

Programmation Web

Licence 2 : Informatique (ISIL)

MySQL+PHP



TP 7:
Persistante des données

- L'objectif de ce TP est de vous familiariser avec la persistante des données avec PHP et MySQL.
- Vous allez mettre ceci en pratique à travers 16 exercices.
- Le fichier **TP7_Bases_Données.pdf** vous guidera étape par étape pour la concrétisation de ce TP.
- Votre code doit impérativement être commenter. Un code sans commentaire est automatiquement rejeté.
- Les nom des id, classe et variables doit minutieusement correspondre à ceux données dans chaque exercice.
- Dans l'exo 16: La base de données doit s'appeler **db_persons** (exo 4) et la table **person**.
- Les paramètres du serveur SGBD doivent être comme suit:
 - **Host:** Localhost **Port:** 3306 **User:** 'root' **Password:** " (reste vide)
- L'énoncé du TP est déposé sur Github. Le lien de l'assignement: est donné dans google classroom.
- **Livrables attendus:**
 - Les fichiers index.html, formAdd.php, formUpdate, formDelete.php, addPerson.php, updatePerson.php, deletePerson.php, dataJson.php
 - style.css

MySQL

- **Exo1:** Installation
- **Exo2:** Démarrage/Arrêt



MySQL Client SQL (cmd)

- **Exo3:** Lancement
- **Exo4:** Création d'une BD
- **Exo5:** Consultation de la liste des BD
- **Exo6:** Création d'une Table
- **Exo7:** Consultation des Tables
- **Exo8:** Lire le contenu d'une Table
- **Exo9:** Afficher la structure d'une Table
- **Exo10:** CRUD sur une Table
- **Exo11:** MySQL: Client SQL Graphique (Lancement)



PHP+MySQL

- **Exo13:** Formulaire: Mise à jour
- **Exo14:** Formulaire: Suppression
- **Exo15:** Sélection
- **Exo16:** Mini-Appli: Gestion Persons (Ajouter)



À vous

Exercice 1

MySQL: Installation

Suivez les étapes décrites dans le cours

À vous

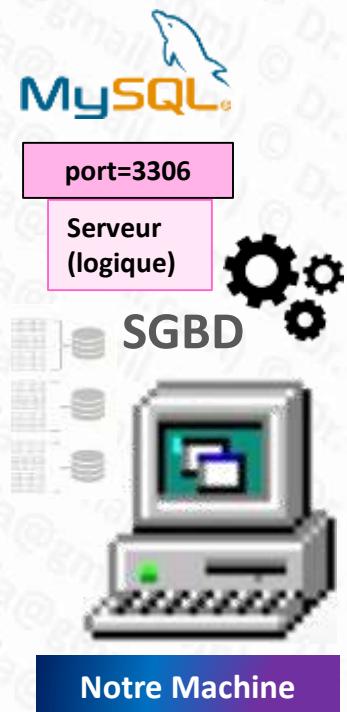
Exercice 2

MySQL: Démarrage/Arrêt

Suivez les étapes décrites dans le cours
(Slides 61-65)

Exercice 2 MySQL: Démarrage/Arrêt

En faisant l'étape précédente, vous avez désormais un serveur de SGDB MySQL lancé sur votre machine, et qui écoute sur le port 3306



Exercice 3

MySQL: Client SQL (cmd)

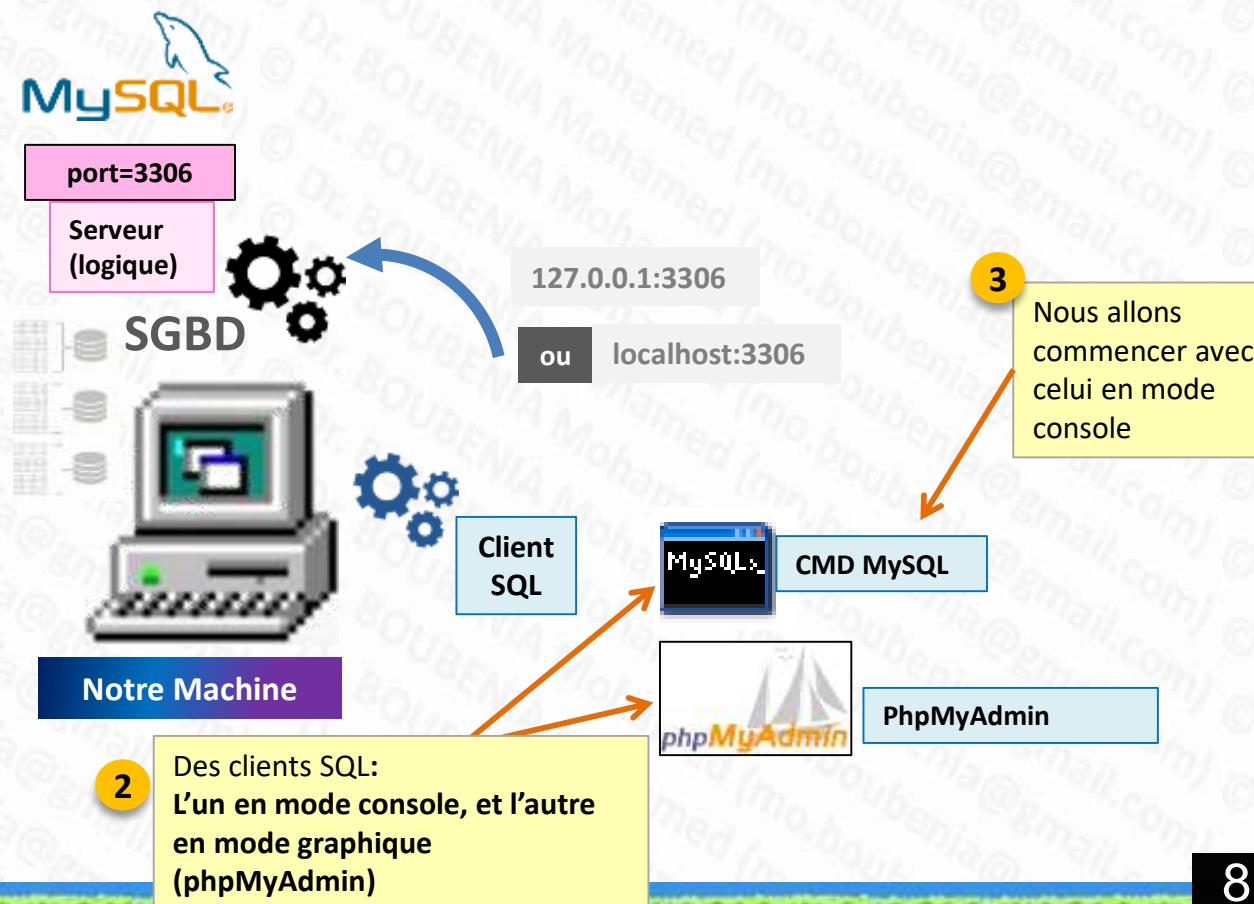
Lancement

Exercice 3 MySQL: Client SQL (cmd): Lancement

1

Rappelez-vous:

Nous avons dit que l'installateur de MySQL vient avec 2 clients SQL



2

Des clients SQL:
L'un en mode console, et l'autre
en mode graphique
(phpMyAdmin)

Exercice 3 MySQL: Client SQL (cmd): Lancement

À vous

1

Pour lancer le client SQL en mode console, vous avez 2 possibilités

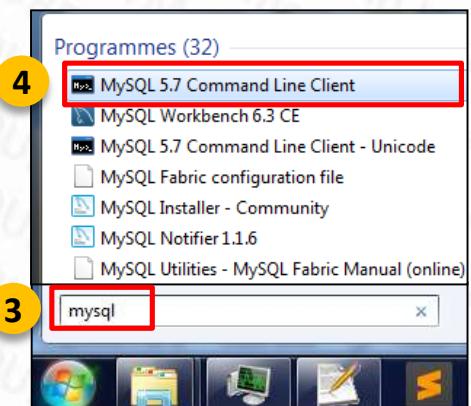
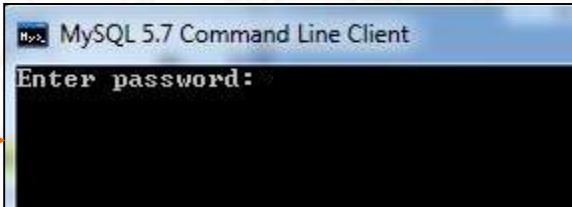
2

La première méthode, qui est la plus simple c'est de retrouver le lanceur "MySQL 5.7 Command Line Client"

5

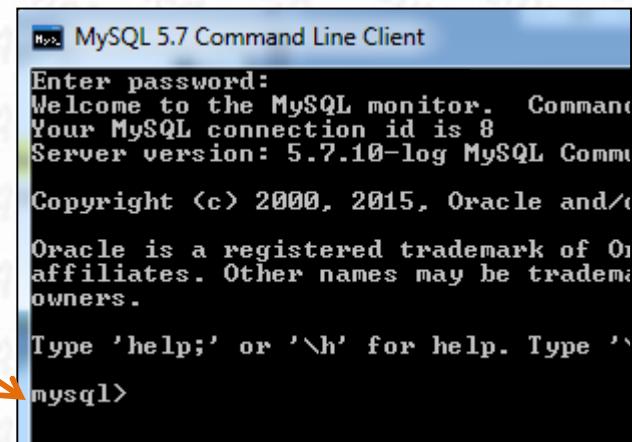
Vous obtiendrez un invite de commande qui vous demande de renseigner le mot de passe de l'utilisateur root.

Si vous avez bien suivi les étapes d'installation, le mot de passe serait initialement `vie`. Il vous suffit alors simplement d'appuyez sur la touche Entrée.



4

Vous obtiendrez ensuite l'invite de commande de MySQL à proprement parler, qui attend des requêtes SQL.



3

Exercice 4

MySQL: Client SQL (cmd)

Création d'une BD

À vous

2

En utilisant le client SQL, vous allez créer la base de données "db_persons", en entrant la requête SQL suivante:

```
CREATE DATABASE db_persons;
```

3

En appuyant sur Entrée, le client SQL vous informe que la requête a bien été exécutée

```
mysql> CREATE DATABASE db_persons;  
Query OK, 1 row affected (0.00 sec)
```

1

Dans ce TP, Vous allez jouer le rôle du DBA



Exercice 5

MySQL: Client SQL (cmd)

**Consultation de
la liste des BD**

Exercice 5 MySQL: Client SQL (cmd): Consultation des BD

À vous

1

Rappelez-vous, dans le cours, nous avons dit qu'un SGBD pouvait gérer plusieurs bases de données.

2

Pour consulter la liste des bases de données déjà créées, entrez la requête SQL suivante:

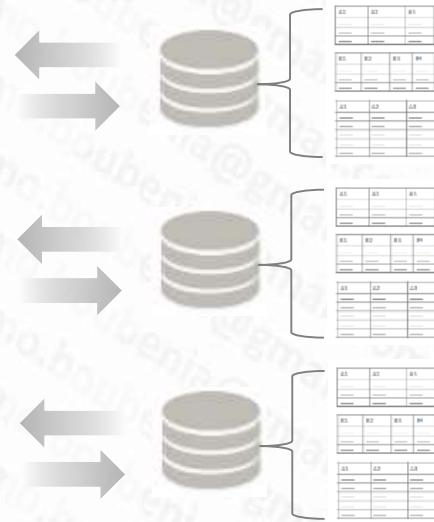
```
SHOW DATABASES;
```

3

Vous devriez voir la liste comme suit.
Remarquez que **db_persons**, celle que nous avons créée, figure dans cette liste.

```
+-----+  
| Database          |  
+-----+  
| information_schema |  
| db_persons        |  
| mysql             |  
| performance_schema |  
| test              |  
+-----+  
5 rows in set (0.00 sec)
```

SGBD



Exercice 6

MySQL: Client SQL (cmd)

Création
d'une Table

Exercice 6 MySQL: Client SQL (cmd): Crédit d'une Table

À vous

1

Avant de pouvoir créer une table dans une base de données. Vous devez d'abord la sélectionner une première fois. Une fois sélectionnée, toutes les manipulations suivantes (création, suppression..etc) se feront sur cette BD.

2

Nous allons sélectionner la BD db_persons, sur laquelle nous créerons ensuite la table person:

```
USE db_persons;
```

3

En utilisant le client SQL, vous allez créer la table "Person", en entrant la requête SQL suivante:

```
CREATE TABLE person (id INT PRIMARY KEY,  
nom VARCHAR(20),  
prenom VARCHAR(20),  
points FLOAT);
```

Créer une table



db_persons

Person

ID	NOM	PRENOM	POINTS
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—

Exercice 7

MySQL: Client SQL (cmd)

**Consultation
des Tables**

Exercice 7 MySQL: Client SQL (cmd): Consultation des Tables

À vous

1

Rappelez-vous, dans le cours, nous avons dit qu'une Base de données pouvait contenir une ou plusieurs tables

2

Pour consulter la liste des tables d'une base de données, commencez d'abord par la sélectionner, comme suit

```
USE db_persons;
```

```
mysql> USE db_persons;
Database changed
mysql> _
```

3

Puis, affichez la liste des tables, comme suit

```
SHOW TABLES;
```

4

Vous devriez voir la liste comme suit. Remarquez que pour l'instant, elle ne contient que la seule table que nous avions créée, à savoir la table PERSON.

```
mysql> SHOW TABLES;
+-----+
| Tables_in_db_persons |
+-----+
| person                |
+-----+
1 row in set (0.00 sec)

mysql>
```



Person			
ID	NOM	PRENOM	POINTS
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—

db_persons

5

Exercez-vous à afficher les tables d'une autre bases de données. Par exemple la base de données mysql.

```
+-----+
| Tables_in_mysql |
+-----+
| columns_priv   |
| db             |
| event          |
| func           |
| general_log    |
| help_category  |
| help_keyword   |
| help_relation  |
| help_topic     |
+-----+
```

Exercice 8

MySQL: Client SQL (cmd)

**Lire le contenu
d'une Table**

1

En utilisant la requête suivante, lisez le contenu de la table **PARTITIONS**, de la base de données **INFORMATION_SCHEMA**.

2

Vous remarquerez qu'il y a beaucoup de lignes et de colonnes.

```
Monter MySQL 5.7 Command Line Client
+-----+-----+-----+-----+
| 1888 | NULL | 0 | 0 | NULL |
+-----+-----+-----+-----+
| def | performance_schema | events_stages_history_long | NULL | |
| NULL | NULL | 0 | 0 | NULL |
| 18888 | NULL | 0 | 0 | NULL |
+-----+-----+-----+-----+
| def | performance_schema | events_stages_summary_by_account | NULL | |
| NULL | NULL | 0 | 0 | NULL |
| 1888 | NULL | 0 | 0 | NULL |
+-----+-----+-----+-----+
| def | performance_schema | events_stages_summary_by_host_by | NULL |
| NULL | NULL | 0 | 0 | NULL |
+-----+-----+-----+-----+
```

4

Affichez que la colonne name

**SELECT table_name
FROM partitions LIMIT 5;**

SELECT * FROM PARTITIONS;

3

Limitons le nombre colonnes, et de lignes à afficher avec la clause **LIMIT <nombre de ligne>**.

Par exemple, affichez simplement que les 5 premières lignes

**SELECT table_catalog, table_name, avg_row_length
FROM partitions LIMIT 5;**

```
mysql> select table_catalog, table_name, avg_row_length from PARTITIONS LIMIT 5;
+-----+-----+-----+
| table_catalog | table_name | avg_row_length |
+-----+-----+-----+
| def | CHARACTER_SETS | 384 |
| def | COLLATIONS | 231 |
| def | COLLATION_CHARACTER_SET_APPLICABILITY | 195 |
| def | COLUMNS | 0 |
| def | COLUMN_PRIVILEGES | 2565 |
+-----+-----+-----+
5 rows in set (0.05 sec)
```

```
mysql> select table_name from PARTITIONS LIMIT 5;
+-----+
| table_name |
+-----+
| CHARACTER_SETS |
| COLLATIONS |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS |
| COLUMN_PRIVILEGES |
+-----+
```

Exercice 9

MySQL: Client SQL (cmd)

Afficher la structure
d'une Table

Exercice 9 MySQL: Client SQL: Afficher la structure d'une Table

À vous

2

Pour consulter le schéma (ou la structure) d'une table existante, il faut exécuter la requête suivante (en sélectionnant d'abord la base de données):

```
DESCRIBE person;
```

```
mysql> use db_persons;
Database changed
mysql> DESCRIBE person;
+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+
| id   | int<11> | NO  | PRI  | NULL    |       |
| nom  | varchar<20> | YES |       | NULL    |       |
| prenom | varchar<20> | YES |       | NULL    |       |
| points | float  | YES |       | NULL    |       |
+-----+-----+-----+-----+
```

Person

ID	NOM	PRENOM	POINTS

1

Rappelez-vous, dans le cours, nous avons dit que chaque table a un **schéma** (par exemple ici:
Nom table: PERSON

Les Colonnes:

*Nom (Chaine de caractère),
 Prenom (Chaine de caractère),
 Points (Entier),*

Les lignes insérées doivent respecter ces colonnes.

3

Exercez-vous à afficher la structure d'autres tables. Par exemple la table **PARTITIONS** de la base de données **INFORMATION_SCHEMA**.

Field	Type	Null	Key	Default	Extra
TABLE_CATALOG	varchar<512>	NO			
TABLE_SCHEMA	varchar<64>	NO			
TABLE_NAME	varchar<64>	NO			
PARTITION_NAME	varchar<64>	YES		NULL	
SUBPARTITION_NAME	varchar<64>	YES		NULL	
PARTITION_ORDINAL_POSITION	bigint<21> unsigned	YES		NULL	
SUBPARTITION_ORDINAL_POSITION	bigint<21> unsigned	YES		NULL	
PARTITION_METHOD	varchar<18>	YES		NULL	
SUBPARTITION_METHOD	varchar<12>	YES		NULL	
PARTITION_EXPRESSION	longtext	YES		NULL	

!

Connaitre le schéma d'une table vous aidera à lire/insérer/modifier les données correctement.

Exercice 10

MySQL: Client SQL (cmd)

**CRUD sur
une Table**

Exercice 10 MySQL: Client SQL (cmd): CRUD sur une Table

Exécutez chacune des requête SQL suivante, et affichez le contenu de la table après chaque requête, afin de voir les effets.

```
/* Insertion de 2 lignes */
INSERT INTO person
    VALUES(1, "JOHN", "DOE", 15);

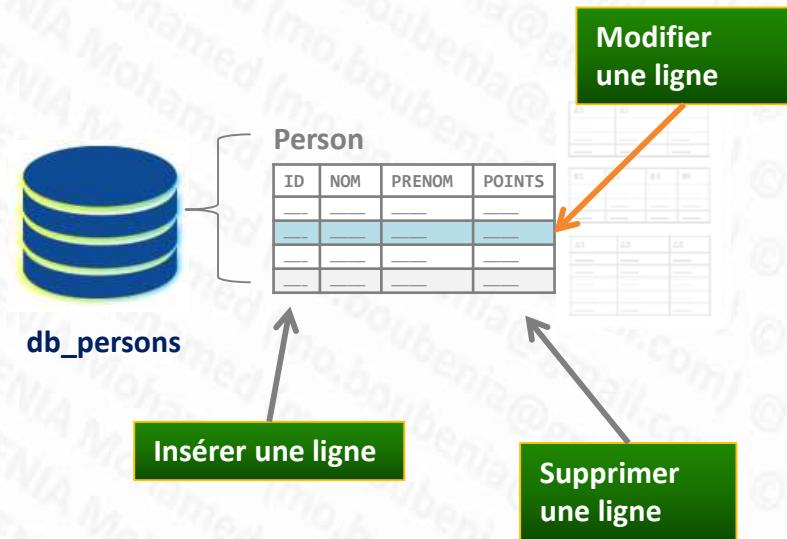
INSERT INTO person
    VALUES(2, "BOB", "CARLTON", 9);

/* Selection de toutes colonnes */
SELECT * FROM person;

/* Selection de 2 colonnes */
SELECT NOM, PRENOM FROM person;

/* Suppression de ligne qui a id=2 */
DELETE FROM person WHERE id = 2;

/* Modification de la colonne points
pour la ligne de ligne qui a id=1 */
UPDATE person SET points=18 WHERE id=1;
```



Exercice 11

MySQL: Client SQL (Graphique)

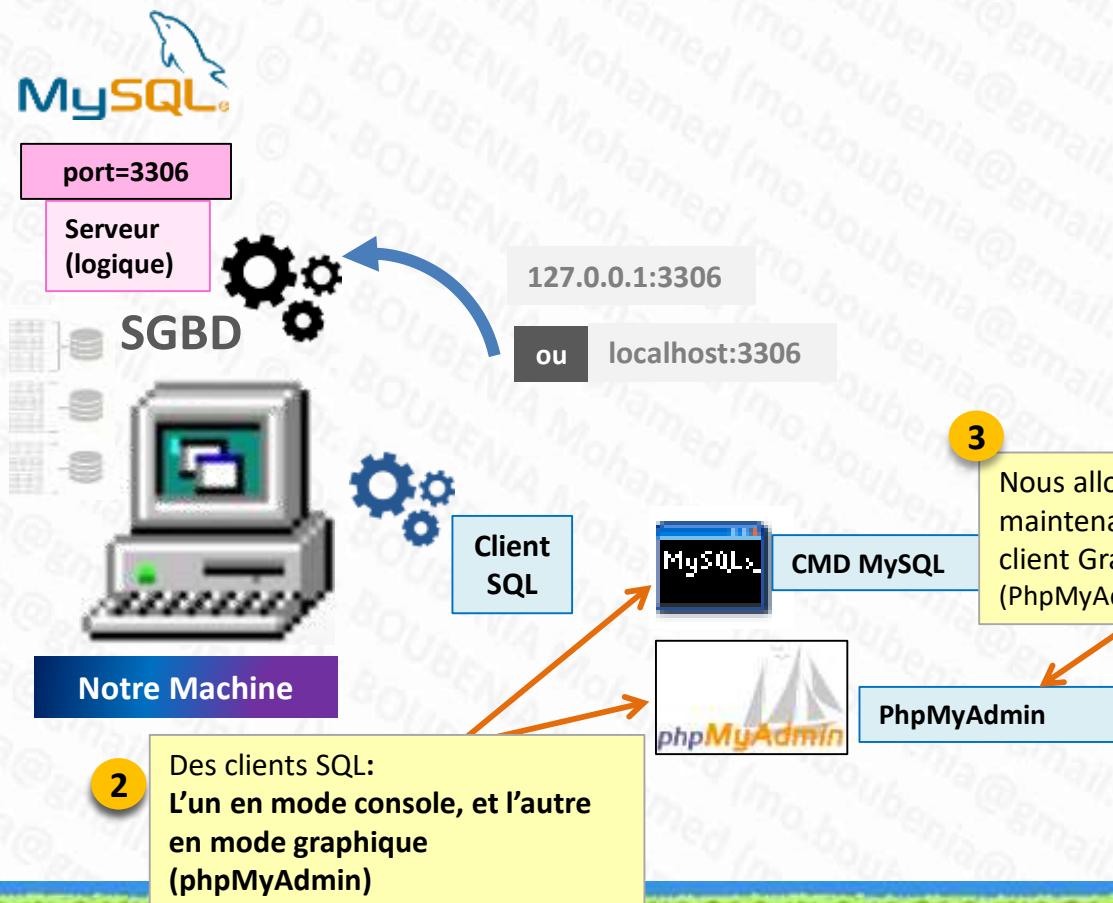
Lancement

Exercice 11 MySQL: Client SQL (Graphique): Lancement

1

Rappelez-vous:

Nous avons dit que l'installateur de MySQL vient avec 2 clients SQL

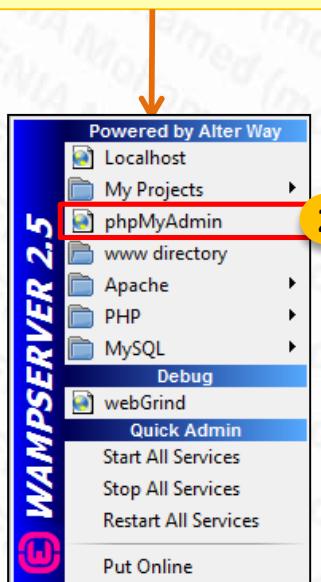


Exercice 11 MySQL: Client SQL (Graphique): Lancement

À vous

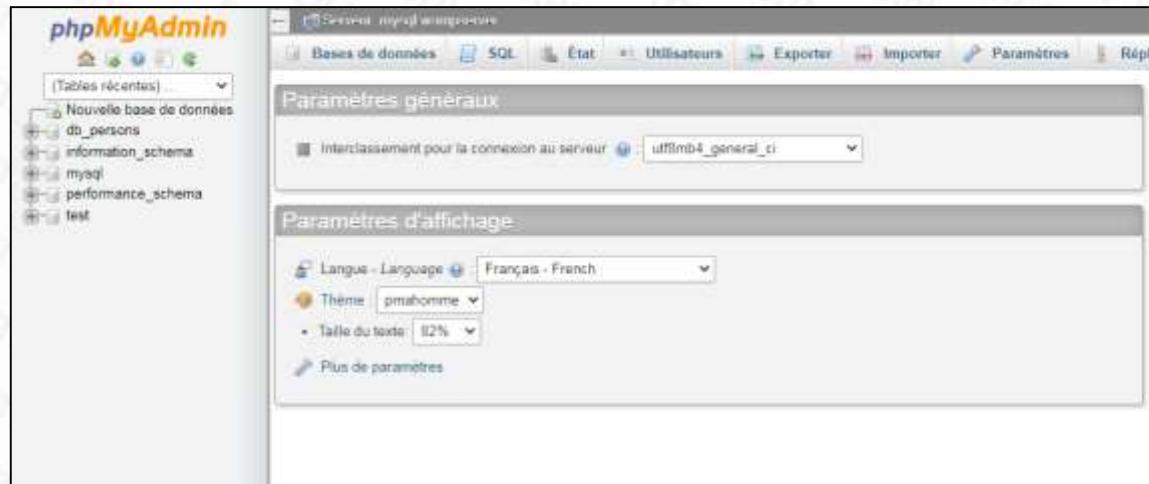
1

Pour lancer le client SQL Graphique (PhpMyAdmin), Utilisez le menu contextuel de Wamp comme suit:



3

Vous devriez voir une page web s'ouvrir sur le navigateur

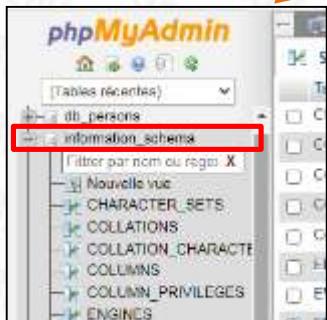
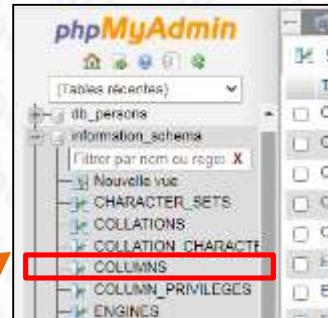


Exercice 11 MySQL: Client SQL (Graphique): Lancement

À vous

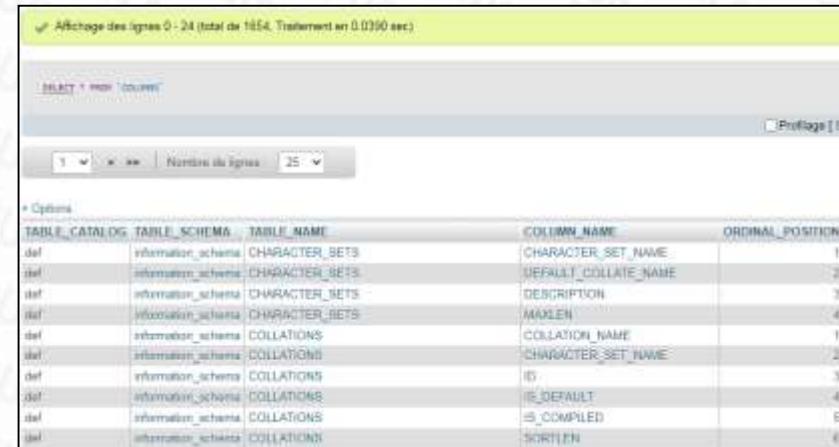
1

Dépliez le contenu de la base de données **INFORAMTION_SCHEMA**, en cliquant dessus. Vous devriez voir les tables de cette base de données

2

Faites clic droit sur une table, pour vous permettre de voir le contenu de la table



TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION
def	information_schema	CHARACTER_SETS	CHARACTER_SET_NAME	1
def	information_schema	CHARACTER_SETS	DEFAULT_COLLATE_NAME	2
def	information_schema	CHARACTER_SETS	DESCRIPTION	3
def	information_schema	CHARACTER_SETS	HASCLEN	4
def	information_schema	COLLATIONS	COLLATION_NAME	1
def	information_schema	COLLATIONS	CHARACTER_SET_NAME	2
def	information_schema	COLLATIONS	ID	3
def	information_schema	COLLATIONS	IS_DEFAULT	4
def	information_schema	COLLATIONS	IS_COMPILED	5
def	information_schema	COLLATIONS	SORTLEN	6

3

Vous devriez voir au milieu les données, comme suit:

```
mysql> SELECT table_catalog, table_schema, table_name, column_name, ordinal_position FROM COLUMNS_LIN  
IT 25;  
  
+-----+-----+-----+-----+  
| table_catalog | table_schema | table_name | column_name |  
| ordinal_position |  
+-----+-----+-----+-----+  
| def | information_schema | CHARACTER_SETS | CHARACTER_SET_NAME | 1  
| def | information_schema | CHARACTER_SETS | DEFAULT_COLLATE_NAME | 2  
| def | information_schema | CHARACTER_SETS | DESCRIPTION | 3  
| def | information_schema | CHARACTER_SETS | MAXLEN | 4  
| def | information_schema | COLLATIONS | COLLATION_NAME | 1  
+-----+-----+-----+-----+
```

VS

Affichage des lignes 0 - 24 (Total de 1654. Traitement en 0.0390 sec)			
SELECT * FROM COLUMNS_LIN			
TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME
def	information_schema	CHARACTER_SETS	CHARACTER_SET_NAME
def	information_schema	CHARACTER_SETS	DEFAULT_COLLATE_NAME
def	information_schema	CHARACTER_SETS	DESCRIPTION
def	information_schema	CHARACTER_SETS	MAXLEN
def	information_schema	COLLATIONS	COLLATION_NAME
def	information_schema	COLLATIONS	ID
def	information_schema	COLLATIONS	IS_DEFAULT
def	information_schema	COLLATIONS	IS_COMPILED
def	information_schema	COLLATIONS	SORTLEN

En générale, le client SQL en mode console est plus rapide en termes d'accès, et en termes de réponse. De plus, il est toujours accessible dans un environnement minimaliste (par exemple un serveur sans interface Windows, disposant seulement de console: Exemple: Unix)

Le client SQL graphique est adapté pour les utilisateurs qui ne sont pas familiers avec le mode "Console", et peut aider pour mieux visualiser le contenu des colonnes trop large.

Exercice 12

Sélection

À vous

Dans un dossier Exo11, créez une page Web PHP, persons.php, qui affiche le contenu de la table **PERSON**.

Result:

1	JOHN	DOE	18.0
2	BOB	CARLTON	9.0
3	RAYANE	SMITH	13.0
4	ROSIE	WOOD	14.0
5	JOHN	DOE	15.0
6	nom6	prenom6	66.0
7	nom7	prenom7	77.0
8	BOB	prenom1	99.0
9	nom9	prenom9	99.0
10	BOB	prenom1	999.0
11	BOB	prenom1	999.0



Veuillez à remplir le contenu de la table **PERSON** avec un client SQL, afin d'avoir un tableau HTML bien rempli.

Exercice 13

Formulaire

Insertion

Exercice 13 Formulaire: Insertion

À vous

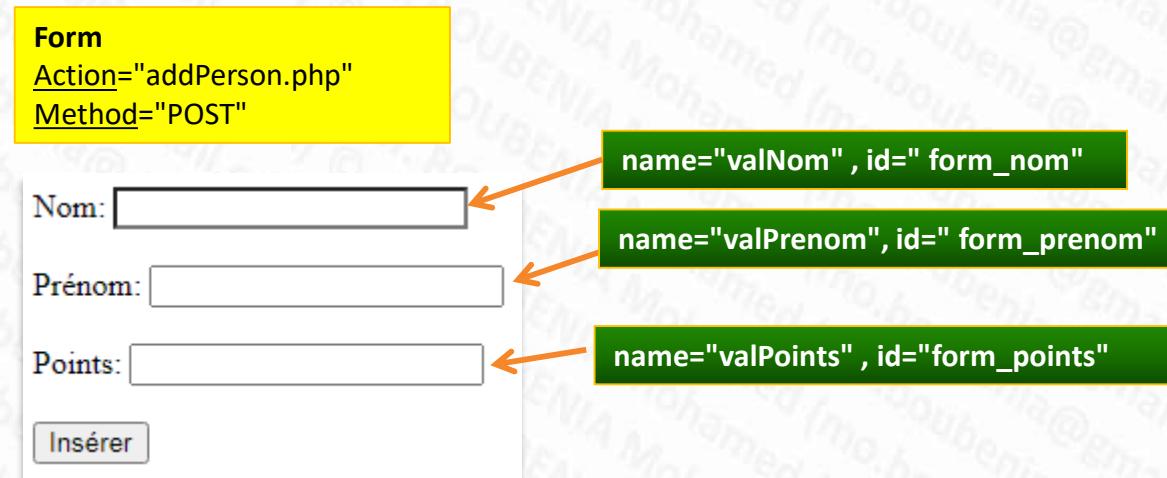
Dans un dossier Exo11, créez une page Web PHP, formAdd.php, qui affiche le formulaire suivant. (ne mettez ni CSS, ni JavaScript)

Form
Action="addPerson.php"
Method="POST"

Nom:

Prénom:

Points:



À vous

1

Dans "addPerson.php", implémenter le code PHP qui permet de récupérer les données reçues du formulaire "formAdd.php", et de les insérer dans la table **PERSON** de la base de données, **db_persons**

2

En utilisant un client SQL, assurez-vous que l'insertion s'est bien déroulée.

3

Ensuite, ajoutez une redirection vers PERSONS.php

Exercice 13 Formulaire: Insertion

À vous

1

Testez sur l'interface

Nom:

Prénom:

Points:

2

Utilisez le client SQL (CMD) pour vérifier que l'insertion s'est bien déroulée

9	nom9	prenom9	99
10	BOB	prenom1	999
11	BOB	prenom1	999
12	ROSIE	WOOD	13

3

Ne vous étonnez pas de retrouver un id différent, tout dépendra de l'état de votre table Person

Exercice 14

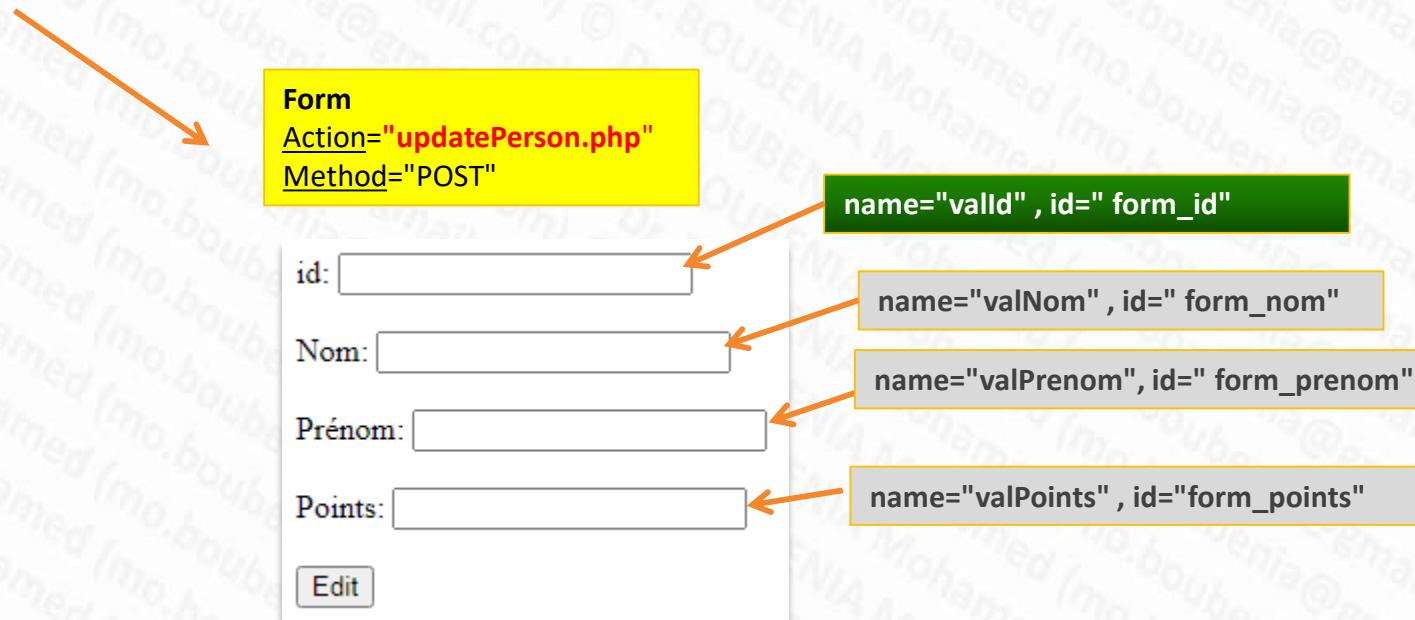
Formulaire

Mise à jour

Exercice 14 Formulaire: Mise à Jour

À vous

Dans un dossier Exo11, créez une page Web PHP, formUpdate.php, qui affiche le formulaire suivant. (ne mettez ni CSS, ni JavaScript)



The diagram illustrates a form update interface. At the top left, a yellow box contains the form definition:

```
Form  
Action="updatePerson.php"  
Method="POST"
```

Below the form definition, there are four input fields: "id:", "Nom:", "Prénom:", and "Points:". To the right of each input field is a green box containing validation logic:

- "id:" is associated with a green box labeled `name="valid" , id=" form_id"`.
- "Nom:" is associated with a green box labeled `name="valNom" , id=" form_nom"`.
- "Prénom:" is associated with a green box labeled `name="valPrenom" , id=" form_prenom"`.
- "Points:" is associated with a green box labeled `name="valPoints" , id="form_points"`.

Each validation box has an orange arrow pointing from it to its corresponding input field.

À vous

1

Dans "updatePerson.php", implémenter le code PHP qui permet de récupérer les données reçues du formulaire "formUpdate.php", et de mettre à jour les données de **PERSON** de la base de données, **db_persons**

2

En utilisant un client SQL, assurez-vous que la mise à jour s'est bien déroulée.

3

Ensuite, ajoutez une redirection vers PERSONS.php

Exercice 14 Formulaire: Mise à Jour

À vous

1

Lancez le serveur, et testez sur l'interface (en précisant le id de la personne que vous voulez modifier)

id:

Nom:

Prénom:

Points:

Edit

2

Utilisez le client SQL (CMD) pour vérifier que la mise à jour s'est bien déroulée



9	nom9	prenom9	99
10	BOB	prenom1	999
11	BOB	prenom1	999
12	ROSIE-12	WOOD-12	133

Exercice 15

Formulaire

Suppression

Exercice 15 Formulaire: Suppression

À vous

Copiez le contenu de Exo13 dans un autre dossier Exo14, puis rajoutez la route "/formDelete" qui affiche le formulaire suivant (ne mettez ni CSS, ni JavaScript)

Form
Action="deletePerson.php"
Method="POST"

id:

Supprimer

name="valid" , id=" form_id"

Exercice 15 Formulaire: Suppression

À vous

1

Lancez le serveur, et testez sur l'interface (en précisant le id de la personne que vous voulez supprimer)

id:

2

Utilisez le client SQL (CMD) pour vérifier que la ligne concernée a bien été supprimée

9	nom9	prenom9	99
10	BOB	prenom1	999
11	BOB	prenom1	999

Exercice 16

Mini-App

Exercice 16 Mini-App

À vous

On voudrait maintenant connecter le tout dans une seule application, en repensant l'application développée dans le TP6, comme ceci:

N°	Nom	Prénom	Points
1	nom1	prenom1	5
2	nom2	prenom2	10
3	nom3	prenom3	15

3 ligne(s) Total point(s)= 30

[Console](#) [Tableau](#) [Supprimer](#) [Modifier](#) [Ajouter](#)

Exercice 16 Mini-App

À vous

On voudrait maintenant connecter le tout dans une seule application, en repensant l'application développée dans le TP6, comme ceci:

