

Bases de données et PDO

<https://github.com/heig-vd-progserv-course/heig-vd-progserv1-course>

Visualiser le contenu complet sur GitHub [à cette adresse](#).

L. Delafontaine, avec l'aide de [GitHub Copilot](#).

Ce travail est sous licence [CC BY-SA 4.0](#).

Retrouvez le contenu complet de cette présentation sur GitHub

*Cette présentation est un résumé du contenu complet disponible
sur GitHub.*

*Pour plus de détails, retrouvez le [contenu complet sur GitHub](#) ou en
cliquant sur l'en-tête de ce document.*

Objectifs (1)

- Expliquer les concepts de base des bases de données et des systèmes de gestion de base de données (SGBD)
- Utiliser l'extension PDO de PHP pour interagir avec une base de données
- Créer une base de données SQLite et des tables avec PDO



Objectifs (2)

- Insérer, mettre à jour, récupérer et supprimer des données dans une base de données SQLite avec PDO



Base de données

- Ensemble d'informations organisées de manière à permettre un accès rapide et efficace à ces informations
- Généralement organisées en tables, qui contiennent des lignes et des colonnes :
 - Lignes : enregistrements uniques
 - Colonnes : champs de données



Systeme de gestion de base de données (SGBD)

- Logiciel permettant de créer, gérer et interroger des bases de données
- Fournit une interface pour interagir avec les données (création, insertion, mise à jour, suppression)
- Chacun avec ces caractéristiques
mais avec les mêmes concepts de



SQLite

- SGBD léger et intégré
- Pas besoin d'installation ou de configuration complexe
- Stocke les données dans un fichier unique
- Prise en charge de SQL standard
- Idéal pour les applications de petite à moyenne taille



PDO

- PHP Data Objects (PDO) est une extension de PHP qui fournit une interface orientée objet pour accéder à des bases de données
- Permet d'interagir avec différentes bases de données (MySQL, PostgreSQL, SQLite, etc.)
- Fournit une API cohérente pour effectuer des opérations sur des base de données



Connexion à une base de données SQLite

```
<?php
// Chemin vers le fichier de base de données SQLite
const DATABASE_FILE = 'path/to/database.db';

// Création d'une instance de PDO pour se connecter à la base de données
$pdo = new PDO("sqlite:" . DATABASE_FILE);
```

Création d'une table

```
// Création d'une table `users`  
$sql = 'CREATE TABLE IF NOT EXISTS users (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name TEXT NOT NULL,  
    email TEXT NOT NULL UNIQUE  
)';  
  
// On exécute la requête SQL pour créer la table  
$pdo->exec($sql);
```

Insertion de données

```
// On définit la requête SQL pour ajouter un utilisateur
$sql = "INSERT INTO users (
    name,
    email
) VALUES (
    'John Doe',
    'john.doe@heig-vd.ch'
)";

// On exécute la requête SQL pour ajouter l'utilisateur
$pdo->exec($sql);
```

```
// On définit la requête SQL pour ajouter un utilisateur
$sql = "INSERT INTO users (
    name,
    email
) VALUES (
    'Jane Doe',
    'jane.doe@heig-vd.ch'
)";

// On exécute la requête SQL pour ajouter l'utilisateur
$pdo->exec($sql);
```

Bases de données et PDO

```
// On récupère l'identifiant de l'utilisateur inséré
$janeDoeId = $pdo->lastInsertId();

// On affiche l'identifiant de l'utilisateur inséré
echo "L'identifiant de l'utilisateur inséré est : $janeDoeId<br>";
```

Récupération de données

```
// On définit la requête SQL pour récupérer l'utilisateur `Jane Doe`  
$sql = "SELECT * FROM users WHERE id = '$janeDoeId'";  
  
// On récupère l'utilisateur spécifique  
$user = $pdo->query($sql);  
  
// On transforme le résultat en tableau associatif  
$user = $user->fetch();  
  
// On affiche l'utilisateur  
print_r($user);
```

Bases de données et PDO

```
// On définit la requête SQL pour récupérer tous les utilisateurs
$sql = "SELECT * FROM users";

// On récupère tous les utilisateurs
$users = $pdo->query($sql);

// On transforme le résultat en tableau associatif
$users = $users->fetchAll();

// On affiche les utilisateurs
print_r($users);
```

Mise à jour de données

```
// On définit la requête SQL pour mettre à jour l'utilisateur `Jane Doe`  
$sql = "UPDATE users SET  
    name = 'Jane Smith',  
    email = 'jane.smith@heig-vd.ch'  
WHERE id = '$janeDoeId'";  
  
// On exécute la requête SQL pour mettre à jour l'utilisateur  
$pdo->exec($sql);
```

Bases de données et PDO

```
// On récupère l'utilisateur mis à jour
$sql = "SELECT * FROM users WHERE id = '$janeDoeId' ";
$user = $pdo->query($sql);
$user = $user->fetch();

// On affiche l'utilisateur mis à jour
print_r($user);
```

Suppression de données

```
// On définit la requête SQL pour supprimer l'utilisateur
$sql = "DELETE FROM users WHERE id = '$janeDoeId'";

// On exécute la requête SQL pour supprimer l'utilisateur
$pdo->exec($sql);

// On récupère tous les utilisateurs
$sql = "SELECT * FROM users";
$users = $pdo->query($sql);
$users = $users->fetchAll();

// On affiche les utilisateurs restants
print_r($users);
```

Conclusion

- Les bases de données permettent de stocker et de gérer des données
- SQLite est un système de gestion de base de données (SGBD)
- PDO est une extension PHP qui permet d'interagir avec SGBDs
- Les opérations de base incluent la création, l'insertion, la mise à jour et la suppression de données



Questions

Est-ce que vous avez des questions ?

À vous de jouer !

- (Re)lire le [support de cours](#).
- Explorer les [exemples de code](#).
- Faire les [exercices](#).
- Réaliser le [mini-projet](#).
- Poser des questions si nécessaire.

N'allez pas trop vite, vous avez le temps ! N'hésitez pas à vous entraidez si vous avez des difficultés.



Sources

- [Illustration principale](#) par [Richard Jacobs](#) sur [Unsplash](#)
- [Illustration](#) par [Aline de Nadai](#) sur [Unsplash](#)
- [Illustration](#) par [Jan Antonin Kolar](#) sur [Unsplash](#)
- [Illustration](#) par [israel palacio](#) sur [Unsplash](#)
- [Illustration](#) par [Nikita Kachanovsky](#) sur [Unsplash](#)