

- L'hôte : chaîne de caractères contenant le nom ou l'adresse IP de l'hôte. Celui-ci correspond à "localhost" ou 127.0.0.1 si vous travaillez en local.
- L'utilisateur : chaîne de caractères contenant le nom de l'utilisateur pour se connecter à la base de données. Si vous travaillez en local, celui-ci correspond à "root". Attention, cet utilisateur a tous les droits sur votre base de données.

- Le mot de passe : chaîne de caractères contenant le mot de passe associé à l'utilisateur. Celui-ci est vide par défaut.
- Le nom de la base de données : chaîne de caractères optionnelle contenant le nom de votre base de données.
- Le port : numéro du port optionnel pour se connecter au serveur MySQL. Celui-ci vaut 3306 par défaut.

Cette fonction retourne false en cas d'erreur ou un objet mysqli contenant l'identifiant de connexion en cas de succès.

Par exemple :

```
<?php
    $connect = mysqli_connect("127.0.0.1", "root", "", "bdd");
?>
```

Autre exemple en gérant les erreurs :

```
<?php

$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur:'.mysqli_get_host_info($base);
}
else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());
} //%d représente un décimal qui est le numéro d'erreur renvoyé par
mysqli connect errno. %s représente une chaîne de caractères qui est
le libellé de l'erreur renvoyé par mysqli connect error
?>
```

La fonction `mysqli_get_host_info($base)` retourne des informations sur votre serveur.

La fonction `mysqli_connect_errno()` retourne le numéro de l'erreur en cas d'échec de connexion.

La fonction `mysqli_connect_error()` retourne le message de l'erreur en cas d'échec de connexion.

Il est possible aussi de se connecter au serveur puis à la base de données grâce à la fonction `mysqli_select_db()`. Cette fonction prend en paramètre l'objet retourné par la fonction `mysqli_connect()`.

Par exemple :

```
<?php
    $base = mysqli_connect('127.0.0.1', 'root', '');
    mysqli_select_db($base, 'bdd') ;
?>
```

3. Déconnexion

La fonction permettant de se déconnecter de MySQL est `mysqli_close()`.

Par exemple :

```
if (mysqli_close($base)) {
    echo 'Déconnexion réussie.<br />';
}
else {
    echo 'Echec de la déconnexion.';
}
```

Il n'est pas obligatoire de fermer la connexion car PHP le fait automatiquement à la fin du script. Cependant, il est recommandé de le faire, cela augmentera la rapidité de vos scripts.

4. Requêtes non préparées

a. Lire des données

La fonction permettant d'exécuter une requête SQL est `mysqli_query()`.

Cette fonction prend en paramètres :

- L'objet de connexion : objet renvoyé par la fonction `mysqli_connect()`.
- La requête : chaîne de caractères contenant la requête SQL.

Cette fonction retourne `false` en cas d'échec et `true` ou un identifiant en cas de succès.

La fonction permettant de connaître le nombre de lignes dans le résultat de la requête est `mysqli_num_rows()`.

Cette fonction prend en paramètre le résultat renvoyé par la fonction `mysqli_query()` et retourne le nombre de lignes renvoyées par la requête SQL.

Par exemple, pour avoir le nombre de lignes dans la table `Personne` :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
    <head>
        <title>Exercice avec mysqli</title>
```

```

        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />

    </head>
<body>
<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).
'<br />';
    // Exécution de la requête
$resultat = mysqli_query($base, 'SELECT * FROM Personne');
if ($resultat == FALSE) {
    echo "Echec de l'exécution de la requête.<br />";
}
else {
    // Affichage du nombre de lignes
    echo 'Nombre de personnes : '.mysqli_num_rows($resultat).'<br />';
}

if (mysqli_close($base)) {
    echo 'Déconnexion réussie.<br />';
}
else {
    echo 'Echec de la déconnexion.';
}

}
else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());
}

?>
</body>
</html>

```

Affiche :

Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Nombre de personnes : 6
Déconnexion réussie.

Par la suite, le code HTML en début de page n'est plus écrit dans les exemples. Seul le code PHP est indiqué.

Maintenant, si vous voulez afficher les données de la table Personne, il faut utiliser la notion de fetch. Le fetch permet de lire la ligne courante et de se déplacer sur la ligne suivante.

Il existe plusieurs fonctions permettant d'utiliser le fetch.

La plus courante est la fonction `mysqli_fetch_assoc()`.

Cette fonction prend en paramètre le résultat renvoyé par la fonction `mysqli_query()` et retourne un tableau associatif ayant pour indice le nom de la colonne de la requête SQL.

Par exemple, pour afficher le nom et le prénom de la table Personne :

```
<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).
'<br />';
    // Exécution de la requête
    $resultat = mysqli_query($base, 'SELECT Nom,Prenom FROM Personne');
    if ($resultat == FALSE) {
        echo "Echec de l'exécution de la requête.<br />";
    }
    else {
        //fetch sur chaque ligne ramenée par la requête
        while ($ligne = mysqli_fetch_assoc($resultat)) {
            // Affichage du nom et prénom des personnes
            echo "Nom : ".$ligne['Nom']." et prénom : ".$ligne['Prenom']."<br />";
        }
    }

    if (mysqli_close($base)) {
        echo 'Déconnexion réussie.<br />';
    }
    else {
```

```

        echo 'Echec de la déconnexion.';
    }

}

else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());
}

?>

```

Affiche :

```

Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Nom : Dupont et prénom : Jean
Nom : Duval et prénom : Monique
Nom : Martin et prénom : Robert
Nom : Durand et prénom : Bernadette
Nom : Martineau et prénom : Alain
Nom : Roland et prénom : Fabien
Déconnexion réussie.

```

La variable `$ligne` est un tableau dont les indices sont les noms des champs ramenés par la requête. Ce tableau est écrasé par les nouvelles valeurs de chaque ligne. Le `fetch` permet de passer de ligne en ligne, et quand il n'y a plus de ligne, la condition de la boucle `while` est fausse donc le script sort de la boucle.

Une autre fonction permettant le `fetch` est `mysqli_fetch_row()`.

Cette fonction prend en paramètre le résultat renvoyé par la fonction `mysqli_query()` et retourne un tableau indexé ayant pour indice un compteur de 0 à n représentant les colonnes dans l'ordre ramenées par la requête.

Par exemple, pour afficher le nom et le prénom de la table `Personne` :

```

<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).
'<br />';
    // Exécution de la requête
$resultat = mysqli_query($base, 'SELECT Nom,Prenom FROM Personne');
if ($resultat == FALSE) {
    echo "Echec de l'exécution de la requête.<br />";
}
}

```

```

    }

    else {

        //fetch sur chaque ligne ramenée par la requête
        while ($ligne = mysqli_fetch_row($resultat)) {

            // Affichage du nom et prénom des personnes
            echo "Nom : ".$ligne[0]." et prénom : ".$ligne[1]."<br />";

        }

    }

    if (mysqli_close($base)) {

        echo 'Déconnexion réussie.<br />';

    }

    else {

        echo 'Echec de la déconnexion.';

    }

}

else {

    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());

}

?>

```

Affiche les mêmes informations que précédemment.

Cette fois le tableau contient la valeur du nom dans `$ligne[0]` et la valeur du prénom dans `$ligne[1]`. Cette fonction est moins pratique car il faut savoir que l'indice 0 correspond au nom et que l'indice 1 correspond au prénom.

Une autre fonction permettant le fetch est `mysqli_fetch_array()`. Elle prend en paramètre supplémentaire une constante permettant de récupérer soit un tableau associatif, soit un tableau indexé, soit les deux à la fois.

Les constantes sont :

- `MYSQLI_ASSOC` : retourne un tableau associatif équivalent à `mysqli_fetch_assoc()`.
- `MYSQLI_NUM` : retourne un tableau indexé équivalent à `mysqli_fetch_row()`.
- `MYSQLI_BOTH` : retourne un tableau à la fois associatif et indexé.

Vous devez avoir lu le chapitre L'objet pour bien comprendre la fin de cette section.

Enfin, la dernière fonction permettant le fetch est `mysqli_fetch_object()`.

Cette fonction prend en paramètre le résultat renvoyé par la fonction `mysqli_query()` et retourne un objet avec un attribut par champ retourné par la requête. Cet attribut a pour nom et pour valeur respectivement celui et celle du champ.

Par exemple, pour afficher le nom et le prénom de la table `Personne` :

```
<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).
'<br />';

    // Exécution de la requête
    $resultat = mysqli_query($base, 'SELECT Nom,Prenom FROM Personne');
    if ($resultat == FALSE) {
        echo "Echec de l'exécution de la requête.<br />";
    }
    else {
        //fetch sur chaque ligne ramenée par la requête
        while ($objet = mysqli_fetch_object($resultat)) {
            // Affichage du nom et prénom des personnes
            echo "Nom : ".$objet->Nom." et prénom : ".$objet->Prenom."<br />";
        }
    }

    if (mysqli_close($base)) {
        echo 'Déconnexion réussie.<br />';
    }
    else {
        echo 'Echec de la déconnexion.';
    }
}
else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());}
?>
```

Affiche :

Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Nom : Dupont et prénom : Jean
Nom : Duval et prénom : Monique
Nom : Martin et prénom : Robert
Nom : Durand et prénom : Bernadette
Nom : Martineau et prénom : Alain
Nom : Roland et prénom : Fabien
Déconnexion réussie.

b. Écrire des données

Pour écrire des données, il faut exécuter une requête de type INSERT.

Pour cela, il faut utiliser la fonction `mysqli_query()`.

Cette fonction prend en paramètres :

- L'objet de connexion : objet renvoyé par la fonction `mysqli_connect()`.
- La requête : chaîne de caractères contenant la requête SQL.

Par exemple, pour insérer une personne nommée Nadia DURAND âgée de 31 ans :

```
<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).'<br />';
    $sql = "INSERT INTO Personne (Nom, Prenom, Age) VALUES
('DURAND', 'Nadia', 31)";
    // Exécution de la requête
    $resultat = mysqli_query($base, $sql);
    if ($resultat == FALSE) {
        echo "Echec de l'exécution de la requête.<br />";
    }
    else {
        echo "Personne enregistrée.<br />";
    }

    if (mysqli_close($base)) {
        echo 'Déconnexion réussie.<br />';
    }
    else {
```

```
        echo 'Echec de la déconnexion.';
    }

}

else {
    printf('Erreur %d :
%s.<br/>',mysql_connect_errno(),mysql_connect_error());
}

?>
```

Vous avez alors une nouvelle ligne dans la base de données :

Vous remarquez que l'Id_personne n'a pas été ajouté dans la requête. En effet, il est auto-incrémental donc c'est la base de données qui va automatiquement affecter un nouveau nombre à l'Id_personne.

Donc lorsque vous insérez une nouvelle personne, vous ne connaissez pas son identifiant. Pour récupérer le dernier Id auto-incrémental ajouté dans la base de données, il faut utiliser la fonction `mysql_insert_id()`. Cette fonction prend en paramètre l'objet de connexion et retourne le dernier identifiant auto-incrémental ajouté en base de données.

Par exemple, pour insérer une personne nommée Albert MENFIN âgée de 61 ans, ajoutez après `echo "Personne enregistrée.
"`; les lignes suivantes :

```
$id = mysql_insert_id($base);
echo "Son identifiant est : ".$id."<br />"
```

Affiche :

```
Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Personne enregistrée.
Son identifiant est : 8.
Déconnexion réussie.
```

c. Supprimer des données

Pour supprimer des données, il faut exécuter une requête de type DELETE.

Pour cela, il faut utiliser la fonction `mysql_query()`.

Cette fonction prend en paramètres :

- L'objet de connexion : objet renvoyé par la fonction `mysql_connect()`.
- La requête : chaîne de caractères contenant la requête SQL.

Par exemple, pour supprimer une personne dont le nom est MENFIN :

```
<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).
'<br />';

    $sql = "DELETE FROM Personne WHERE Nom = 'MENFIN'";
    // Exécution de la requête
    $resultat = mysqli_query($base, $sql);
    if ($resultat == FALSE) {
        echo "Echec de l'exécution de la requête.<br />";
    }
    else {
        echo "Personne supprimée.<br />";
    }

    if (mysqli_close($base)) {
        echo 'Déconnexion réussie.<br />';
    }
    else {
        echo 'Echec de la déconnexion.';
    }
}
else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());
}
?>
```

Affiche :

Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Personne supprimée.
Déconnexion réussie.

Même si la personne n'existe pas, vous avez le message « Personne supprimée » car ce message s'affiche lorsqu'il n'y a pas d'erreur dans la requête. Si vous voulez connaître le nombre de lignes supprimées, il faut utiliser la fonction `mysqli_affected_rows()`.

d. Mettre à jour des données

Pour modifier des données, il faut exécuter une requête de type UPDATE.

Pour cela, il faut utiliser la fonction `mysqli_query()`.

Cette fonction prend en paramètre :

- L'objet de connexion : objet renvoyé par la fonction `mysqli_connect()`.
- La requête : chaîne de caractères contenant la requête SQL.

Vous pouvez aussi utiliser la fonction `mysqli_affected_rows()` qui prend en paramètre le résultat renvoyé par la fonction `mysqli_connect()` et retourne le nombre de lignes modifiées par la requête.

Par exemple, pour modifier la personne dont le nom est Durand en changeant son nom par DUCHEMIN et son âge de 23 à 48 :

```
<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : ' . mysqli_get_host_info($base).
'<br />';

    $sql = "UPDATE Personne SET Nom = 'DUCHEMIN', Age = 48 WHERE Nom =
'Durand'";

    // Exécution de la requête
$resultat = mysqli_query($base, $sql);
if ($resultat == FALSE) {
    echo "Echec de l'exécution de la requête.<br />";
}
else {
    $nombre_personne = mysqli_affected_rows($base);
    echo "Nombre de personnes modifiées : ".$nombre_personne."<br />";
}

if (mysqli_close($base)) {
    echo 'Déconnexion réussie.<br />';
}
```

```

    else {
        echo 'Echec de la déconnexion.';
    }

}

else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());
}

?>

```

Affiche :

Connexion réussie.
 Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
 Nombre de personnes modifiées : 1.
 Déconnexion réussie.

5. Requêtes préparées

a. Introduction

Une requête préparée est très utilisée car elle a plusieurs avantages. Premièrement, elle évite l'injection SQL car les données de la requête ne peuvent pas être exécutées. Deuxièmement, dans le cas où vous voulez exécuter plusieurs fois de suite une requête de type INSERT avec des valeurs différentes, il n'est pas nécessaire de reconstruire la requête à chaque fois. Il suffit simplement de lier les nouvelles valeurs et d'exécuter à nouveau la requête. Une requête préparée est donc plus sécurisée et parfois plus rapide qu'une requête non préparée.

Il faut en prérequis avoir ces données :

+ Options							Id_personne	Nom	Prenom	Age
  										
<input type="checkbox"/>		Modifier		Copier		Effacer	1	Dupont	Jean	55
<input type="checkbox"/>		Modifier		Copier		Effacer	2	Duval	Monique	60
<input type="checkbox"/>		Modifier		Copier		Effacer	3	Martin	Robert	35
<input type="checkbox"/>		Modifier		Copier		Effacer	4	DUCHEMIN	Bernadette	48
<input type="checkbox"/>		Modifier		Copier		Effacer	5	Martineau	Alain	20
<input type="checkbox"/>		Modifier		Copier		Effacer	6	Roland	Fabien	26
<input type="checkbox"/>		Modifier		Copier		Effacer	7	DURAND	Nadia	31

b. Lire des données

Avant d'exécuter la requête, il faut préparer la requête, c'est-à-dire qu'il faut écrire la requête en remplaçant les valeurs par des points d'interrogation qui sont remplacés ensuite par les valeurs.

La fonction permettant de préparer la requête est `mysqli_prepare()`.

Cette fonction prend en paramètres :

- L'objet de connexion : objet renvoyé par la fonction `mysqli_connect()`.
- La requête : chaîne de caractères contenant la requête SQL avec les points d'interrogation.

Cette fonction retourne un objet `mysqli_stmt` utilisé par les fonctions `mysqli_stmt_bind_param()` et `mysqli_stmt_bind_result()`.

La fonction `mysqli_stmt_bind_param()` permet de lier les variables à une requête SQL.

Cette fonction a trois paramètres :

- L'objet requête : résultat renvoyé par la fonction `mysqli_prepare()`.
- Les types : chaîne de caractères contenant les types des données passées en paramètres dans le même ordre. Les types sont :
 - `s` pour les chaînes de caractères,
 - `i` pour les entiers,
 - `d` pour les nombres décimaux,
 - `b` pour les blob.
- Les variables : liste des variables liées aux paramètres de la requête sql, séparées par des virgules.

Cette fonction retourne `false` en cas d'erreur et `true` sinon.

Pour exécuter la requête, il faut utiliser la fonction `mysqli_stmt_execute()`. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()` et retourne `false` en cas d'erreur et `true` sinon.

Dans le cas d'une requête de type `SELECT`, il faut pouvoir stocker les valeurs retournées par la requête SQL. Pour cela, il faut utiliser la fonction `mysqli_stmt_bind_result()` qui va associer les colonnes d'un résultat à des variables.

Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()` et les variables correspondant à chaque colonne. Elle retourne `true` en cas de succès et `false` en cas d'échec.

Puis, il faut exécuter la requête préparée avec la fonction `mysqli_stmt_execute()`. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()` et retourne `false` en cas d'erreur et `true` sinon.

Pour lire les valeurs dans une boucle, il faut utiliser la fonction `mysqli_stmt_fetch()`. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()` et retourne `false` en cas d'erreur et `true` sinon.

Enfin, il faut fermer une requête préparée avec la fonction `mysqli_stmt_close()`. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()`. Elle retourne `true` en cas de succès et `false` en cas d'échec.

Par exemple, pour afficher le nom et le prénom des personnes âgées de plus de 35 ans :

```
<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).'
<br />';

    $sql = "SELECT Nom, Prenom FROM Personne WHERE Age > ?";
    // Préparation de la requête
    $resultat = mysqli_prepare($base, $sql);
    // Liaison des paramètres.
    $ok = mysqli_stmt_bind_param($resultat, 'i', $age);
    $age=35; // Tant que vous n'avez pas exécuté la requête,
            // vous pouvez initialiser la variable
            // Exécution de la requête.
    $ok = mysqli_stmt_execute($resultat);

    if ($ok == FALSE) {
        echo "Echec de l'exécution de la requête.<br />";
    }
    else {
        // Association des variables de résultat.
        $ok = mysqli_stmt_bind_result($resultat, $Nom, $Prenom);
        // Lecture des valeurs.
        echo "Nom et prénom des personnes ayant un age > 35<br />";
        while (mysqli_stmt_fetch($resultat)) {
            echo $Nom.", ".$Prenom."<br />";
        }
        mysqli_stmt_close($resultat);
    }

    if (mysqli_close($base)) {
        echo 'Déconnexion réussie.<br />';
    }
}
```

```

        else {
            echo 'Echec de la déconnexion.';
        }

    }

else {
    printf('Erreur %d : %s.<br/>',mysql_connect_errno(),
mysql_connect_error());
}

?>

```

Affiche :

```

Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Nom et prénom des personnes ayant un age >35
    Dupont, Jean
    Duval, Monique
    DUCHEMIN, Bernadette
Déconnexion réussie.

```

c. Écrire des données

Pour écrire des données, il faut exécuter une requête de type INSERT.

Pour cela il suffit de préparer la requête avec `mysqli_prepare()`, de lier les variables à la requête avec `mysqli_stmt_bind_param()` et de l'exécuter avec `mysqli_stmt_execute()`. Ces trois fonctions sont expliquées dans la section précédente.

Par exemple, pour insérer une personne nommée Monique ROVIER âgée de 63 ans :

```

<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).'
<br />';

    $sql = "INSERT INTO Personne (Nom, Prenom, Age) VALUES (?, ?, ?)";
    // Préparation de la requête
    $resultat = mysqli_prepare($base, $sql);
    // Liaison des paramètres.

```



```

    $ok = mysqli_stmt_bind_param($resultat, 'ssi',$nom,$prenom,$age);

    $nom = 'ROVIER';
    $prenom = 'Monique';
    $age = 63;

    // Exécution de la requête.
    $ok = mysqli_stmt_execute($resultat);

    if ($ok == FALSE) {
        echo "Echec de l'exécution de la requête.<br />";
    }
    else {
        echo "Personne ajoutée.<br />";
    }
    mysqli_stmt_close($resultat);
    if (mysqli_close($base)) {
        echo 'Déconnexion réussie.<br />';
    }
    else {
        echo 'Echec de la déconnexion.';
    }
}
else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());
}
?>

```

Affiche :

```

Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Personne ajoutée.
Déconnexion réussie.

```

d. Modifier des données

Pour avoir le nombre de lignes modifiées par une requête de type UPDATE, il faut utiliser la fonction `mysqli_stmt_affected_rows()`.

Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()` et retourne -1 en cas d'erreur et le nombre de lignes mises à jour sinon.

Par exemple, pour modifier la personne dont le nom est DURAND en changeant son nom par DUPUIS et son âge de 31 à 33 :

```
<?php
// Connexion à la base de données
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).
'<br />';

    $sql = "UPDATE Personne SET Nom = ?, Age = ? WHERE Nom = 'DURAND'";
    // Préparation de la requête
    $resultat = mysqli_prepare($base, $sql);
    // Liaison des paramètres.
    $ok = mysqli_stmt_bind_param($resultat, 'si', $nom, $age);
    $nom = 'DUPUIS';
    $age = 33; //Tant que vous n'avez pas exécuté la requête,
                //vous pouvez initialiser les variables
    // Exécution de la requête.
    $ok = mysqli_stmt_execute($resultat);

    if ($ok == FALSE) {
        echo "Echec de l'exécution de la requête.<br />";
    }
    else {
        echo "Nombre de personnes modifiées :
".mysqli_stmt_affected_rows($resultat).".<br />";
    }
    mysqli_stmt_close($resultat);
    if (mysqli_close($base)) {
        echo 'Déconnexion réussie.<br />';
    }
    else {
        echo 'Echec de la déconnexion.';
    }
}
else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
```

```
mysqli_connect_error());  
}  
?>
```

Affiche :

Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Nombre de personnes modifiées : 1.
Déconnexion réussie.

e. Supprimer des données

De la même façon que pour écrire ou modifier des données, vous allez utiliser les fonctions `mysqli_prepare()`, `mysqli_stmt_bind_param()` et `mysqli_stmt_execute()`.

Par exemple, pour supprimer Martineau Alain :

```
<?php  
// Connexion à la base de données  
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");  
if ($base) {  
    echo 'Connexion réussie.<br />';  
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).  
'<br />';  
  
    $sql = "DELETE FROM Personne WHERE Nom = ?";  
    // Préparation de la requête  
    $resultat = mysqli_prepare($base, $sql);  
    // Liaison des paramètres.  
    $ok = mysqli_stmt_bind_param($resultat, 's', $nom);  
    $nom = 'Martineau'; //Tant que vous n'avez pas exécuté la requête,  
                        //vous pouvez initialiser la variable  
    // Exécution de la requête.  
    $ok = mysqli_stmt_execute($resultat);  
  
    if ($ok == FALSE) {  
        echo "Echec de l'exécution de la requête.<br />";  
    }  
    else {  
        echo "Personne supprimée.<br />";  
    }  
}
```

```

mysqli_stmt_close($resultat);

if (mysqli_close($base)) {
    echo 'Déconnexion réussie.<br />';
}
else {
    echo 'Echec de la déconnexion.';
}

}
else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());
}
?>

```

Affiche :

```

          Connexion réussie.
    Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
          Personne supprimée.
          Déconnexion réussie.

```

f. Stocker un résultat

Si vous voulez connaître le nombre de lignes sélectionnées dans une requête de type SELECT, il existe une fonction permettant de stocker le résultat et une autre pour avoir le nombre de lignes de ce résultat.

La fonction `mysqli_stmt_store_result()` permet de stocker en mémoire le résultat d'une requête. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()` et retourne `false` en cas d'erreur et `true` sinon.

La fonction `mysqli_stmt_num_rows()` permet de retourner le nombre de lignes d'une requête préparée. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()` et retourne 0 en cas d'erreur et le nombre de lignes de la requête sinon.

La fonction `mysqli_stmt_free_result()` permet de libérer le résultat obtenu par `mysqli_stmt_store_result()` de la mémoire. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysqli_prepare()` et ne retourne rien.

Par exemple, pour afficher le nombre de personnes qui ont un âge supérieur à 35 :

```

<?php
// Connexion à la base de données

```

```
$base = mysqli_connect("127.0.0.1", "root", "", "bdd");
```

```
if ($base) {  
    echo 'Connexion réussie.<br />';  
    echo 'Informations sur le serveur : '.mysqli_get_host_info($base).  
'<br />';  
  
    $sql = "SELECT Nom, Prenom FROM Personne WHERE Age > ?";  
    // Préparation de la requête  
    $resultat = mysqli_prepare($base, $sql);  
    // Liaison des paramètres.  
    $ok = mysqli_stmt_bind_param($resultat, 'i', $age);  
    $age = 35; //tant que vous n'avez pas exécuté la requête,  
              //vous pouvez initialiser la variable  
    // Exécution de la requête.  
    $ok = mysqli_stmt_execute($resultat);  
  
    if ($ok == FALSE) {  
        echo "Echec de l'exécution de la requête.<br />";  
    }  
    else {  
        // Association des variables de résultat.  
        $ok = mysqli_stmt_bind_result($resultat, $Nom, $Prenom);  
        // Stockage des valeurs.  
        $ok = mysqli_stmt_store_result($resultat);  
        echo "Nombre de personnes ayant un age > 35 :  
".mysqli_stmt_num_rows($resultat)."<br />";  
        //Libération du résultat  
        mysqli_stmt_free_result($resultat);  
        mysqli_stmt_close($resultat);  
    }  
  
    if (mysqli_close($base)) {  
        echo 'Déconnexion réussie.<br />';  
    }  
    else {  
        echo 'Echec de la déconnexion.';  
    }  
}
```

```

}
else {
    printf('Erreur %d : %s.<br/>',mysql_connect_errno(),
mysql_connect_error());
}
?>

```

Affiche :

```

Connexion réussie.
Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
Nombre de personnes ayant un age > 35 : 4
Déconnexion réussie.

```

Cette méthode est consommatrice de mémoire, donc il faut faire attention à ce que vous stockez dedans.

g. Afficher les erreurs d'une requête préparée

Les fonctions permettant de récupérer les erreurs d'une requête préparée sont `mysql_stmt_errno()` et `mysql_stmt_error()`.

La fonction `mysql_stmt_errno()` permet de retourner le numéro d'erreur. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysql_prepare()` et retourne 0 s'il n'y a pas d'erreur.

La fonction `mysql_stmt_error()` permet de retourner le message d'erreur. Cette fonction prend en paramètre l'objet requête renvoyé par la fonction `mysql_prepare()`.

Ces fonctions affichent les erreurs une fois la requête exécutée. Pour voir les erreurs de préparation, il faut utiliser les fonctions `mysql_errno()` et `mysql_error()`.

Par exemple, si vous ajoutez une personne avec un identifiant déjà existant :

```

<?php
// Connexion à la base de données
$base = mysql_connect("127.0.0.1", "root", "", "bdd");
if ($base) {
    echo 'Connexion réussie.<br />';
    echo 'Informations sur le serveur : '.mysql_get_host_info($base).
'<br />';

    $sql = "INSERT INTO Personne (Id_personne, Prenom, Age)
VALUES (?, ?, ?)";
    // Préparation de la requête

```

```

$resultat = mysqli_prepare($base, $sql);

// Liaison des paramètres.

$ok = mysqli_stmt_bind_param($resultat, 'isi',$id,$prenom,$age);

$id = 1; //Ajout d'un identifiant déjà existant

$prenom = 'Monique';

$age = 63; //Tant que vous n'avez pas exécuté la requête,
           //vous pouvez initialiser les variables

// Exécution de la requête.

$ok = mysqli_stmt_execute($resultat);


if ($ok == FALSE) {
    echo "Echec de l'exécution de la requête.<br />";
    echo "erreur : ".mysqli_stmt_errno($resultat).' -
'.mysqli_stmt_error($resultat)."<br />";
}
else {
    echo "Personne ajoutée.<br />";
}

mysqli_stmt_close($resultat);

if (mysqli_close($base)) {
    echo 'Déconnexion réussie.<br />';
}
else {
    echo 'Echec de la déconnexion.';
}

}

else {
    printf('Erreur %d : %s.<br/>',mysqli_connect_errno(),
mysqli_connect_error());
}

?>

```

Affiche :

Connexion réussie.
 Informations sur le serveur : MySQL host info: 127.0.0.1 via TCP/IP
 Echec de l'exécution de la requête.
 erreur : 1062 - Duplicate entry '1' for key 'PRIMARY'
 Déconnexion réussie.