



Atelier Web – Les langages du Web

Le MySQL

Module 1 - Les bases

Définition

MySQL = SGBD = Système de Gestion de Base de Données

Le rôle du MySQL est de vous aider à enregistrer les données de manière organisée afin de vous aider à les retrouver plus facilement.

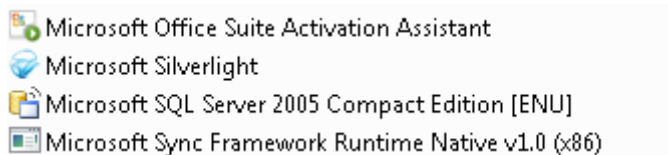
C'est notamment grâce au MySQL que vous pouvez enregistrer la liste des membres de votre site, les messages postés sur le forum.

Le langage qui permet de communiquer avec la base de données se nomme le SQL.

Les bases de données (BDD) les plus connues !

- MySQL : libre et gratuit, le plus connu.
- PostgreSQL : libre et gratuit, moins connu.
- SQLite : Libre et gratuit, léger mais limité.
- Oracle : Très complet, mais réservé aux grosses entreprises, pas libre et très cher.
- Microsoft SQL Server : Le SGDB de Microsoft

Le SQL de Microsoft ? ... Mais je l'ai déjà vu quelque part Regardez dans la liste des programmes du Panneau de Configuration Et vous trouverez !



Question : A quoi peut-donc servir MySQL ?

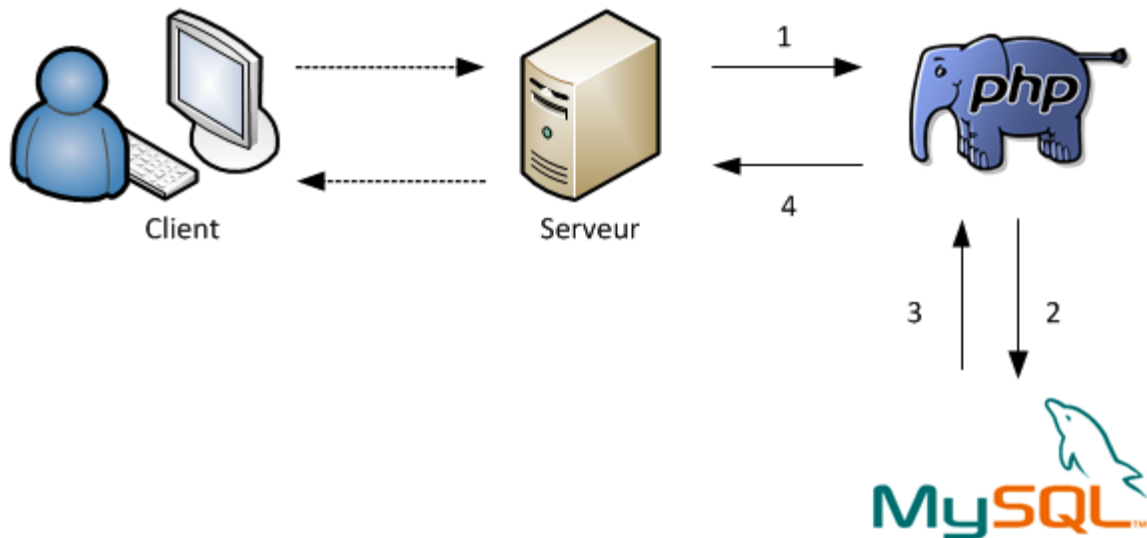
Réponse : A donner des instructions du genre : « *Récupère moi les 10 dernières news de mon site* », ou « *Supprime le dernier message posté dans ce forum* ».

1- Comment ça fonctionne ?

Bon, fallait pas rêver... Il faut bien que ça se complique un peu... !

Il n'est pas possible de dialoguer directement avec MySQL... il faut passer par le PHP
Autrement dit : On va demander à PHP : « Va dire à MySQL de faire ceci »

Voyons cela par un petit schéma :



Voici ce qu'il peut se passer lorsque le serveur a reçu une demande d'un client qui veut poster un message sur vos forums :

1. Le serveur utilise toujours PHP, il lui fait donc passer le message.
2. PHP effectue les actions demandées et se rend compte qu'il a besoin de MySQL. En effet, le code PHP contient à un endroit "Va demander à MySQL d'enregistrer ce message". Il fait donc passer le travail à MySQL.
3. MySQL fait le travail que PHP lui avait soumis et lui répond "OK, c'est bon !"
4. PHP renvoie au serveur que MySQL a bien fait ce qui lui était demandé.

2- Structure d'une Base de Données

Supposons que notre base de données ressemble à une armoire...

Armoire sera appelée La base

Un tiroir sera appelé Une table

Une table ... va contenir Un tableau

Le tableau .. va contenir Des colonnes appelées des champs

Le tableau .. va contenir Des lignes appelées des entrées

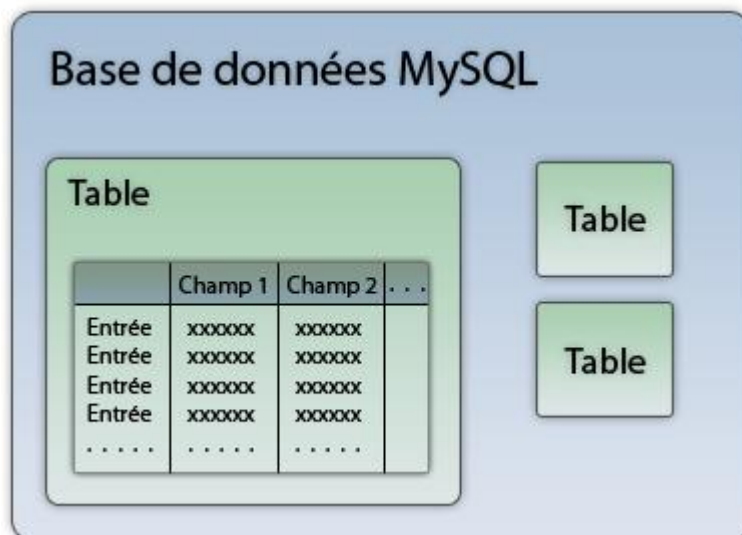
Voici à quoi va ressembler une table appelée « Visiteurs »

Table "visiteurs"

Numéro	Pseudonyme	E-mail	Age
1	Kryptonik	kryptonik@free.fr	24
2	Serial_Killer	serialkiller@unitedgamers.com	16
3	M@teo21	top_secret@siteduzero.com	18
4	Bibou	bibou557@laposte.net	29
...

Explication :

- les champs dans ce tableau sont : Numéro, Pseudonyme, E-mail et Âge
- Chaque ligne de 1 à 4 est une entrée (>>>> une table peut contenir jusque 100.000 entrées).



Rappel : La Base de Données contient plusieurs tables. Chaque table étant un tableau dans lequel on trouve des (colonnes) champs et des (lignes) entrées

3- Quel genre de table trouvent-on dans MySQL ?

- news : stocke toutes les news qui sont affichées à l'accueil.
- livre_or : stocke tous les messages postés sur le livre d'or.
- forum : stocke tous les messages postés sur le forum.
- newsletter : stocke les adresses e-mails de tous les visiteurs inscrits à la newsletter.

4- Résumé du module 1 consacré à MySQL

- Une base de données de type MySQL est un outil qui stocke vos données de manière organisée, afin de les retrouver plus facilement.

- On communique avec MySQL grâce au langage SQL. Langage commun à tous les systèmes de gestion de Base de Données (SGBD)

- PHP joue le rôle d'intermédiaire entre votre PC et MySQL
- Une Base de Données contient plusieurs tables.
- Chaque table est un tableau où les colonnes sont appelées « champs » et les lignes « entrées ».

Recommandation

Je terminerai ce premier module en faisant la recommandation suivante :

Comme vous l'avez remarqué plus haut, Microsoft utilise sa base de données SQL....
Il vous est sans doute déjà arrivé d'installer des mises à jour sur votre PC.

N'intervenez **JAMAIS** directement dans ces fichiers pour les modifier.

Ce premier module consacré à MySQL est terminé.

Module 2 – Comment manipuler une base de données

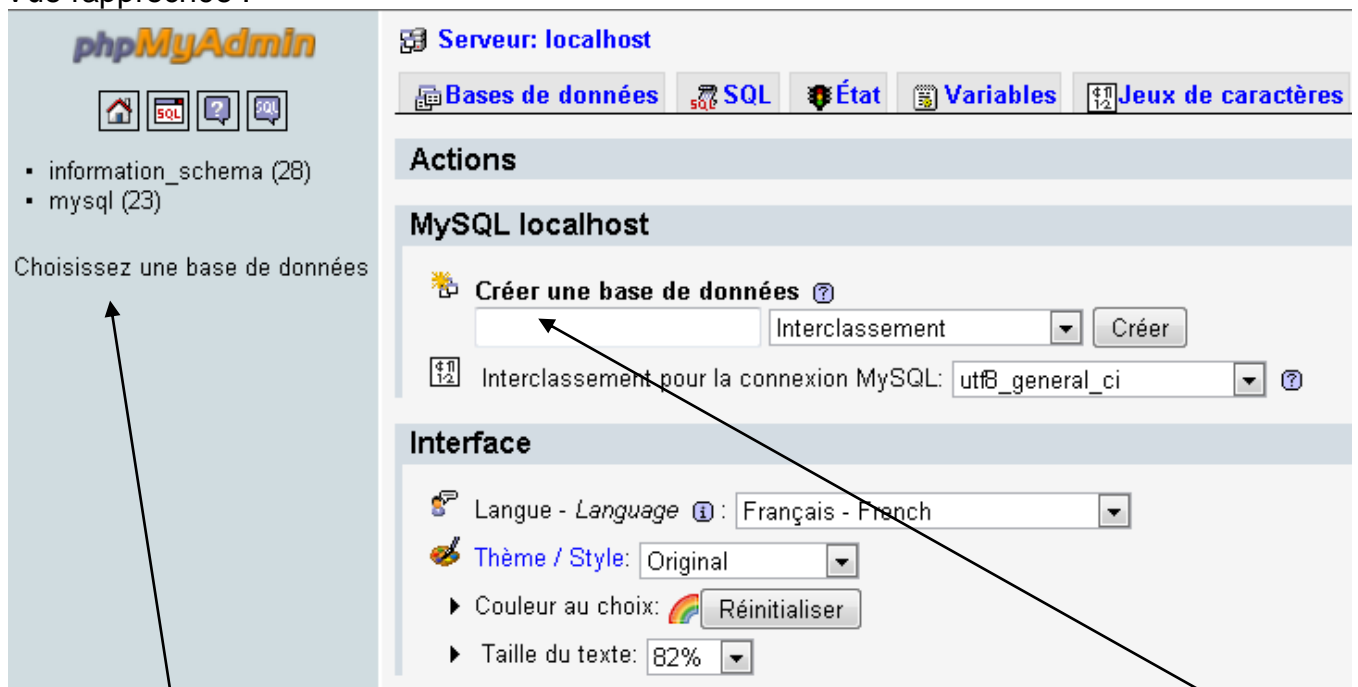
La création d'une base de données

Nous allons utiliser [phpMyAdmin] livré avec le package Wamp.
phpMyAdmin n'est pas un programme mais un ensemble de pages PHP déjà prêtes...

Interface d'accueil de phpMyAdmin



Vue rapprochée :



(1) ----- Remarquons deux endroits importants ----- (2)

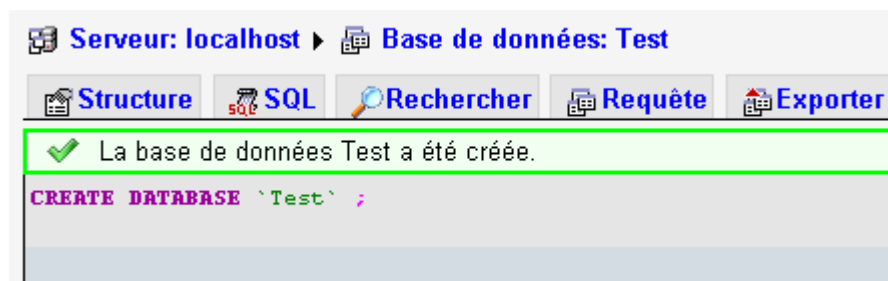
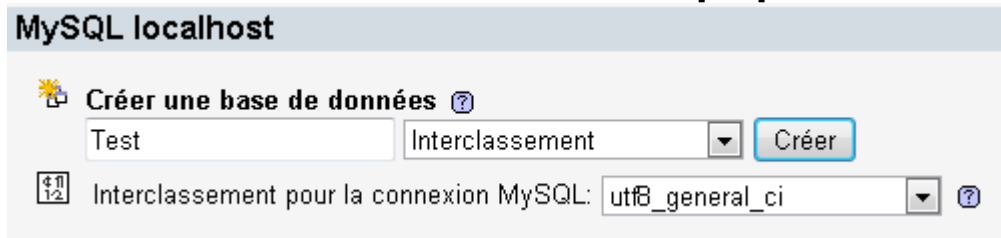
- (1) La liste des bases : C'est la liste de vos bases de données ; Le nombre entre parenthèses est le nombre de tables qu'il y a dans la base.
Il y a donc deux bases. Ces bases servent au fonctionnement interne de MySQL.
Il ne faut pas y toucher.

information_schema contient 28 tables
mysql contient 23 tables

(2) Créer une base : Pour créer une nouvelle base de données, entrez un nom dans la champ de formulaire à droite et cliquez sur « Créer ».

Rappel de recommandation : Ne touchez-pas aux bases existantes.

Nous allons maintenant créer une nouvelle base [test]



Notre nouvelle base a été créée, elle se nomme *Test*



Il est précisé « Aucune table n'a été trouvée dans cette base »
Nous allons donc la créer....

Nous allons entrer **news** en nouvelle table, et demander un nombre de champs égal à 3

Serveur: localhost ▶ Base de données: Test

Structure SQL Rechercher Requête Exporter

La base de données Test a été créée.

```
CREATE DATABASE `Test` ;
```

Aucune table n'a été trouvée dans cette base.

Créer une nouvelle table sur la base Test

Nom: Nombre de champs:

Cliquez sur le bouton « Exécuter » en bas à droite....

Voici un aperçu du tableau :

Nouvelle table

Nom de la table:

Structure

Colonne	<input type="text"/>	<input type="text"/>	<input type="text"/>
Type	<input type="text" value="INT"/>	<input type="text" value="INT"/>	<input type="text" value="INT"/>
Taille/Valeurs* ¹	<input type="text"/>	<input type="text"/>	<input type="text"/>
Défaut ²	<input type="text" value="Aucune"/>	<input type="text" value="Aucune"/>	<input type="text" value="Aucune"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>
Interclassement	<input type="text"/>	<input type="text"/>	<input type="text"/>
Attributs	<input type="text"/>	<input type="text"/>	<input type="text"/>
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index	<input checkbox"="" text"="" type="text" value="---</input></td> </tr> <tr> <td>AUTO_INCREMENT</td> <td><input type="/>	<input type="checkbox"/>	<input type="checkbox"/>

Commentaires

Commentaires sur la table:

Moteur de stockage: InnoDB

Interclassement:

Définition de PARTITION:

Sauvegarder Ou Ajouter 1 colonne(s) Exécuter

¹ Les différentes valeurs des colonnes de type enum/set sont à spécifier sous la forme 'a','b','c'...
 Pour utiliser un antislash("\") ou une apostrophe (") dans l'une de ces valeurs, faites-le précéder d'un antislash (par exemple '\\xyz' ou 'a\b').
² Pour les valeurs par défaut, veuillez n'entrer qu'une seule valeur, sans caractère d'échappement ou apostrophes, sous la forme: a

La création d'une Table

Serveur: localhost Base de données: Test Table: news

Champ			
Type	INT	INT	INT
Taille/Valeurs ¹			
Défaut ²	Aucun	Aucun	Aucun
Interclassement			
Attributs			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index	---	---	---
AUTO_INCREMENT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Commentaires			

Commentaires sur la table:

Moteur de stockage: MyISAM

Interclassement:

Définition de PARTITION:

- Nous allons maintenant créer les trois champs suivants :
- **id** : C'est le numéro d'identification. Toutes les entrées seront numérotées. Nous aurons ainsi la news 1 la news 2, la news 3.
 - **titre** : Ce champs contiendra le titre de la news
 - **contenu** : Ce champs contiendra la news elle-même.

Serveur: localhost ▶ Base de données: Test ▶ Table: news

Champ	id	titre	contenu
Type ?	INT	INT	INT
Taille/Valeurs* ¹			
Défaut ²	Aucun	Aucun	Aucun
Interclassement			
Attributs			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index	---	---	---
AUTO_INCREMENT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Commentaires			

Commentaires sur la table:

Moteur de stockage: ? MyISAM

Interclassement:

Définition de PARTITION: ?

Rappel : Chaque colonne représente un « champ » ; nous avons demandé trois champs, donc trois colonnes.

Pour l'instant, nous n'allons pas nous préoccuper des autres sections.... Sauf en ce qui concerne :

Index : Dans le menu déroulant, nous le placerons sur PRIMARY

Explications : Sert à activer l'indexation du champs. L'index PRIMARY est le mieux indiqué pour les champs de type id

AUTO_INCREMENT : Cette case doit être cochée. Cela permet au champ de s'incrémenter tout seul à chaque entrée.

>>>>> Voici ce que cela donne concernant la PREMIERE colonne

🏠 Serveur: localhost ▶ 📄 Base de données: Test ▶ 📄

Champ	id
Type ?	INT
Taille/Valeurs* ¹	
Défaut ²	Aucun
Interclassement	
Attributs	
Null	<input type="checkbox"/>
Index	PRIMARY
AUTO_INCREMENT	<input checked="" type="checkbox"/>
Commentaires	

Commentaires sur la table:

Définition de PARTITION: ?

Question Que signifie la valeur **INT** correspondant au type :
 Réponse.... Il s'agit de la valeur **nombre entier**.

>>>> Voyons maintenant la **DEUXIEME** colonne

titre
VARCHAR
255
Aucun
<input type="checkbox"/>

<input type="checkbox"/>

Moteur de stockage: ?

 MyISAM

Valeur **VARCHAR** : Pour un texte court.
 Valeur **255** : Valeur maximale (de 1 à 255). Cette case doit obligatoirement contenir une valeur pour que la création de la table soit validée.

>>>> Enfin, la TROISIEME colonne

contenu

TEXT

Aucun

Interclassement:

Valeur **TEXT** : Pour un **texte long** sans limite particulière.

Lorsque toutes ces valeurs ont été entrées dans le 3 champs ; cliquez sur « Sauvegarder ».
















Résultat :






Serveur: localhost ▶ Base de données: test ▶ Table: news

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Vider Supprimer

✓ La table 'test`.`news` a été créée.


```
CREATE TABLE `test`.`news` (
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `titre` VARCHAR( 255 ) NOT NULL ,
  `contenu` TEXT NOT NULL
) ENGINE = MYISAM ;
```



Champ	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/> id	int(11)			Non	Aucun	auto_increment	    
<input type="checkbox"/> titre	varchar(255)	latin1_swedish_ci		Non	Aucun		    
<input type="checkbox"/> contenu	text	latin1_swedish_ci		Non	Aucun		    

Tout cocher / Tout décocher Pour la sélection :     

Version imprimable Suggérer des optimisations quant à la structure de la table

Ajouter 1 champ(s) En fin de table En début de table Après id Exécuter

Index: 

Action	Nom de l'index	Type	Unique	Compressé	Champ	Cardinalité	Interclassement	Null	Commentaire
 	PRIMARY	BTREE	Oui	Non	id	0	A		

Créer un index sur 1 colonne(s) Exécuter

Conclusion :

Nous venons de créer une base de données appelée « test » dans laquelle nous avons créé une table appelée « news »

La liste détaillée des types

Si nous ouvrons la liste détaillée des types, un choix plus complet de type de données est proposé :

NUMERIC	Ce sont les nombres : Petits nombres (Tinyint), gros nombres (Bigint).
DATE and TIME	Types permettant de stocker une date, une heure, ou les deux à la fois.
STRING	Chaînes de caractères. Types adaptés à toutes les tailles.
SPATIAL	Types de données destinées à la cartographie.

Les quatre types de données les plus courants sont proposés en tête de liste :

INT	Nombre entier
VARCHAR	Texte court, entre 1 et 255 caractères
TEXT	Long texte
DATE	Date (jour, mois, année)

Les clés primaires

Toute table doit posséder un champ qui joue le rôle de clé primaire.

La clé primaire permet d'identifier de manière unique une entrée dans la table.

En général, on utilise le champ *id* comme clé primaire.

Chaque news de notre site devra être identifiée de manière unique.

Pour cela, il suffira de donner un numéro unique dans un champ nommé *id*

⇒ Il ne peut pas y avoir deux news dans la même *id*

C'est la raison pour laquelle l'index PRIMARY doit être associé au champ *id*
AUTO_INCREMENT fera le nécessaire pour que ce champ gère lui-même les nouvelles valeurs automatiquement (1, 2, 3 4...)

Rappel

Voici comment se présente le tableau :

Nouvelle table

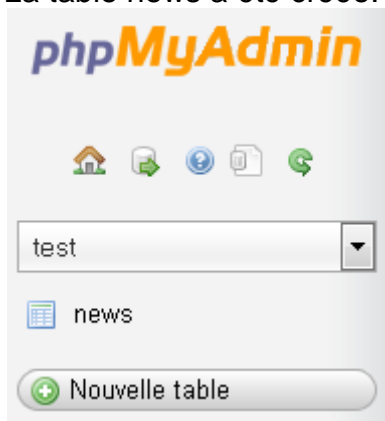
Nom de la table:

test

Structure

Colonne	id	titre	contenu
Type	INT	VARCHAR	TEXT
Taille/Valeurs* ¹		255	
Défaut ²	Aucune	Aucune	Aucune
Interclassement			
Attributs			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index	PRIMARY	---	---
AUTO_INCREMENT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cliquez sur « Sauvegarder »
La table news a été créée.



Cliquez sur la table news pour l'ouvrir....

```
SELECT *
FROM `news`
LIMIT 0, 30
```

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer plus
2	titre	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus
3	contenu	text	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus

Tout cocher / Tout décocher Pour la sélection : Afficher Modifier Supprimer Primaire Unique Index

Version imprimable Gestion des relations Suggérer des optimisations quant à la structure de la table

Ajouter 1 colonne(s) En fin de table En début de table Après id Exécuter

Index:

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
Modifier Supprimer	PRIMARY	BTREE	Oui	Non	id	0	A		

La table news ne contient encore aucune entrée.
 Le champ id est souligné car c'est la clé primaire de la table.
 Il est possible d'apporter des modifications : Ajouter des champs, les renommer, les supprimer, etc.....

Comment travailler à l'intérieur de cette nouvelle table

localhost > test > news

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

MySQL a retourné un résultat vide (aucune ligne). (Traitement en 0.0006 sec)

```
SELECT *
FROM `news`
LIMIT 0, 30
```

#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer plus
2	titre	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus
3	contenu	text	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus

Tout cocher / Tout décocher Pour la sélection : Afficher Modifier Supprimer Primaire Unique

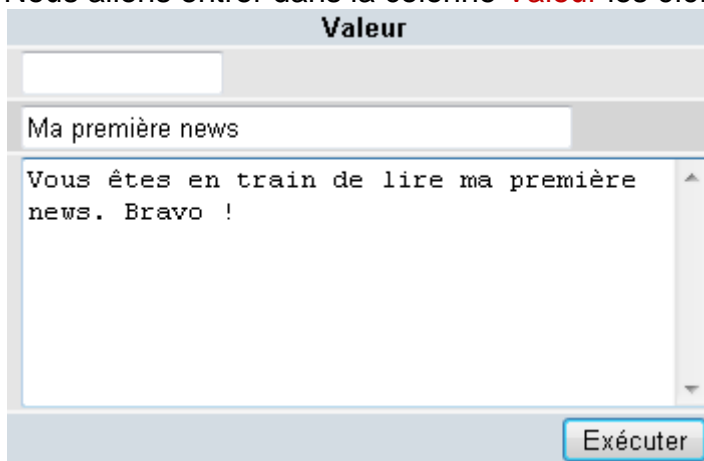
Pour manipuler cette base, nous allons utiliser les onglets .
 Premier onglet : Insérez

Nous allons ajouter une nouvelle entrée à cette table.



Colonne	Type	Fonction	Null	Valeur
id	int(11)			
titre	varchar(255)			
contenu	text			

Nous allons entrer dans la colonne **Valeur** les éléments suivants :



Valeur
Ma première news
Vous êtes en train de lire ma première news. Bravo !

Exécuter

Cliquez sur « Exécuter »



localhost ▶ test ▶ news

Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations

✓ 1 ligne insérée.
Identifiant de la ligne insérée : 1

```
INSERT INTO `test`.`news` (  
  `id`,  
  `titre`,  
  `contenu`  
)  
VALUES (  
  NULL, 'Ma première news', 'Vous êtes en train de lire ma première news. Bravo !'  
);
```

Exécuter une ou des requêtes SQL sur la base test:

```
INSERT INTO `test`.`news` (`id`, `titre`, `contenu`) VALUES (NULL, 'Ma première news', 'Vous êtes en train de lire ma première news. Bravo !');
```

Ajoutez maintenant deux entrées en procédant de la même manière.

Affichons maintenant le contenu de la table news :

	id	titre	contenu
<input type="checkbox"/> Modifier Éditer en place Copier Effacer	1	Ma première news	Vous êtes en train de lire ma première news. Bravo...
<input type="checkbox"/> Modifier Éditer en place Copier Effacer	2	Autres news	Ceci est une autre news !
<input type="checkbox"/> Modifier Éditer en place Copier Effacer	3	Exclusif !	Ceci est une news !

Cette table news contient maintenant trois entrées.
Vous constatez que les n° *id* se sont créés tout seuls.

Question

A chaque fois que je souhaite ajouter ou supprimer un élément de la table, je vais donc devoir passer par phpMyAdmin ?

Réponse

Non. phpMyAdmin est un outil d'administration.
Il permet de voir rapidement la structure et le contenu des tables.
Il faudra créer des pages en PHP qui insèrent ou suppriment des éléments directement depuis le site Web.

Les onglets de la table news



Rappel : Jusqu'à présent, nous avons utilisé trois onglets :

Afficher	<p>Affiche le contenu de la table</p>																																				
Structure	<p>Présente la structure de la table (Liste des champs)</p> <table border="1"> <thead> <tr> <th>#</th> <th>Colonne</th> <th>Type</th> <th>Interclassement</th> <th>Attributs</th> <th>Null</th> <th>Défaut</th> <th>Extra</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> 1</td> <td>id</td> <td>int(11)</td> <td></td> <td></td> <td>Non</td> <td>Aucune</td> <td>AUTO_INCREMENT</td> <td> Modifier Supprimer plus ▼</td> </tr> <tr> <td><input type="checkbox"/> 2</td> <td>titre</td> <td>varchar(255)</td> <td>latin1_swedish_ci</td> <td></td> <td>Non</td> <td>Aucune</td> <td></td> <td> Modifier Supprimer plus ▼</td> </tr> <tr> <td><input type="checkbox"/> 3</td> <td>contenu</td> <td>text</td> <td>latin1_swedish_ci</td> <td></td> <td>Non</td> <td>Aucune</td> <td></td> <td> Modifier Supprimer plus ▼</td> </tr> </tbody> </table>	#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action	<input type="checkbox"/> 1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer plus ▼	<input type="checkbox"/> 2	titre	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼	<input type="checkbox"/> 3	contenu	text	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼
#	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action																													
<input type="checkbox"/> 1	id	int(11)			Non	Aucune	AUTO_INCREMENT	Modifier Supprimer plus ▼																													
<input type="checkbox"/> 2	titre	varchar(255)	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼																													
<input type="checkbox"/> 3	contenu	text	latin1_swedish_ci		Non	Aucune		Modifier Supprimer plus ▼																													
Insérer	<p>Permet d'insérer de nouvelles entrées dans la table.</p>																																				

Colonne	Type	Fonction	Null	Valeur
id	int(11)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
titre	varchar(255)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
contenu	text	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>

Voyons maintenant les autres onglets utiles :

Onglet SQL

Afficher Structure SQL Rechercher

Exécuter une ou des requêtes SQL sur la base test: 

```
SELECT * FROM `news` WHERE 1
```

C'est ici que nous pouvons exécuter ce que l'on appelle des requêtes SQL.

Les requêtes seront tapées dans la zone de texte.

```
SELECT * FROM `news` WHERE 1
```

Signifie : « Afficher tout le contenu de la table news »

Onglet Importer

Afficher Structure SQL Rechercher Insérer Exporter

Importation dans la table «news»

Fichier à importer:

Le fichier peut être comprimé (gzip, zip) ou non.

Le nom du fichier comprimé doit se terminer par **[format].[compression]**. Exemple: **.sql.zip**

Parcourir : Aucun fichier sélectionné. (Taille maximum: 2 048Kio)

Jeu de caractères du fichier :

Dans « Parcourir », rechercher un fichier [nom.sql] sur notre disque dur et cliquer sur « Exécuter ».

Onglet Exporter

Exportation des lignes de la table «news»

Méthode d'exportation:

- Rapide - n'afficher qu'un minimum d'options
- Personnalisée - afficher toutes les options possibles

Format:

SQL

Exécuter

Va permettre de récupérer la base de données sur le disque dur :

Pour ensuite, pouvoir par exemple :

⇒ Transmettre la base de données sur Internet, chez notre hébergeur.

⇒ Faire une copie de sauvegarde de la base de données.

En mode personnalisé, veillez à ce que la sauvegarde soit complète, c'est-à-dire structure ET données.

Exporter la table

- structure
- données
- structure et données

Onglet OPERATIONS

Ordonner la table par


id ▼ (à refaire après insertions/destructions) Croissant ▼

Options pour cette table

Changer le nom de la table pour

news

Commentaires sur la table

Moteur de stockage 

InnoDB ▼

Interclassement

latin1_swedish_ci ▼

AUTO_INCREMENT

4

ROW_FORMAT

COMPACT ▼

- Pour changer le nom de la table
- Pour déplacer la table vers une autre base de données
- Copier la table dans une autre base ou dans la même (Dans ce cas, il faudra qu'elle ait un nom différent).
- Optimiser la table : Concerne surtout la réorganisation automatique des bases lorsque celles-ci sont particulièrement volumineuses.

Réorganiser une table

>>>> J'ai effectué plusieurs autres insertions à la volée...

>>>> Voici maintenant le contenu affiché de la table

Afficher : 30 enregistrement(s) à partir de l'enregistrement n° 0
 en mode horizontal et répéter les en-têtes à chaque groupe de 100
 Trier sur l'index: aucune

+ Options

	id	titre	contenu
<input type="checkbox"/>	1	Ma première news	Vous êtes en train de lire ma première news. Bravo...
<input type="checkbox"/>	2	Autre news	Ceci est une autre news !
<input type="checkbox"/>	3	Exclusif !	Ce est une news !
<input type="checkbox"/>	9	Autre news	Ceci est une autre news !
<input type="checkbox"/>	10	Exclusif !	Ce est une news !
<input type="checkbox"/>	11	Autre news	Ceci est une autre news !
<input type="checkbox"/>	12	Exclusif !	Ce est une news !
<input type="checkbox"/>	13	Autre news	Ceci est une autre news !
<input type="checkbox"/>	14	Exclusif !	Ce est une news !

↑ Tout cocher / Tout décocher Pour la sélection :   

Afficher : 30 enregistrement(s) à partir de l'enregistrement n° 0
 en mode horizontal et répéter les en-têtes à chaque groupe de 100




>>>> Vous constatez que certaines **entrées** sont en doublon.... Et que l'ordre défini dans la colonne **id** n'est pas régulier.. C'est parce que j'ai procédé à des annulations. Nous allons maintenant procéder à un nettoyage de cette table **news**, en utilisant l'option **modifier.** :

Voilà le résultat

Afficher : 30 enregistrement(s) à partir de l'enregistrement n° 0
 en mode horizontal et répéter les en-têtes à chaque groupe de 100
 Trier sur l'index: aucune

+ Options

	id	titre	contenu
<input type="checkbox"/>	1	Ma première news	Vous êtes en train de lire ma première news. Bravo...
<input type="checkbox"/>	2	Autre news	Ceci est une autre news !
<input type="checkbox"/>	3	Exclusif !	Ce est une news !
<input type="checkbox"/>	4	Dernière news	Voici la dernière news !

↑ Tout cocher / Tout décocher Pour la sélection :   

Afficher : 30 enregistrement(s) à partir de l'enregistrement n° 0
 en mode horizontal et répéter les en-têtes à chaque groupe de 100

Et voilà, comme vous pouvez le constater, j'ai supprimé les **id** 9, 10, 11,12,13 et 14 J'ai rajouté une entrée sous **id** 4 intitulée « Dernière news ».

Synthèse de MySQL Module 2

Dans ce second module consacré à MySQL, nous avons d'abord ouvert le programme phpMyAdmin, composant de Wamp puis, nous avons successivement :

- 1- Visualisé les Bases de Données existantes,
- 2- Créé une nouvelle Base de Données (test)
- 3- Créé une nouvelle table (news)
- 4- Inséré plusieurs entrées
- 5- Supprimé des entrées créées, et réorganisé l'ordre des (id)

Par contre, nous avons fait connaissance avec certains nouveaux termes :

INT

PRIMARY

AUTO_INCREMENT

VARCHAR

255

TEXT

.... Et d'autres encore !

Module 3 – Lire des données avec PHP – PDO

L'objectif maintenant est de nous connecter à la base de données en PHP. PHP propose plusieurs types d'extensions pour permettre de nous connecter. Certaines de ces extensions sont considérées comme obsolètes : mysql_

Les plus utilisées sont actuellement mysqli_ et PDO
PDO est appelée à devenir l'extension la plus utilisée.
PDO est une extension Programmation orientée Objet (PHP Data Objects)

Activer PDO sur Wamp

- Clic gauche sur l'icône de lancement de Wamp
- PHP
- Extension PHP
- Dans la liste déroulante : vérifier que php_pdo_mysql est bien coché.

4 éléments pour se connecter à PDO

- 1- Le nom de l'hôte : en mode local, ce sera localhost
- 2- La base : Nom de la base de données : ici... test
- 3- Le login : login FTP de l'hébergeur . Ici : root
- 4- Le mot de passe : Pas de mot de passe en localhost, donc le champ sera vide "

Ce qui donnera en code PHP :

```
( ?php
$dbdd = new PDO('mysql :host=localhost ;dbname=test', 'root', '' );
?>
```

La ligne de code crée un objet \$bdd
Ce n'est pas vraiment une variable. On crée une connexion en indiquant dans l'ordre les 4 paramètres.

Tester la présence d'erreurs

Il est recommandé de traiter l'erreur que peut contenir un code. PDO renvoie alors ce que l'on appelle une exception qui permet de capturer l'erreur.

```
< ?php
try
{
$dbdd = new PDO('mysql :host=localhost ;dbname=test', 'root', '' );
}
catch (Exception $e)
{
die ('Erreur : ' . $e->getMessage() );
}
?>
```

- ⇒ PHP essaye d'exécuter les instructions à l'intérieur du bloc try
- ⇒ S'il y a une erreur, il rentre dans le bloc catch
L'exécution de la page est arrêtée et affiche un message décrivant l'erreur.
- ⇒ Si tout se passe bien, PHP ignore les fonctions du bloc catch

La récupération des données

Pour récupérer des données, il faut effectuer une requête (query)

```
$reponse = $bdd->query('SELECT * FROM news');
```

- ⇒ \$bdd : Objet représentant la connexion à la base de données test
- ⇒ query : requête
- ⇒ SELECT : Demande faite à MySQL d'afficher ce que contient la table
- ⇒ * : Etoile entre SELECT et FROM : Après SELECT, on doit indiquer quels champs MySQL doit récupérer dans la table. * veut dire « tout »
- On peut donc taper au choix :
SELECT titre, contenu FROM news (ou) SELECT * FROM news
SELECT titre FROM news
SELECT contenu FROM news
- ⇒ FROM : Signifie « dans »
- ⇒ news : Nom de la table dans laquelle il faut aller récupérer les données

Afficher le résultat

Pour ne pas être submergé d'informations par une table trop volumineuse, il faut procéder à une extraction ligne par ligne : c'est-à-dire entrée par entrée.

Nous allons donc exécuter la commande fetch qui signifie « va chercher »

```
< ?php
$donnees = $reponse->fetch();
?>
```

- ⇒ fetch : va chercher
- ⇒ \$donnees : array qui contient champ par champ les valeurs de la première entrée
Pour un seul champ : \$donnees['contenu'].
- ⇒ \$reponse->fetch() : Il s'agit d'une boucle pour parcourir les entrées une par une.
La boucle est donc répétée autant de fois qu'il y a d'entrées dans la table.

Pour afficher les entrées une par une

```
// On affiche chaque entrée une à une
while ($donnees = $reponse->fetch())
{
echo $donnees['titre']. '<br />';
echo $donnees['contenu']. '<br />';
}
```

Pour terminer le traitement de la requête

```
$reponse->closeCursor(); // Termine le traitement de la requête
```

Connexion à la base MySQL en PHP

Voici maintenant le code PHP au complet

```
<?php
try
{
    // On se connecte à MySQL
    $bdd = new PDO('mysql:host=localhost;dbname=test', 'root', '');
}
catch(Exception $e)
{
    // En cas d'erreur, on affiche un message et on arrête tout
    die('Erreur : '.$e->getMessage());
}

// Si tout va bien, on peut continuer

// On récupère tout le contenu de la table news
$reponse = $bdd->query('SELECT * FROM news');

// On affiche chaque entrée une à une
while ($donnees = $reponse->fetch())
{
    echo $donnees['titre']. '<br />';
    echo $donnees['contenu']. '<br />';
}

$reponse->closeCursor(); // Termine le traitement de la requête

?>
```

Enregistrons ce fichier en connect.php

Et plaçons le dans le répertoire www de Wamp, dans le dossier dédié au site

Ici : maquettefptphp

Et voici le résultat :


```
<?php
try
{
    // On se connecte à MySQL
    $bdd = new PDO('mysql:host=localhost;dbname=test',
    'root', '');
}
catch(Exception $e)
{
    // En cas d'erreur, on affiche un message et on arrête
    tout
    die('Erreur : '.$e->getMessage());
}

// Si tout va bien, on peut continuer

// On récupère tout le contenu de la table jeux_video
$reponse = $bdd->query('SELECT * FROM news');

// On affiche chaque entrée une à une
while ($donnees = $reponse->fetch())
{
    echo $donnees['titre']. '<br />';
    echo $donnees['contenu']. '<br />';
}

$reponse->closeCursor(); // Termine le traitement de la
requête
?>
```

Ma première news
Vous êtes en train de lire ma première news. Bravo !
Autres news
Ceci est une autre news !
Exclusif !
Ceci est une news !

⇒ Différence entre \$reponse et \$donnees

\$reponse contenait toute la réponse de MySQL en vrac, sous forme d'objet.

\$donnees est un array renvoyé par le fetch ()

Chaque fois qu'on fait une boucle, fetch va chercher dans \$reponse l'entrée suivante et organise les champs dans l'array \$donnees.

⇒ Que signifie while (\$donnees = \$reponse ->fetch())

La ligne récupère une nouvelle entrée et place son contenu dans \$donnees ;

Elle vérifie si \$donnees est vrai ou faux

Le fetch renvoie faux (false) dans \$donnees lorsqu'il est arrivé à la fin des données, c'est-à-dire que toutes les données ont été passées en revue. Dans ce cas, la condition du while vaut faux et la boucle s'arrête.

⇒ Que signifie la ligne < ?php \$reponse->closeCursor() ; ?>

Elle provoque la fermeture du curseur d'analyse des résultats.

Il faut avoir fini de traiter le retour d'une requête afin d'éviter d'avoir des problèmes à la requête suivante .

⇒ **Que signifie echo \$donnees['titre'].**
Ou **echo \$donnees['contenu'].**

Cette ligne de code permet d'afficher seulement le contenu des champs choisis

Ci-dessous, j'ai supprimé l'affichage du champ ['contenu'] en le mettant sous forme de commentaire.

```
<?php
try
{
    // On se connecte à MySQL
    $bdd = new PDO('mysql:host=localhost;dbname=test',
    'root', '');
}
catch(Exception $e)
{
    // En cas d'erreur, on affiche un message et on arrête
    tout
    die('Erreur : '.$e->getMessage());
}

// Si tout va bien, on peut continuer

// On récupère tout le contenu de la table jeux_video
$reponse = $bdd->query('SELECT * FROM news');

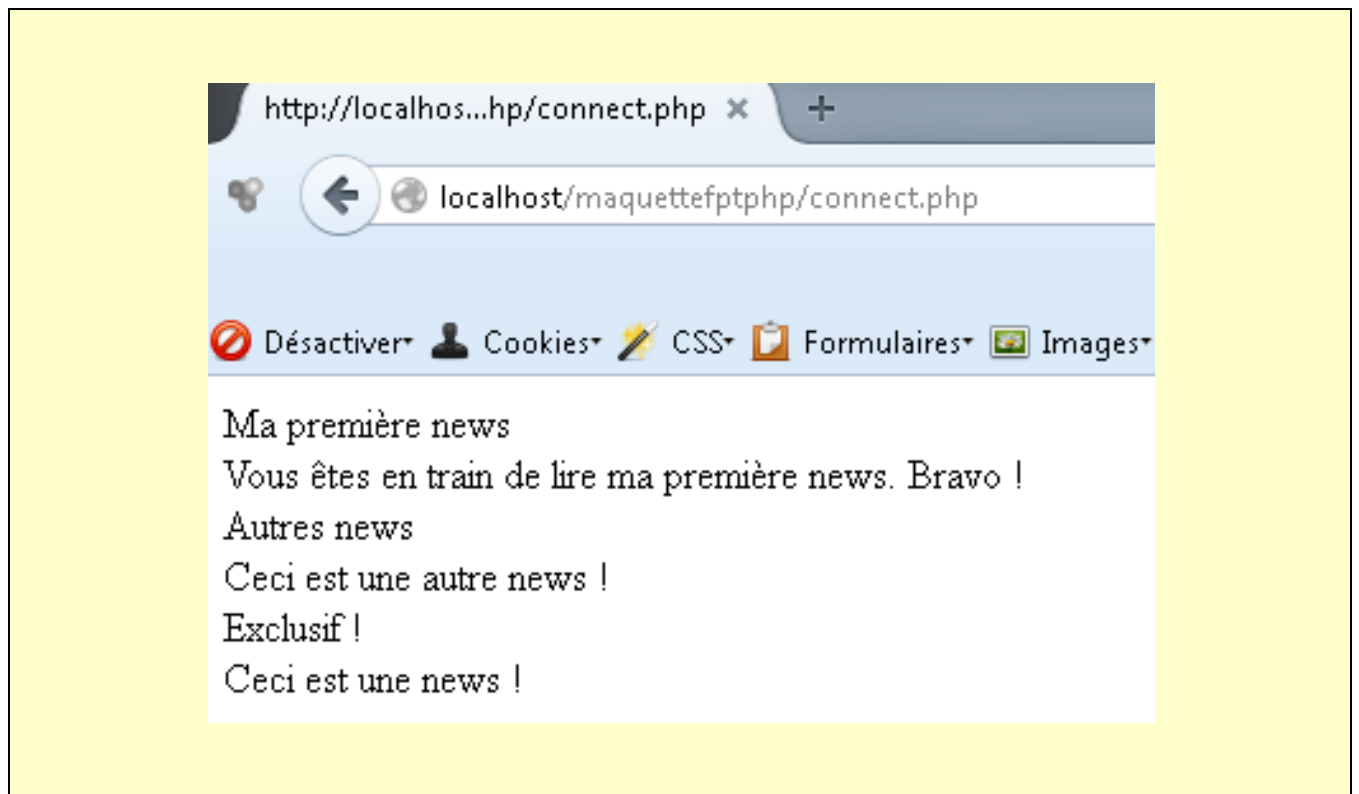
// On affiche chaque entrée une à une
while ($donnees = $reponse->fetch())
{
    echo $donnees['titre']. '<br />';

    /*
    echo $donnees['contenu']. '<br />';
    */
}
$reponse->closeCursor(); // Termine le traitement de la
requête
?>
```

Ma première news
Autres news
Exclusif !

Connexion directe à partir du navigateur Firefox

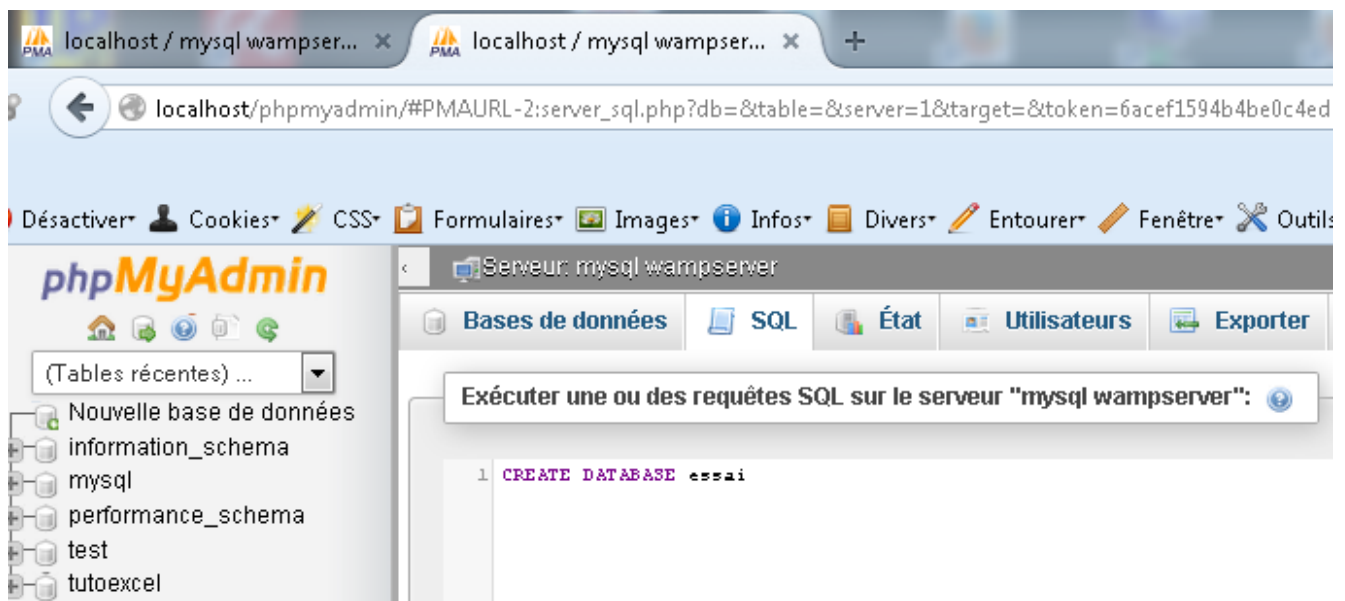
Dans la barre URL, je rentre la ligne suivante : `http://localhost/maquettefptphp/connect.php`



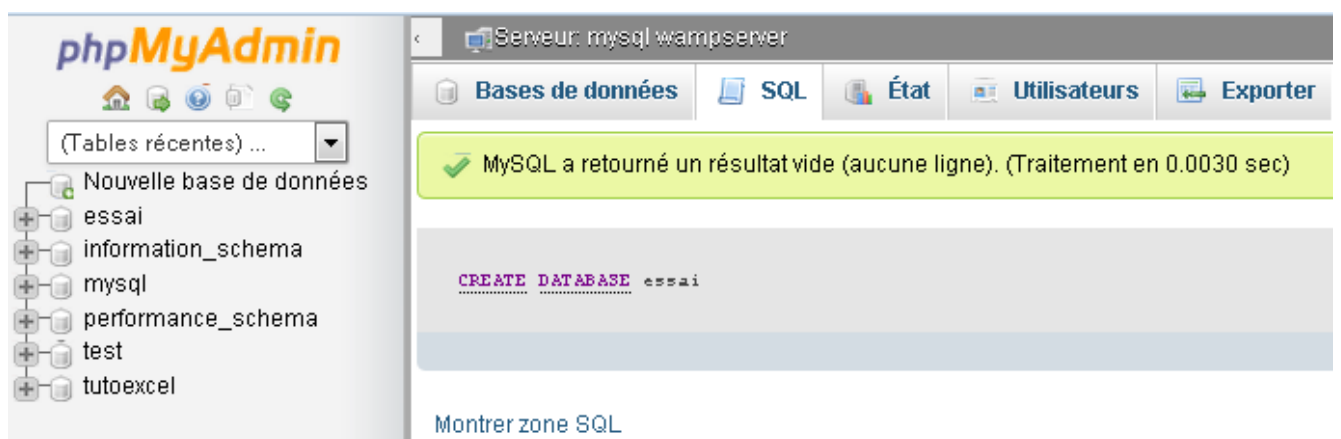
Module 4 – Créer des bases de données et des tables en requêtes SQL à partir de PhpMyadmin

4-1. Création d'une base de données

Ouvrez PhpMyadmin – Panneau des bases de données.
Cliquez sur l'onglet SQL
Entrez la ligne d'instruction : CREATE DATABASE (nom de la base)
Ici : CREATE DATABASE essai
Cliquez sur Exécuter.



La base [essai] est maintenant créée, et il est 'évidemment' mentionné qu'elle ne contient aucune données.



4-2. Créer une table dans la base de données [essai]

Nous allons maintenant créer une table appelée [famille_tbl] dans la base de données [essai].

Se positionner dans la base de données nouvellement créée.

Ouvrez l'onglet SQL et entrez les lignes d'instruction suivante :

```
CREATE TABLE famille_tbl ( id int(11) NOT NULL auto_increment, nom varchar(255) NOT NULL, prenom varchar(255) NOT NULL, statut varchar(255) NOT NULL, date date DEFAULT '0000-00-00' NOT NULL, PRIMARY KEY (id), KEY id (id), UNIQUE id_2 (id) );

INSERT INTO famille_tbl VALUES( '', 'Dupond', 'Grégoire', 'Grand-père', '1932-05-17');
INSERT INTO famille_tbl VALUES( '', 'Dupond', 'Germaine', 'Grand-mère', '1939-02-15');
INSERT INTO famille_tbl VALUES( '', 'Dupond', 'Gérard', 'Père', '1959-12-22');
INSERT INTO famille_tbl VALUES( '', 'Dupond', 'Marie', 'Mère', '1961-03-02');
INSERT INTO famille_tbl VALUES( '', 'Dupond', 'Julien', 'Fils', '1985-05-17');
IN
SERT INTO famille_tbl VALUES( '', 'Dupond', 'Manon', 'Fille', '1990-11-29');
```

Ces lignes d'instruction ont été créées à partir du tableau suivant :

id	nom	prenom	statut	date
1	Dupond	Grégoire	Grand-père	1932-05-17
2	Dupond	Germaine	Grand-mère	1939-02-15
3	Dupond	Gérard	Père	1959-12-22
4	Dupond	Marie	Mère	1961-03-02
5	Dupond	Julien	Fils	1985-05-17
6	Dupond	Manon	Fille	1990-11-29

Cliquez sur Exécuter.

La table [famille_tbl] est visible dans le panneau de contrôle de la Bdd [essai]

The screenshot shows the phpMyAdmin interface for a database named 'essai'. The table 'famille_tbl' is selected, and its structure and data are displayed. The table has 6 rows and 5 columns: id, nom, prenom, statut, and date. The data is as follows:

id	nom	prenom	statut	date
1	Dupond	Grégoire	Grand-père	1932-05-17
2	Dupond	Germaine	Grand-mère	1939-02-15
3	Dupond	Gérard	Père	1959-12-22
4	Dupond	Marie	Mère	1961-03-02
5	Dupond	Julien	Fils	1985-05-17
6	Dupond	Manon	Fille	1990-11-29

4-3. Comment faire apparaître le tableau [famille_tbl] en Php

Nous allons éditer un fichier Php qui se nommera : tableau.php

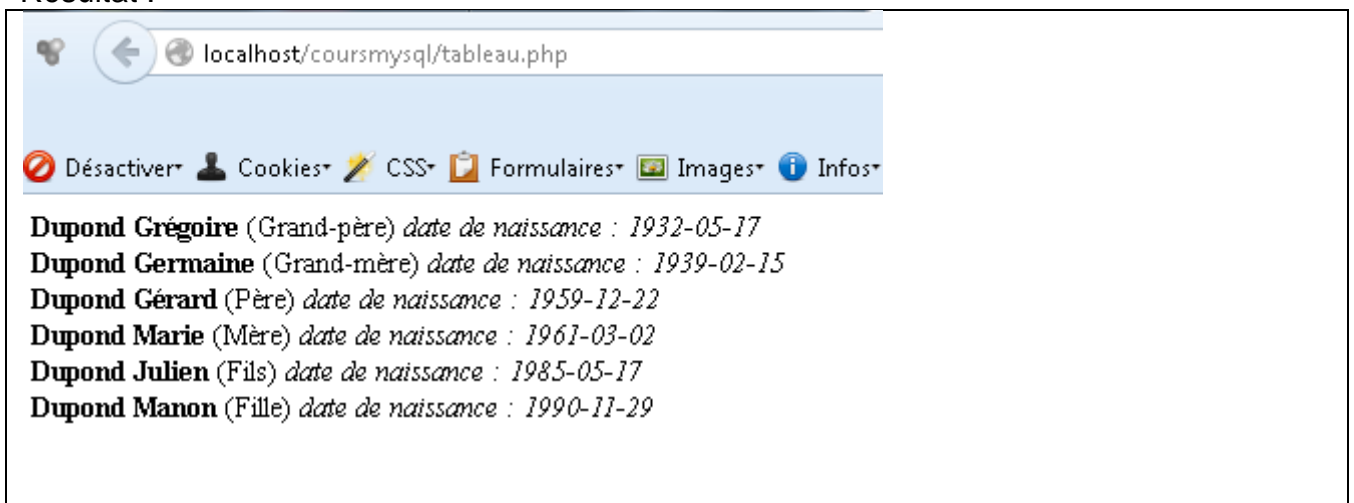
```
<?php
try
{
    // On se connecte à MySQL
    $bdd = new PDO('mysql:host=localhost;dbname=essai', 'root', '');
}
catch(Exception $e)
{
    // En cas d'erreur, on affiche un message et on arrête tout
    die('Erreur : '.$e->getMessage());
}
// Si tout va bien, on peut continuer
// On récupère tout le contenu de la table famille_tbl
$reponse = $bdd->query('SELECT * FROM famille_tbl');

// On affiche chaque entrée une à une
// on fait une boucle qui va faire un tour pour chaque enregistrement
while($data = $reponse->fetch())
{
    // on affiche les informations de l'enregistrement en cours
    echo '<b>'.$data['nom'].' '.$data['prenom'].'</b> ('.$data['statut'].')';
    echo ' <i>date de naissance : '.$data['date'].'</i><br>';
}
$reponse->closeCursor(); // Termine le traitement de la requête
?>
```

4-4. Contrôle du résultat à partir du navigateur

Dans la barre URL du navigateur, entrez le chemin de votre fichier [tableau.php]

Résultat :



Module 5 – Ecrire des données

5-1. Création de la Bdd et de la table vide

Dans ce paragraphe, nous avons créé une base de données [tutoexcel] composée de champs, puis, nous avons créé une table [famille].

A ce stade de notre cours, nous ne savons pas encore créer la ligne d’instruction qui servira à déclarer notre table dans la base de données. En conséquence, nous utilisons un modèle :

```
CREATE TABLE famille ( id int(11) NOT NULL auto_increment, nom varchar(255) NOT NULL, prenom varchar(255) NOT NULL, statut varchar(255) NOT NULL, date date DEFAULT '0000-00-00' NOT NULL, PRIMARY KEY (id), KEY id (id), UNIQUE id_2 (id) );
```

 Version imprimable  Dictionnaire de données

 Nouvelle table

Nom: Nombre de colonnes:

Créons les 5 champs, ce qui nous donnera ceci :
Voir le § Création d’une table – Page 8

Rappel :

id	Primary
Nom	VARCHAR
Prenom	VARCHAR - 255
Statut	VARCHAR - 255
date	Dans la liste déroulante

Notre table [famille] est vide. Nous allons maintenant entrer les lignes par [insérer]

5-2. Insérer des lignes dans les champs

Depuis la table [famille]
Onglet [Insérer]

Colonne	Type	Fonction	Null	Valeur
id	int(11)			
nom	varchar(255)			Dupond
prenom	varchar(255)			Grégoire
statut	varchar(255)			Grand-père
date	date		<input type="checkbox"/>	1932-05-17

Exécuter

Ignorer

Colonne	Type	Fonction	Null	Valeur
id	int(11)			
nom	varchar(255)			Dupond
prenom	varchar(255)			Germaine
statut	varchar(255)			Grand-mère
date	date		<input type="checkbox"/>	1932-02-19

Exécuter

Remarque : Il est inutile de lister la valeur de l'id, puisque l'incrémentatation est automatique.

Exécuter : Les deux lignes sont enregistrées.

Afficher	Structure	SQL	Rechercher	Insérer	Exporter	Importer	Privilèges	Opérations	Déclencheurs
<p>✓ 2 lignes insérées. Identifiant de la ligne insérée : 5</p>									
<pre>INSERT INTO `tutoexcel`.`famille` (`id`, `nom`, `prenom`, `statut`, `date`) VALUES (NULL, 'Dupond', 'Grégoire', 'Grand-père', '1932-05-17'), (NULL, 'Dupond', 'Germaine', 'Grand-mère', '1932-02-19');</pre>									
<p>Exécuter une ou des requêtes SQL sur la base tutoexcel: <input type="button" value="ⓘ"/></p> <pre>1 INSERT INTO `tutoexcel`.`famille` (`id`, `nom`, `prenom`, `statut`, `date`) VALUES (NULL, 'Dupond', 'Grégoire', 'Grand-père', '1932-05-17'), (NULL, 'Dupond', 'Germaine', 'Grand-mère', '1932-02-19');</pre>									

Contrôle dans la table [famille]

+ Options

	id	nom	prenom	statut	date
<input type="checkbox"/>  Modifier  Copier  Effacer	4	Dupond	Grégoire	Grand-père	1932-05-17
<input type="checkbox"/>  Modifier  Copier  Effacer	5	Dupond	Germaine	Grand-mère	1932-02-19

Rappel

Ligne à entrer	INSERT INTO famille VALUES(' ', 'Dupond', 'Grégoire', 'Grand-père', '1932-05-17');
INSERT TO	Permet d'ajouter une entrée - Entrez le nom de la table - Lister entre parenthèses les noms des champs dans lesquels vous souhaitez placer des informations. Exemple : INSERT TO famille (nom, prénom, statut, date) Note : Il n'est pas nécessaire de lister le champ <i>id</i>
VALUES	Insérer les valeurs dans le même ordre des champs listés. INSERT TO famille (nom, prénom, statut, date) VALUES ('Dupond', 'Grégoire', 'Grand-père', '1932-05-17');


5-3. Créons maintenant nous-mêmes notre modèle de table SQL [essai]

```
CREATE TABLE essai (  
numero INT (5) NOT NULL AUTO_INCREMENT,  
nom VARCHAR (20) NOT NULL,  
prenom VARCHAR (20) NOT NULL,  
telephone VARCHAR (14) NOT NULL,  
mail VARCHAR (20) NOT NULL,  
PRIMARY KEY (numero)  
);
```

Exécuter une ou des requêtes SQL sur la base tutoexcel: 

```
1 CREATE TABLE essai (  
2 numero INT (5) NOT NULL AUTO_INCREMENT,  
3 nom VARCHAR (20) NOT NULL,  
4 prenom VARCHAR (20) NOT NULL,  
5 telephone VARCHAR (14) NOT NULL,  
6 mail VARCHAR (20) NOT NULL,  
7 PRIMARY KEY (numero)  
8 );  
9
```

La table est enregistrée :

 MySQL a retourné un résultat vide (aucune ligne). (Traitement en 0.3588 sec)

```
CREATE TABLE essai ( numero INT (5) NOT NULL AUTO_INCREMENT, nom VARCHAR (20) NOT NULL, prenom VARCHAR (20) NOT NULL, telephone VARCHAR (14) NOT NULL, mail VARCHAR (20) NOT NULL, PRIMARY KEY (numero) )
```

NOT NULL	Le champ doit obligatoirement contenir une valeur.
VARCHAR (20)	(..) indique le nombre de caractères. Exemple : code postal VARCHAR (5)