php
2
3 $dataBase = new PDO( dsn: 'MYSQL:host=localhost;dbname=store', username: 'root', passwd: '');
4
```

The screenshot shows a code editor with a file named 'index.php'. The code is written in PHP and demonstrates the creation of a PDO object. The DSN (Data Source Name) is 'MYSQL:host=localhost;dbname=store', the username is 'root', and the password is an empty string. The code is color-coded: 'MYSQL:host=localhost;dbname=store' is red, 'root' is green, and the empty string is blue.

Le nom de l'hôte;Le nom de la bd

Le nom du user

Le password

```
index.php x
1 <?php
2
3 $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
4
5 $results = $dataBase->query( statement: 'SELECT first_name FROM customers');
6
7 var_dump($results);
8
9
10
11
12
```

La méthode `->query()` est une méthode de la classe PDO qui permet de faire simplement une requête dans la base de donnée choisie.

On peut voir ce que retourne la méthode est un objet statement (une déclaration) de requête.

localhost/projetStore/php/

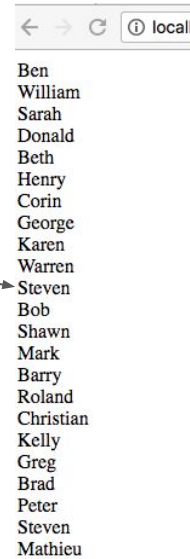
```
object(PDOStatement)#2 (1) { ["queryString"]=> string(33) "SELECT first_name FROM customers" }
```

▼ Expand All ▲ Collapse All

```
▼ PDOStatement (object) [Object ID #2][1 property]
  queryString: (string) SELECT first_name FROM customers
```

On doit appliquer la méthode ->fetch() pour aller chercher la data de notre statement \$results se que l'on récupère est toujours un tableau donc **il faut l'appeler en tant que tel.**

```
index.php x
1 <?php
2
3 $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
4
5 $results = $dataBase->query( statement: 'SELECT first_name FROM customers');
6
7 $myData = $results->fetch();
8
9 while($row = $results->fetch()){ // tant que result peut faire la methode fetch
10     echo $row['first_name'].'<br>'; //sur de la donné et valorisier $row.
11 }
12
13
14
15
```

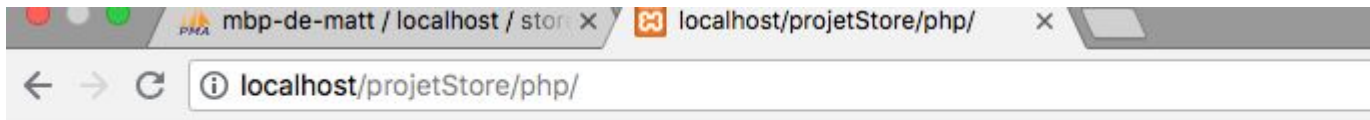


appel que d'une seul ligne car fetch() ne récupère qu'une ligne.

```
index.php x
1 <?php
2
3 $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
4
5 $results = $dataBase->query( statement: 'SELECT first_name FROM customers');
6
7 $myData = $results->fetch();
8
9 echo $myData['first_name'];
10
```

On utilise un try/catch pour se connecter à la bd dans le cas où la connexion échoue (pour la démo on a fait exprès de rajouter un mauvais mots de passe) on tue le programme avec la méthode die() et on récupère le message.

```
index.php x
1 <?php
2 |
3
4 try{
5     $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: 'd');
6 }catch(Exception $e){
7     die('ERROR: '.$e->getMessage());
8 }
9
10
11 $results = $dataBase->query( statement: 'SELECT first_name FROM customers');
12
13
14 while($row = $results->fetch()){ // tant que result peut faire la methode fetch
15     echo $row['first_name'].'<br>'; //sur de la donné et valorisier $row.
16 }
17
18 ?>
```



ERROR: SQLSTATE[HY000] [1045] Access denied for user 'root'@'localhost' (using password: YES)