PHP - SQLite + PDO : Opérations CRUD

Cette fiche présente la mise en œuvre des opérations fondamentales `CRUD` (Créer, Lire, Mettre à jour, Supprimer) sur une base de données `SQLite` à l'aide de `PDO` en `PHP`. Elle met l'accent sur la rigueur, la sécurité et la clarté du code.

1 - Créer une table

Avant toute manipulation, il est nécessaire de disposer d'une `table` structurée dans la base `SQLite`. * La création s'effectue via une requête `CREATE TABLE` exécutée par `PDO`. * Il est recommandé de prévoir une clé primaire et des types adaptés à chaque colonne.

Exemple :

```php
$pdo->exec("CREATE TABLE IF NOT EXISTS utilisateurs (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  nom TEXT,
  email TEXT
)");
```

Insérer des données dans une table

L'insertion de données dans une `table` s'effectue à l'aide d'une requête `INSERT INTO`. * Il est impératif d'utiliser des **`requêtes préparées`** pour garantir la sécurité et l'intégrité des données. * Les paramètres sont liés à la requête pour éviter les injections SQL. * Il est conseillé de valider les données avant l'insertion pour éviter les erreurs.

```php
$nom = "Alice";
$email = "alice@mail.com";

$stmt = $pdo->prepare("INSERT INTO utilisateurs (nom, email) VALUES (:nom, :email)");

$stmt->bindParam(':nom', $nom, PDO::PARAM_STR);
$stmt->bindParam(':email', $email, PDO::PARAM_STR);
$stmt->execute();
```

Supprimer des données

La suppression d'un enregistrement s'effectue à l'aide d'une requête `DELETE`. * Il est impératif de spécifier une condition `WHERE` pour éviter la suppression de toutes les données. * L'utilisation de **`requêtes préparées`** est recommandée pour sécuriser l'opération.

Exemple :

```php
$id = 1;
$stmt = $pdo->prepare("DELETE FROM utilisateurs WHERE id = :id");
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->execute();
```

Lire des données depuis une table

La lecture des données s'effectue à l'aide d'une requête `SELECT`. * L'utilisation de `PDOStatement` permet de parcourir les résultats de manière sécurisée et efficace. * Il est conseillé d'utiliser le mode `PDO::FETCH_ASSOC` pour obtenir des tableaux associatifs.

Exemple :

```php
$stmt = $pdo->query("SELECT * FROM utilisateurs");
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    echo "Nom: " . $row['nom'] . ", Email: " . $row['email'] . "<br>";
}
```

Mettre à jour des données

La modification des données s'effectue via une requête `UPDATE`. * Il est essentiel de cibler précisément les enregistrements à modifier à l'aide d'une clause `WHERE`. * Les **requêtes préparées** assurent la sécurité de l'opération.

Exemple :

```php
$id = 1; // ID de l'enregistrement à mettre à jour
$nom = "Bob";
$email = "bob@mail.com";
$stmt = $pdo->prepare("UPDATE utilisateurs SET nom = :nom, email = :email WHERE id = :id");
$stmt->bindParam(':nom', $nom, PDO::PARAM_STR);
$stmt->bindParam(':email', $email, PDO::PARAM_STR);
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->execute();
```

Supprimer des données

La suppression d'un enregistrement s'effectue à l'aide d'une requête `DELETE`. * Il est impératif de spécifier une condition `WHERE` pour éviter la suppression de toutes les données. * L'utilisation de **requêtes préparées** est recommandée pour sécuriser l'opération.

Exemple :

```php
$id = 1;
$stmt = $pdo->prepare("DELETE FROM utilisateurs WHERE id = :id");
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->execute();
```

À RETENIR

Points essentiels :

- Les opérations `CRUD` sont la base de toute interaction avec une base de données.
- L'utilisation de requêtes préparées avec `PDO` est indispensable pour la sécurité.
- Toujours cibler précisément les enregistrements lors des opérations de mise à jour ou de suppression.
- Le mode `PDO::FETCH_ASSOC` facilite la manipulation des résultats.

Concept	Description
`CRUD`	Ensemble des opérations de base : Créer, Lire, Mettre à jour, Supprimer.
`PDO`	Interface d'abstraction de base de données en PHP.
`requête préparée`	Requête SQL sécurisée utilisant des paramètres nommés ou positionnels.
`PDOStatement`	Objet représentant une requête préparée ou exécutée.
`execute()`	Méthode pour exécuter une requête préparée.
`bindParam()`	Méthode pour lier des paramètres à une requête préparée.
`query()`	Méthode pour exécuter une requête SQL et retourner un objet `PDOStatement`.
`prepare()`	Méthode pour préparer une requête SQL avant son exécution.
`exec()`	Méthode pour exécuter une requête SQL sans retourner de résultats.
`INSERT INTO`	Instruction SQL pour insérer de nouvelles données.
`UPDATE`	Instruction SQL pour modifier des données existantes.
`DELETE`	Instruction SQL pour supprimer des données.
`WHERE`	Clause SQL permettant de cibler des enregistrements précis.
`FETCH_ASSOC`	Mode de récupération des résultats sous forme de tableau associatif.
`fetch()`	Méthode pour lire une ligne de résultat d'une requête SELECT.
`fetchAll()`	Méthode pour lire toutes les lignes de résultat d'une requête SELECT.