



Ecole Nationale
Supérieure
d'Ingénieurs
de Tunis

Année Universitaire : 2019/2020

Session : 1 ING Info

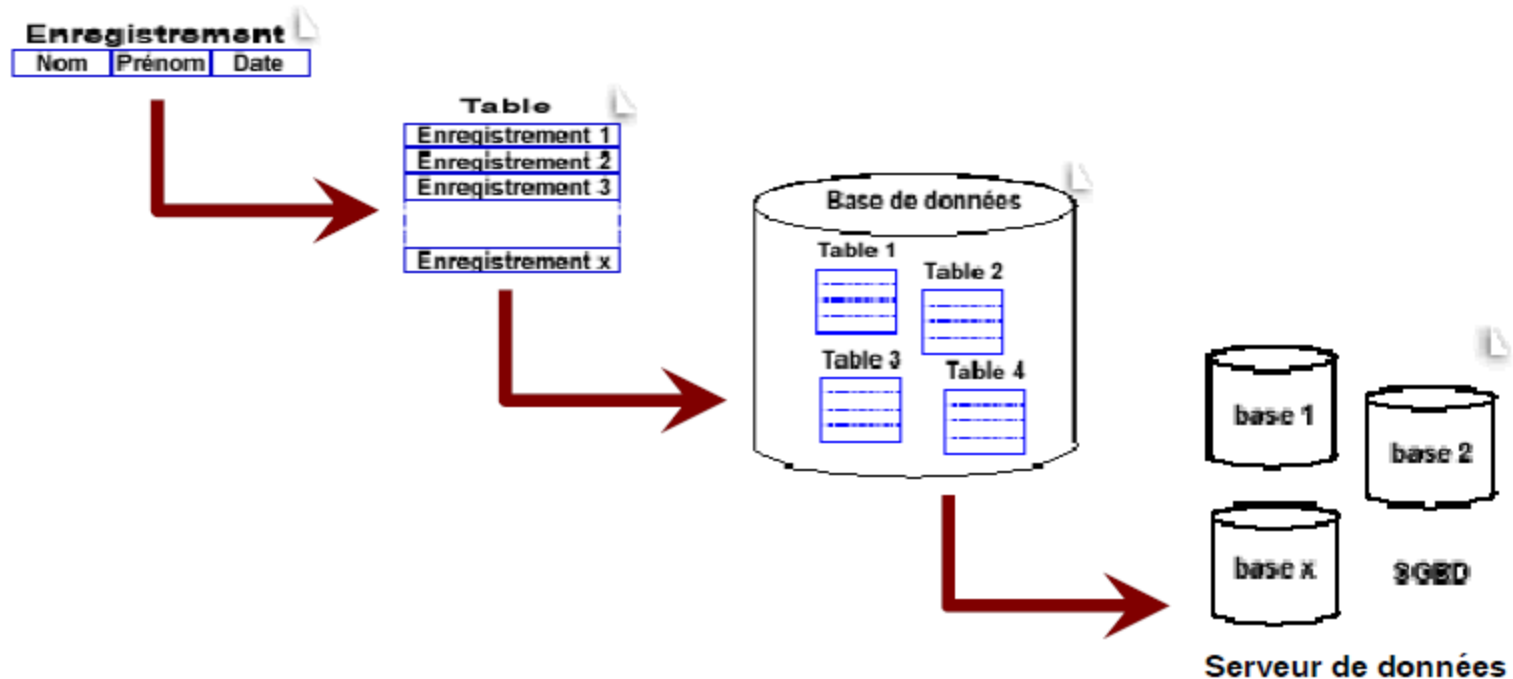
Matière : Cours WEB2

Utilisation de bases de données avec PHP PhpMyAdmin

Présentation

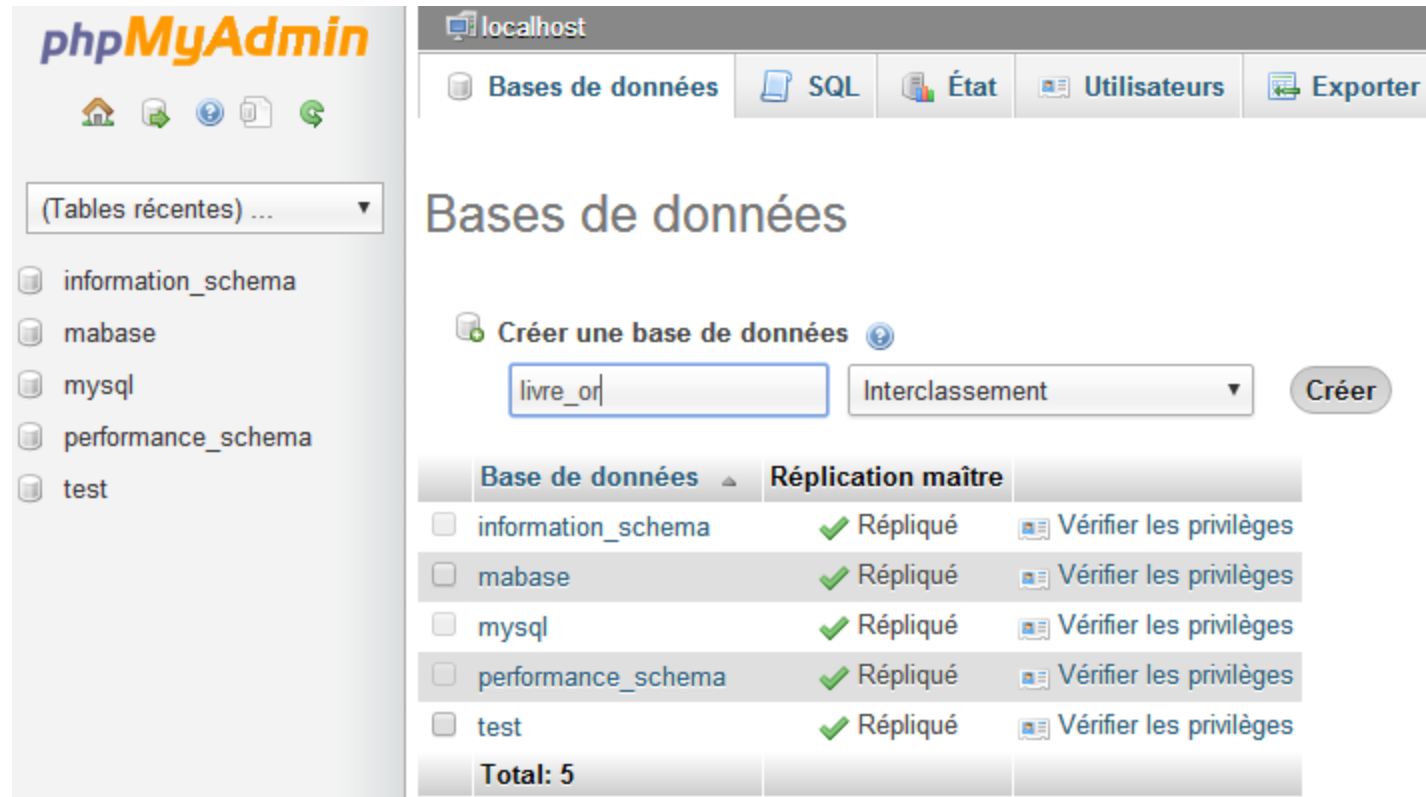
PHP propose de nombreux outils permettant de travailler avec la plupart des SGBDR (Oracle, Sybase, Microsoft SQL Server, PostgreSQL ou encore MySQL).

Un SGBD est un ensemble d'applications permettant de manipuler les données (ajout, suppression, modification et lecture), mais aussi de contrôler l'accès. Les données sont structurées de la manière suivante :



Création de base de données: PhpMyAdmin

PhpMyAdmin est véritablement le panneau d'administration standard d'une base de données. C'est en réalité un ensemble de pages PHP (et en rien un programme) qui simplifient la tâche du programmeur en offrant une interface simple et efficace pour gérer les différentes bases de données du site.



The screenshot displays the PhpMyAdmin web interface. On the left, a sidebar contains the 'phpMyAdmin' logo and a list of databases: information_schema, mabase, mysql, performance_schema, and test. The main content area is titled 'Bases de données' and includes a navigation bar with links for 'Bases de données', 'SQL', 'État', 'Utilisateurs', and 'Exporter'. Below the navigation bar, there is a form to 'Créer une base de données' (Create a database) with a text input field containing 'livre_or' and a dropdown menu set to 'Interclassement'. A 'Créer' button is to the right of the form. Below the form is a table showing the status of existing databases.

| Base de données | Réplication maître | |
|---|--------------------|---|
| <input type="checkbox"/> information_schema | ✓ Répliqué | Vérifier les privilèges |
| <input type="checkbox"/> mabase | ✓ Répliqué | Vérifier les privilèges |
| <input type="checkbox"/> mysql | ✓ Répliqué | Vérifier les privilèges |
| <input type="checkbox"/> performance_schema | ✓ Répliqué | Vérifier les privilèges |
| <input type="checkbox"/> test | ✓ Répliqué | Vérifier les privilèges |
| Total: 5 | | |

Création de base de données: PhpMyAdmin

☀️ Créer une nouvelle table sur la base **livre_or**

Nom: Nombre de colonnes:

Nom de la table: Ajouter colonne(s)

| Structure ? | | | | | | | |
|--------------------------------------|--|----------------------|---|----------------------|---|--------------------------|--------------------------------------|
| Nom | Type ? | Taille/Valeurs* ? | Défaut ? | Interclassement | Attributs | Null | Index |
| <input type="text" value="id"/> | <input type="text" value="INT"/> | <input type="text"/> | <input type="text" value="Aucune"/> | <input type="text"/> | <input type="text" value="UNSIGNED"/> | <input type="checkbox"/> | <input type="text" value="PRIMARY"/> |
| <input type="text" value="pseudo"/> | <input type="text" value="TINYTEXT"/> | <input type="text"/> | <input type="text" value="Aucune"/> | <input type="text"/> | <input type="text"/> | <input type="checkbox"/> | <input type="text" value="---"/> |
| <input type="text" value="message"/> | <input type="text" value="TEXT"/> | <input type="text"/> | <input type="text" value="Aucune"/> | <input type="text"/> | <input type="text"/> | <input type="checkbox"/> | <input type="text" value="---"/> |
| <input type="text" value="date"/> | <input type="text" value="TIMESTAMP"/> | <input type="text"/> | <input type="text" value="CURRENT_TIME"/> | <input type="text"/> | <input type="text" value="on update CI"/> | <input type="checkbox"/> | <input type="text" value="---"/> |

Commentaires sur la table:

Moteur de stockage: ?

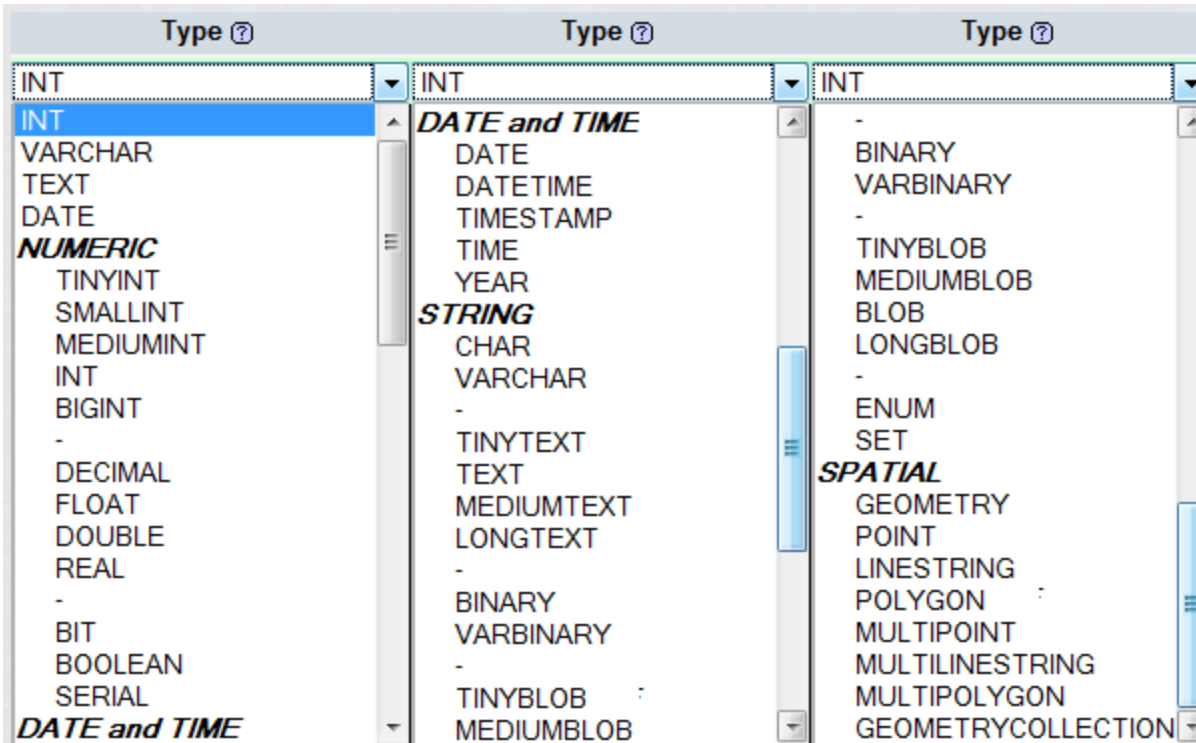
Interclassement:

Définition de PARTITION: ?

Les types de données: PhpMyAdmin

Il y a 4 grand types de donnée dans MySQL :

- Les types Numériques : les nombres en somme
- Les Dates et Heures
- Les Textes et chaines de caractères
- Les éléments spatiaux comme les formes de géométrie (utilisés pour la cartographie par exemple)



Les types de données: PhpMyAdmin

Les 4 premières valeurs de la liste ne sont que des raccourcis pour les 4 entrées les plus courantes :

« INT » est nom nombre entier

« VARCHAR » un texte cours de 0 à 65.535 caractères

« TEXT » pour du texte en grande quantité (livres, articles, news...)

« DATE » une date sous la forme (jour, mois, année)

Les Types numériques

Les types numériques se divisent en deux grands groupes : les nombres entiers et les doubles

| Les entiers | | | |
|------------------------------|---------------------------|-----------------|-----------------|
| Type, avec attribut de signe | Description | Limite minimale | Limite maximale |
| TINYINT | Très petit entier | -128 | 127 |
| TINYINT UNSIGNED | Très petit entier positif | 0 | 255 |
| SMALLINT | Petit entier | -32 768 | 32 767 |
| SMALLINT UNSIGNED | Petit entier positif | 0 | 65 535 |
| MEDIUMINT | Entier «moyen» | -8388608 | 8388607 |
| MEDIUMINT UNSIGNED | Entier «moyen» positif | 0 | 16777215 |
| INT (INTEGER) | Entier «normal» | -2147483648 | 2147483647 |
| INT UNSIGNED | Entier «normal» positif | 0 | 4294967295 |
| BIGINT | «Grand» entier | -9,22E+018 | 9,22E+018 |
| BIGINT UNSIGNED | «Grand» entier positif | 0 | 1,84E+019 |

| Les doubles | | | | |
|-------------|--|---|---|--|
| Type | Description | Limites négatives | 0 | Limites positives |
| FLOAT | Nombre en virgule flottante (simple précision) | - 3,402823466*10 ⁺³⁸ - 1,175494351*10 ⁻³⁸ | 0 | 1,175494351*10 ⁻³⁸ 3,402823466*10 ⁺³⁸ |
| DOUBLE REAL | Nombre en virgule flottante (double précision) | -1,7976931348623157*10 ⁺³⁰⁸ - 2,2250738585072014*10 ⁻³⁰⁸ | 0 | 2,2250738585072014*10 ⁻³⁰⁸ 1,7976931348623157*10 ⁺³⁰⁸ |

String: Text

| Type | Limites | Description |
|-----------------------------|---------------|--|
| CHAR (CHARACTER) | 1 à 255 | Stockage d'une chaîne de caractères de longueur fixe, dont la longueur maximale est fixée par un nombre entre parenthèses suivant le nom du type, si la chaîne n'atteint pas la longueur fixée, elle est complétée à droite par des caractères d'espacement. |
| VARCHAR (CHARACTER VARYING) | variable | Equivalent de CHAR où la longueur est variable. Dont la longueur maximale est fixée de la même façon que pour le type CHAR. |
| * TINYTEXT | 255 | Gestion de chaînes de caractères de longueur variable. |
| * TEXT | 65 535 | Gestion de chaînes de caractères de longueur variable. |
| * MEDIUMTEXT | 16777215 | Gestion de chaînes de caractères de longueur variable. |
| * LONGTEXT | 4294967295 | Gestion de chaînes de caractères de longueur variable. |
| * TINYBLOB | 255 | BLOB : Binary Large Object (donnée binaire arbitraire) Aptitude à faire la distinction entre MAJ et min. |
| * BLOB | 65 535 | |
| * MEDIUMBLOB | 16777215 | |
| * LONGBLOB | 4294967295 | |
| ENUM | 65 535 au max | Gestion des énumérations. Une énumération SQL est une chaîne de caractères qui peut prendre sa valeur parmi un certain nombre de valeurs. Une énumération ne peut prendre qu'une valeur à la fois. |
| SET | 64 au max | Gestion des ensembles. Un ensemble SQL peut comprendre plusieurs valeurs. |

Les types de données : PhpMyAdmin

DATE and TIME :

les dates« DATE » stocke une date au format « YYYY-MM-DD »

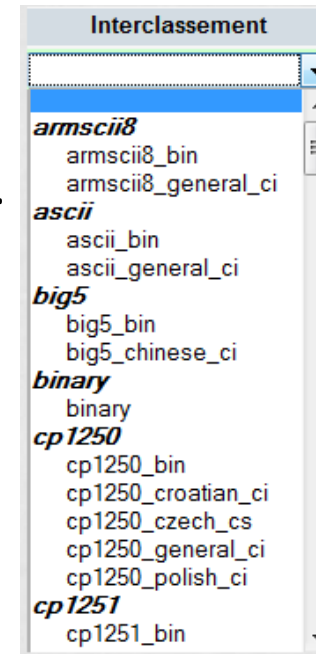
« DATETIME » stocke une date au format « YYYY-MM-DD HH:MM:SS »

« TIMESTAMP » se comporte comme un DATETIME mais cette fois-ci stocke le nombre de secondes écoulées depuis le 1er janvier 1970. La date résultante est aussi sous la forme « YYYY-MM-DD HH:MM:SS ».

« TIME » stocke une heure sous la forme « HH:MM:SS »

« YEAR » stocke

La colonne d'interclassement permet de spécifier le jeu de caractère à utiliser pour stocker l'information.



Les types de données: PhpMyAdmin

Le champ index possède plusieurs valeurs. Vous l'utiliserez au moins une fois par table pour définir le champ primaire.

| Null | Index | A I | Commentaires |
|--------------------------|--|--------------------------|----------------------|
| <input type="checkbox"/> | <div><div></div><div>PRIMARY UNIQUE INDEX FULLTEXT</div></div> | <input type="checkbox"/> | <input type="text"/> |
| <input type="checkbox"/> | | <input type="checkbox"/> | <input type="text"/> |
| <input type="checkbox"/> | | <input type="checkbox"/> | <input type="text"/> |

« PRIMARY » est très important. **Toute table doit avoir un champ primaire.** Ce champ est utilisé pour améliorer les performances et décider de la manière dont vont être stocké les informations. C'est le premier champ que va regarder MySQL lors de vos requêtes sur la base. Par conséquent, chaque valeur pour le champ primaire doit être unique. Cela est assez simple puisque l'on utilise un champ ID qui lui est unique pour chaque entrée.

Si vous utilisez la valeur « UNIQUE », chaque entrée de ce champ devra être distinct et une erreur sera retournée si vous entrez une valeur identique à une entrée déjà existante dans la base.

Les actions de base sur une table: PhpMyAdmin

Les actions de bases que l'on peut effectuer sur une table :

- Insérer une entrée
- Supprimer une entrée
- Modifier une entrée
- Afficher les entrées
- Effectuer une requête sur la table

Puis les différentes actions qui vous seront utiles comme:

- Sauvegarder une base MySQL
- Restaurer une base MySQL
- Vider une table
- Supprimer une table

Insertion

| Colonne | Type | Fonction | Null | Valeur |
|---------|------------------|----------|------|--------------------------|
| id | int(10) unsigned | | | |
| pseudo | tinytext | | | Craym |
| message | text | | | Premier message livre or |
| date | timestamp | | | CURRENT_TIMESTAMP |

Suppression

+ Options

| | | | | | id | pseudo | message | date |
|--------------------------|----------|--------|---------|--|----|--------|--------------------------|---------------------|
| <input type="checkbox"/> | Modifier | Copier | Effacer | | 0 | Craym | Premier message livre or | 2016-04-03 18:44:21 |
| <input type="checkbox"/> | Modifier | Copier | Effacer | | 32 | crayam | premier message | 2016-04-03 18:45:09 |

Exercice d'application:

Créer une base de données nommée « gusers ».

La BD comporte la table contacts:

■ Exemple de table :

```
-- TABLE : contacts
NOM      PRENOM      ADRESSE      VILLE      CP      PAYS
-----
Alpha    Alain      1 rue AAA    Paris      75000    FRANCE
Beta     Benjamin   34 rue G.    Lyon       69000    FRANCE
Gamma    Gaëlle     3, bakerstreet London      1234     U.K.
```

Ajouter cette table à la base créée.



Ecole Nationale
Supérieure
d'Ingénieurs
de Tunis

Année Universitaire : 2019/2020

Session : 1 ING Info

Matière : Cours WEB2

Utilisation de bases de données avec PHP MySQL

Connexion et sélection de la base de données

Une base de données contient une ou plusieurs tables, chaque table ayant un ou plusieurs enregistrements.

La gestion d'une BD est rendue possible grâce à un Système de Gestion de Base de Données (S.G.B.D). Il a pour rôle de : manipuler, compléter, mettre à jour les données dans les tables et de pouvoir les interroger via des requêtes, par le langage SQL (Structured Query Language: Langage de requêtes structurées)

Il existe plusieurs S.G.B.D., certains gratuits, d'autres payants: pour les gratuits, on trouve MySQL.

Connexion et sélection de la base de données

Avant de faire le moindre accès à une base de données, il faut impérativement établir une connexion et choisir sa base de données.

```
<?php
//connexion
$serveurBD = "localhost";    //nom du serveur
$nomUtilisateur = "root";    //nom utilisateur est toujours root si on travaille en local
$motDePasse = "";
$baseDeDonnees = "bcontacts"; //indiquer le nom de la base de données

//se connecter au serveur
@mysql_connect($serveurBD, $nomUtilisateur, $motDePasse) or die("Impossible de
se connecter au serveur de bases de données.");
//selectionner la base de données à utiliser dans le script
@mysql_select_db($baseDeDonnees) or die("Cette base de donnees n'existe pas");
//fermer la connexion à la BD
mysql_close();
?>
```

Connexion et sélection de la base de données

\$serveurBD est le nom (ou adresse IP) du serveur où se trouve la base de données MySQL (C'est une information qui vous est donnée par votre hébergeur. Chez vous, en local, c'est "localhost")

\$nomUtilisateur est l'identifiant sous lequel vous souhaitez vous connecter à la base de données (C'est également une information qui vous est donnée par votre hébergeur). Chez vous par défaut, c'est généralement "root".

\$motDePasse est le mot de passe associé à l'identifiant ci-dessus (Ceci est également communiqué par votre hébergeur). Chez vous c'est une chaîne vide par défaut.

\$baseDeDonnees est un nom de base de données qui doit exister (voir également avec votre hébergeur). Chez vous, vous devrez en créer une en passant par exemple par PHPMyAdmin (A vous de choisir son nom).

mysql_connect() retourne un identifiant de connexion ou FALSE en cas d'échec.

mysql_select_db() renvoie un booléen utile pour savoir si la connexion a été réussie.

Connexion et sélection de la base de données

NB: or *die("")* permet d'afficher un texte et de stopper le script si la fonction précédente renvoie une erreur. Notez que pour ne pas avoir à l'écran les messages d'erreur remontés par PHP, il suffit de mettre un @ devant les appels aux fonctions, par exemple *@mysql_connect()*

La déconnexion

La déconnexion se fait par appel à la fonction **mysql_close()**. Il est indispensable, de faire appel à cette fonction avant la fin de l'exécution du script PHP. Sinon, les ressources ne sont pas libérées et cela peut conduire à ce que la base de données devienne (pour un temps) inaccessible (le nombre de connexions pouvant être simultanément ouvertes étant généralement limité).

Requêtes SQL

Une fois que nous sommes connectés à MySQL et à une base de données, nous pouvons effectuer des tâches classiques dans un SGBD :

- afficher une table
- supprimer un enregistrement
- mettre à jour un enregistrement
- afficher le résultat d'une requête...

Il est donc indispensable d'avoir au préalable une bonne connaissance du langage SQL...

Toutes ces actions sont directement effectuées en SQL.

Requêtes SQL: la clause Select

Lorsqu'on effectue une requête de sélection de tuples à l'aide de la fonction **mysql_query**, il est essentiel de stocker le résultat de la requête (les enregistrements) dans une variable, que l'on nomme généralement \$result.

Toutefois, cette variable contient l'ensemble des enregistrements et n'est donc pas exploitable telle quelle. Ainsi on utilise la fonction **mysql_fetch_row()**, qui découpe les lignes de résultat en colonnes (par exemple Nom,adresse,...) et les affecte à une variable tableau dans l'ordre où elles arrivent.

Requêtes SQL: la clause Select

//lancement d'une requête SELECT pour afficher le contenu de la table contacts

// 1) On crée la requête

```
$req = "SELECT * FROM contacts";
```

// 2) on execute la requête et on sauvegarde le resultat dans la variable result

```
$result = mysql_query($req);
```

// 3) on va scanner tous les tuples un par un

```
echo "<table border='1'>";
```

```
while ($ligne = mysql_fetch_row($result))
```

```
{ // on affiche les résultats
```

```
echo "<tr>";
```

```
echo "<td>".$ligne[0]."</td>"; // extraire la valeur du champ nom de cette ligne
```

```
echo "<td>".$ligne[1]."</td>"; // extraire le prenom de cette ligne
```

```
echo "<td>".$ligne[2]."</td>"; // extraire la ville de cette ligne
```

```
echo "<td>".$ligne[3]."</td>"; // extraire le cp de cette ligne
```

```
echo "<td>".$ligne[4]."</td>"; // extraire le pays de cette ligne
```

```
echo "</tr>";
```

```
}
```

```
echo "</table>";
```

Requêtes SQL: la clause Select

La fonction **mysql_fetch_row()** analyse donc chaque ligne de résultat de la requête et stocke les colonnes dans le tableau `$ligne[]`. Ainsi, le champ nom sera stocké dans `$ligne[0]`, le champ prenom dans `$ligne[1]`, le champ ville dans `$ligne[2]`,.....

On inclut généralement **mysql_fetch_row()** dans une boucle while de telle façon à ce que l'ensemble des lignes de résultat soient traitées. Lorsqu'il n'y a plus de ligne à traiter, la boucle while se termine et l'interpréteur exécute la suite des instructions.

Requêtes SQL: la clause Select

La clause WHERE

La clause WHERE permet de spécifier quelles sont les lignes à sélectionner dans une table. Elle est suivie d'un prédicat (expression logique ayant la valeur vrai ou faux) qui sera évalué pour chaque ligne. Les lignes pour lesquelles le prédicat est vrai seront sélectionnées.

Un prédicat simple est la comparaison de deux expressions ou plus au moyen d'un opérateur logique :

WHERE exp1 = exp2

WHERE exp1 != exp2

WHERE exp1 < exp2

WHERE exp1 > exp2

WHERE exp1 <= exp2

WHERE exp1 >= exp2

WHERE exp1 BETWEEN exp2 AND exp3

Requêtes SQL: la clause Select

Exemple :

//requete 2

```
$req2 = "SELECT nom, prenom, ville FROM contacts where ville='Paris'";  
$result = mysql_query($req2);  
echo "<table border='2'>";  
while ($ligne = mysql_fetch_row($result))  
{ // on affiche les résultats  
echo "<tr>";  
echo "<td>".$ligne[0]."</td>"; //premier champ retourné nom  
echo "<td>".$ligne[1]."</td>"; //deuxieme champ retourné prenom  
echo "<td>".$ligne[2]."</td>"; //troisième champ retourné ville  
echo "</tr>";  
}  
echo "</table>";
```

Requêtes SQL: INSERT, DELETE, UPDATE

Pour ajouter un enregistrement dans un tableau: INSERT

// On crée la requête

```
$req = "INSERT INTO nomtable (champ1, champ2, champ3..., champ n)  
VALUES ('valeur1', 'valeur2', 'valeur3', ...'valeurn')";
```

// on envoie la requête

```
$res = mysql_query($req);
```

Exemple

//définir une requête d'insertion

```
$reqAjout = "insert INTO contacts (nom, prenom, adresse, ville, CP, pays)  
VALUES ('Dupont', 'Marie', '9 rue parisienne', 'normandie', '2019', 'France') ";
```

//exécuter la requete

```
$res = mysql_query($reqAjout);
```

//tester si la requête est exécutée ou non

```
if($res!=false) echo "Ajout du contact avec succès!";  
else echo "Problème d'ajout du contact";
```


Requêtes SQL: INSERT, DELETE, UPDATE

Pour supprimer un enregistrement dans un tableau: DELETE

// On crée la requête

```
$req = "DELETE FROM nomtable WHERE champ='valeur' ";
```

// on envoie la requête en utilisant mysql_query()

```
$res = mysql_query($req);
```

Exemple

//définir une requête de suppression

```
$reqSupp = "delete from contacts where ville='Paris' ";
```

//executer la requete

```
$res = mysql_query($reqSupp);
```

//tester si la requête est exécutée ou non

```
if($res!=false) echo "Suppression du contact avec succès!";
```

```
else echo "Problème de suppression du contact";
```

Requêtes SQL: INSERT, DELETE, UPDATE

Pour modifier les valeurs d'un enregistrement dans un tableau: UPDATE

// On crée la requête

```
$req = "UPDATE nomtable SET champ1= 'valeur', champ2='exp2', ...  
WHERE predicat ";
```

// on envoie la requête à l'aide de mysql_query

```
$res = mysql_query($req);
```

Exemple

//lancer une requête de modification

```
$reqModif = "update contacts set nom='ali' where ville='London' ";
```

//executer la requete

```
$res = mysql_query($reqModif);
```

//tester si la requête est exécutée ou non

```
if($res!=false) echo "Modification du contact avec succès!";
```

```
else echo "Problème de modification du contact";
```



Ecole Nationale
Supérieure
d'Ingénieurs
de Tunis

Année Universitaire : 2019/2020

Session : 1 ING Info

Matière : Cours WEB2

Utilisation de bases de données avec PHP MySQL