

Découverte des bases de données

Notions de base qui seront abordés au cours du premier semestre :

- qu'est-ce qu'une base de données, un SGBD;
- comment créer une base;
- comment y stocker des données;
- comment interroger une base de données avec SQL,
- etc....

Généralités sur les bases de données

Définitions

- Une base de données est un ensemble structuré de données enregistrées dans un ordinateur et accessibles de façon sélective
- Un logiciel qui permet d'interagir avec une base de données s'appelle un système de gestion de base de données (SGBD)

Fonctions d'un SGBD

On fait toujours appel à des logiciels spécialisés pour accéder à une BD: Les Systèmes de Gestion de Bases de Données (SGBD), ils se chargent de :

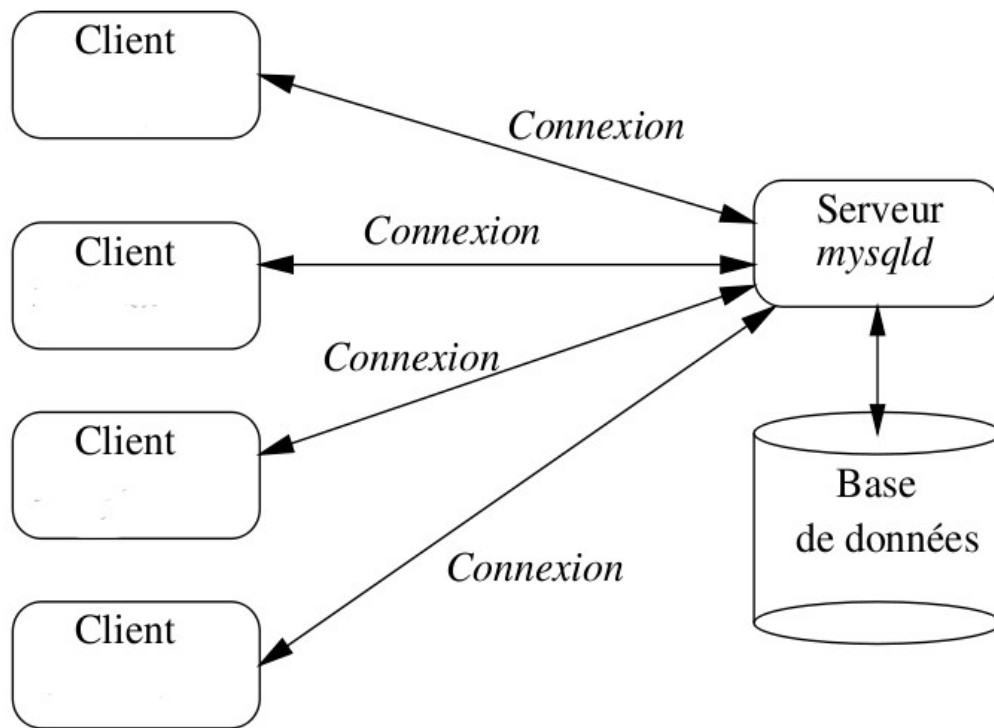
- Décrire les données qui seront stockées
- Manipuler ces données (ajouter, modifier, supprimer des informations)
- Consulter les données et traiter les informations obtenues (sélectionner, trier, calculer, ...)
- Définir des protections d'accès (mots de passe, autorisations,...)
- Résoudre les problèmes d'accès multiples aux données
- Prévoir des procédures de reprise en cas d'incident (sauvegardes, journaux,...)

Exemples de SGBD: ORACLE, MySQL, PostgreSQL, ...

Présentation de MySQL

La base de données MySQL® est devenue la base de données open source la plus populaire au monde grâce à sa haute performance, sa fiabilité et sa simplicité d'utilisation. Il existe actuellement une très forte demande de développeur d'application connaissant PHP-MYSQL, aussi au cours de ce semestre nous travaillerons avec ce SGBD.

MYSQL fonctionne en client/serveur: le serveur gère la base, les clients communiquent avec le serveur.



En pratique le client ne se soucie pas de ce que fait le serveur. Il doit :

- choisir une application cliente qui permet de dialoguer avec un serveur MySQL.
- se connecter en désignant le serveur, et en donnant un login et un mot de passe;
- envoyer des instructions.

LE SGBD : MYSQL

Les TABLES

MySQL nous présente les données sous forme de table.

Exemple : la Table Film (les succès du Box Office source Wikipédia)

idfilm	titre	realisateur	genre	année	recettes (millions \$)
1	Twilight chapitre II : Tentation	Chris Weitz	Fantastique	2009	710
2	Transformers	Michael Bay	Fantastique	2007	709
3	Twilight chapitre III : Hésitation	David Slade	Fantastique	2010	698
4	Forrest Gump	Robert Zemeckis	Histoire	1994	677
5	Avatar	James Cameron	Fantastique	2009	2782
6	Titanic	James Cameron	Drame	1997	1843
7	Harry Potter et les Reliques de la Mort II	David Yates	Fantastique	2011	1294
8	Le Seigneur des anneaux : Le Retour du roi	Peter Jackson	Histoire	2003	1119
9	Sixième Sens	M. Night Shyamalan	Science fiction	1999	673
10	L Âge de glace 2	Carlos Saldanha	Animation	2006	655
11	Pirates des Caraïbes : La Malédiction du Black Pearl	Gore Verbinski	Fantastique	2003	654

Une base de données est un ensemble de tables.

- chaque table a un nom unique (par exemple Film);
- une table comprend une ou plusieurs colonnes ayant chacune un nom;
- une table comprend une ou plusieurs lignes, chacune constituée d'une valeur pour chaque colonne.
- les noms de table et de colonne constituent le schéma de la base; les lignes constituent le contenu de la base.

ACCÉDER AUX DONNÉES DÉJÀ PRÉSENTES DANS UNE BASE

Pour accéder aux données présentes dans une base, nous allons utiliser le langage SQL. Le langage SQL (*Structured Query Language*) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est aujourd'hui supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel que Access ou par les produits plus professionnels tels que Oracle.

De manière synthétique, on peut dire que SQL est un langage relationnel, il manipule donc des tables par l'intermédiaire de requêtes qui produisent également des tables.

SELECT : sélection de tout ou d'une partie de la table

syntaxe : SELECT "nom de colonne" FROM "nom de table"

Pour sélectionner le titre des films

SELECT titre **FROM** Film ;

Pour sélectionner le titre et l'année des films

SELECT titre, annee **FROM** Film ;

Pour sélectionner toutes les colonnes de la table Film

SELECT * FROM Film ;

Pour sélectionner les réalisateurs

SELECT realisateur **FROM** Film ;

Le résultat de cette requête SQL fait apparaître plusieurs fois certains réalisateurs (Cameron, Lucas ...), si on vous demande d'afficher uniquement les différents réalisateurs, vous devez utiliser le mot clé **DISTINCT**.

```
SELECT DISTINCT realisateur FROM Film ;
```

SQL WHERE : par exemple pour sélectionner tous les films dont les recettes sont supérieures à 1000.

```
SELECT titre,recettes  
FROM Film  
WHERE recettes > 1000 ;
```

pour sélectionner tous les films de George Lucas.

```
SELECT titre  
FROM Film  
WHERE realisateur = 'George Lucas';
```

SQL AND OR : par exemple pour sélectionner tous les films dont les recettes sont supérieures à 1000 et inférieures ou égale à 1500.

```
SELECT titre,recettes  
FROM Film  
WHERE recettes > 1000 and recettes <= 1500;
```

SQL BETWEEN : les films entre 2000 et 2005

```
SELECT titre,annee  
FROM Film  
WHERE annee BETWEEN 2000 AND 2005 ;
```

SQL LIKE : souvent utilisé avec des caractères de remplacement, voici quelques exemples :

- 'A_Z' : toutes les chaînes commençant par 'A', ou un autre caractère, et terminant par 'Z'. Par exemple, 'ABZ' et 'A2Z' satisferaient la condition, alors 'AKKZ' ne le ferait pas (car il y a deux caractères entre A et Z au lieu d'un).
- 'ABC%' : toutes les chaînes commençant par 'ABC'. Par exemple, 'ABCD' et 'ABCABC' satisferaient la condition.
- '%XYZ' : toutes les chaînes terminant par 'XYZ'. Par exemple, 'WXYZ' et 'ZZXYZ' satisferaient la condition.
- '%AN%' : toutes les chaînes contenant le modèle 'AN' quelle que soit sa position. Par exemple, 'LOS ANGELES' et 'SAN FRANCISCO' satisferaient la condition.

Par exemple pour sélectionner tous les films de Harry Potter

```
SELECT titre  
FROM Film  
WHERE titre LIKE 'Harry Potter%';
```

SQL ORDER BY :

```
SELECT "nom de colonne"  
FROM "nom de table"  
[WHERE "condition"]  
ORDER BY "nom de colonne" [ASC, DESC]
```

Les crochets [] signifient que l'instruction **WHERE** est optionnelle. Toutefois, si une clause

WHERE est déjà présente, elle se trouvera devant la clause **ORDER BY**. La clause **ASC** ou **DESC** provoque respectivement un tri des résultats dans un ordre ascendant ou descendant. En cas d'omission de l'une d'elles, la clause **ASC** est implicitement appelée, provoquant donc un tri en ordre ascendant par défaut.

Par exemple pour afficher le titre des films par ordre décroissant des recettes :

```
SELECT titre,recettes  
FROM Film  
ORDER BY recettes DESC;
```

TRAVAIL À FAIRE :

Rédiger en langage SQL les requêtes suivantes :

Quels sont les films fantastiques réalisés par Steven Spielberg ?

Afficher le titre des films d'animation par ordre décroissant des recettes.

Afficher tous les films fantastiques réalisés entre 2000 et 2010 .

Afficher tous les films dont le titre commence par « T ».

Afficher tous les films dont le titre comporte le mot « Monde ».

Afficher les différentes catégories.

Plus durs , mais cherchez un peu, regarder du coté des GROUP BY , COUNT : (vous pouvez vous aider d'internet)

Affichez pour chaque réalisateur le nombre de ses films présents dans la base

Affichez le nombre de film pour chaque catégorie

Affichez pour chaque année le film qui a réalisé le plus de recettes

Affichez la moyenne des recettes

Affichez la recette maximale

Affichez la recette minimale

Trouver 3 autres requêtes que l'on pourrait réaliser sur cette table.

Complément prise en main de phpMyAdmin avec Laragon

Par défaut Laragon utilise l'outil HeidiSQL pour gérer vos bases de données, nous devons installer phpMyAdmin qui est une [application Web](#) de gestion pour les [systèmes de gestion de base de données MySQL](#) réalisée principalement en [PHP](#) et distribuée sous licence [GNU GPL](#). C'est le gestionnaire de base de données le plus utilisé.

Pour installer phpMyAdmin dans Laragon suivez les étapes suivantes :

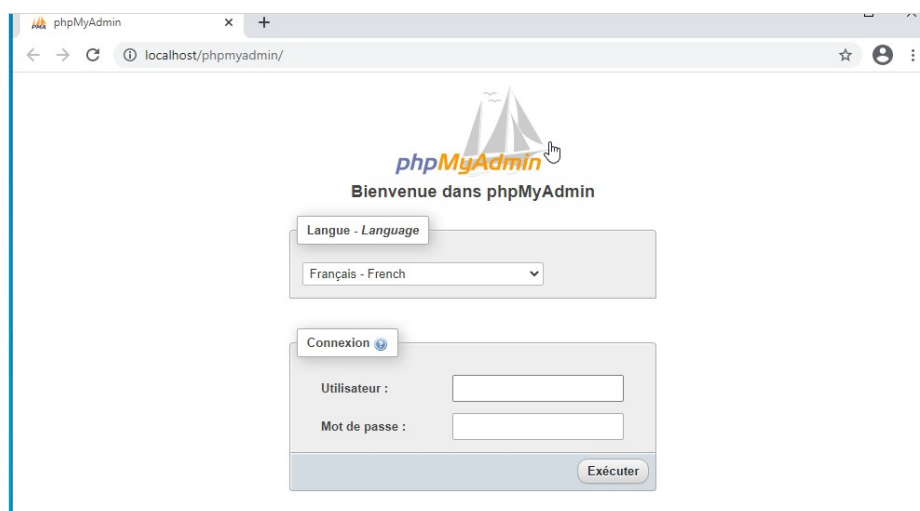
1. Télécharger la dernière version phpMyAdmin zip from [here](#)
2. Extraire votre zip dans : path/to/laragon/etc/apps Votre répertoire doit impérativement se nommer **phpMyAdmin (extraire votre zip dans le répertoire laragon/etc/apps et renommer votre répertoire phpMyAdmin-version..... en phpMyAdmin)**
3. Restart Laragon et dans un navigateur écrivez l'URL suivante : **localhost/phpmyadmin**

« laragon » etc » apps

Rechercher dans : apps

	Nom	Modifié le	Type	Taille
	adminer	28/08/2020 03:04	Dossier de fichiers	
	laragon	28/08/2020 03:04	Dossier de fichiers	
	memcached	28/08/2020 03:04	Dossier de fichiers	
	phpMyAdmin	11/01/2021 12:25	Dossier de fichiers	
	phpRedisAdmin	28/08/2020 03:04	Dossier de fichiers	

Pour vous connecter : utilisateur - root sans mot de passe



phpMyAdmin

Bienvenue dans phpMyAdmin

Langue - Language

Français - French

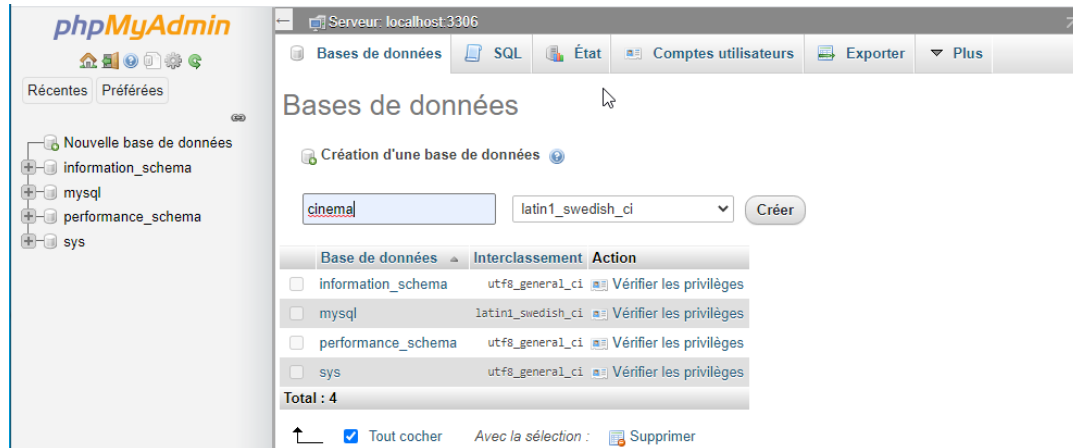
Connexion

Utilisateur :

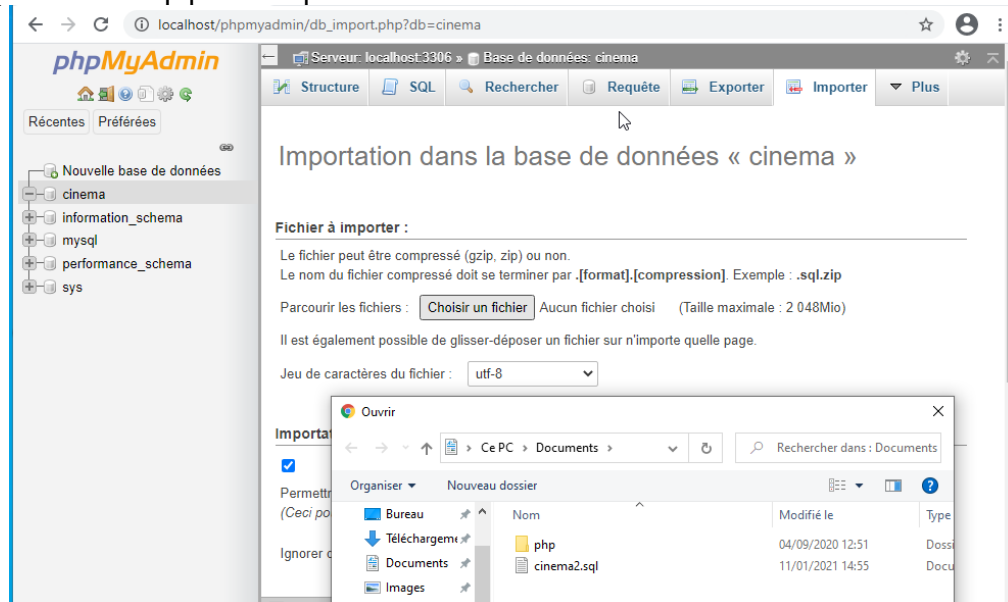
Mot de passe :

Exécuter

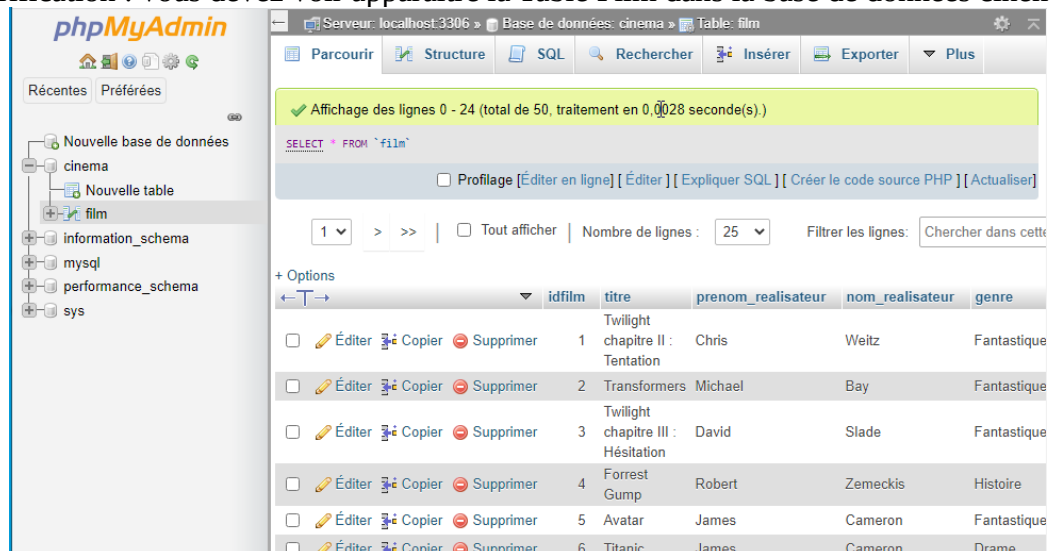
Création de la base cinema :



Exécution d'un script SQL qui va créer la table Film et y insérer des données : télécharger le script cinema2.sql puis l'importer



Vérification : vous devez voir apparaître la Table Film dans la base de données cinema



À partir de l'onglet SQL, vous pouvez commencer à rédiger vos requêtes

