

MySQL, le SQL et les bases de données



Qu'est-ce que c'est une base de données ?

Des tableaux qui ont un lien entre eux.

Exemple :

Films		
titre	annee	genre
Trainspotting	1996	drama
Batman	1989	action
Whiplash	2014	music
Arrival	2016	sci-fi

Acteurs		
ID	nom	prenom
1	Mirren	Helen
2	Asano	Tadanobu
3	Waititi	Taika
4	Lee	Pace

Qu'est-ce que c'est MySQL ?

Un système de gestion de bases de données relationnelles (database management system ou DBMS). C'est ça qui permet de manipuler des bases de données.

Il en existe d'autres comme SQLite, postgresQL, mongoDB.

Qu'est-ce que c'est phpMyAdmin ?

Une application Web pour gérer des bases de données MySQL.

L'application est faite en PHP.

Qu'est-ce que c'est le SQL ?

C'est le langage qui permet d'interagir avec les bases de données.
Dans un premier temps, on va faire du CRUD :

- **C**reate (ajouter une nouvelle entrée dans la DB)
- **R**ead (lire les données)
- **U**ppdate (mettre à jour des données)
- **D**elete (en supprimer)

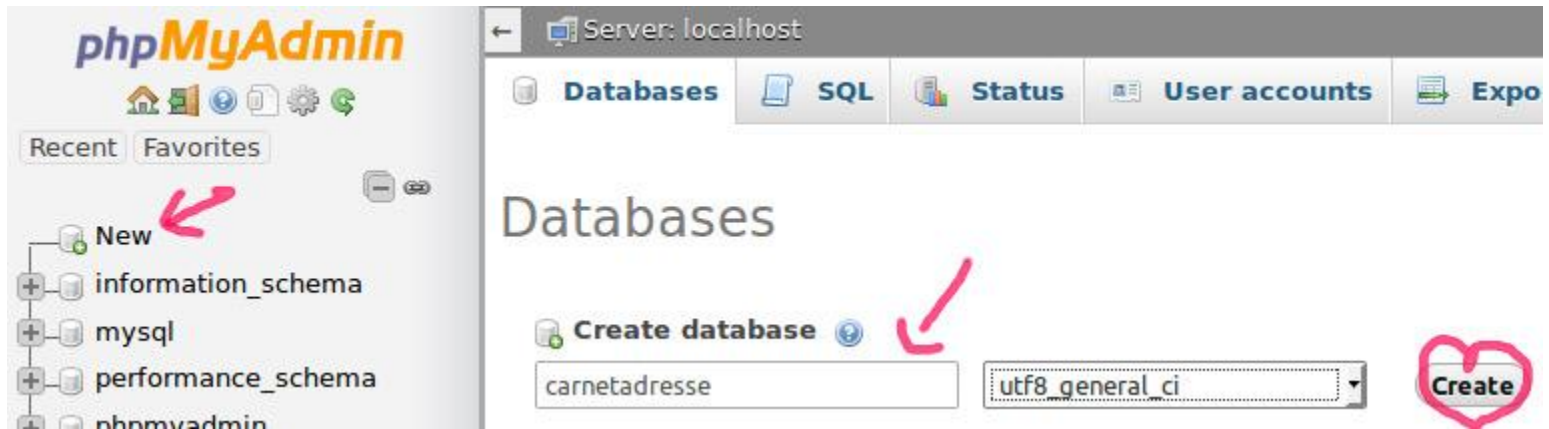
Au boulot !

Installer phpmyadmin

Normalement, vous avez déjà installé [LAMP](#) sur vos machines. Vous avez déjà fait du PHP. Il vous manque une interface pour créer mais surtout pour visualiser puis jouer avec les bases de données.

Un wiki est disponible [sur GitHub à cette adresse](#). Sinon, vous pouvez directement aller sur [la doc officielle](#).

Créer une nouvelle base de donnée



Créer une nouvelle table

On va faire ça en deux temps. D'abord on crée la table en lui donnant un nom et en définissant un certain nombre de colonnes



A screenshot of a 'Create table' dialog box. The dialog has a title bar with a table icon and the text 'Create table'. Below the title bar, there are two input fields. The first is labeled 'Name:' and contains the text 'clients'. The second is labeled 'Number of columns:' and contains the number '6'. The 'Number of columns' field has a small spinner control to its right.


Name:	Number of columns:
clients	6

Créer une nouvelle table

Table name: clients Add 1 column(s) **Go**

id	TINYINT		None		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	identifiant
Pick from Central Columns									
nom	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	personne
Pick from Central Columns									
societe	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Pick from Central Columns									
phone	VARCHAR	15	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Pick from Central Columns									
adresse	VARCHAR	100	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Pick from Central Columns									
email	VARCHAR	100	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Pick from Central Columns									

Créer une nouvelle table

	#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	<u>id</u> 	tinyint(3)		UNSIGNED	No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	<u>nom</u>	varchar(50)	utf8_general_ci		No	None	
<input type="checkbox"/>	3	societe	varchar(50)	utf8_general_ci		No	None	
<input type="checkbox"/>	4	phone	varchar(15)	utf8_general_ci		No	None	
<input type="checkbox"/>	5	adresse	varchar(100)	utf8_general_ci		No	None	
<input type="checkbox"/>	6	email	varchar(100)	utf8_general_ci		No	None	

Qu'est-ce que c'est que ces types ?

Définir le type d'une colonne pour savoir si on va y stocker du texte, des nombres, des dates,...

Numérique	INT, TINYINT, SMALLINT, MEDIUMINT et BIGINT
Alphanumérique	CHAR, VARCHAR, TEXT
Temporel	DATE, DATETIME, TIME, TIMESTAMP et YEAR

Il est important de correctement définir le type d'une colonne. Sinon :

- Gaspillage de mémoire
- Problème de performance (recherche + rapide sur un nombre qu'une chaîne de caractère)
- Impossibilité d'utiliser des fonctionnalités propres à un type de données

Qu'est-ce que c'est que ces attributs ?

Binary	
Unsigned	Ne peut pas être négatif
Unsigned Zerotill	Exemple : SMALLINT(4) UNSIGNED ZEROFILL veut dire que dans la colonne on peut inscrire des nombres à 4 chiffres et ceux qui en ont moins s'inscriront avec des 0. 23 => 0023
On update CURRENT_TIMESTAMP	la date sera automatiquement mise à jour à chaque mise à jour de l'enregistrement

Qu'est-ce que c'est cette clef primaire ?

Il existe deux sortes de clefs :

- **Clef primaire** : sert à identifier une ligne de manière unique
- **Clef étrangère** : sert à gérer les relations entre différentes tables et à garantir la cohérence des données (on verra ça plus tard). Également nommée FK pour Foreign Key.

Jouons un petit jeu...

Créez une DB du nom de **becode**

Ensuite, créez une table du nom de **students**.

Créer les colonnes :

- **idstudent** (clef primaire, auto_increment, non signé)
- **nom**,
- **prenom**,
- **datenaissance**,
- **genre**,
- **ecole**

Dans un premier temps, remplis la table manuellement en ajoutant, modifiant, supprimant des données via phpmyadmin.

ATTENTION : fais en sorte que la colonne **genre** ne contienne que deux choix.
(exemple : F ou M)

Pour la colonne **ecole**, mettez soit Andy, soit Central.

Fais en sorte qu'il y ait au moins 10 personnes dedans.



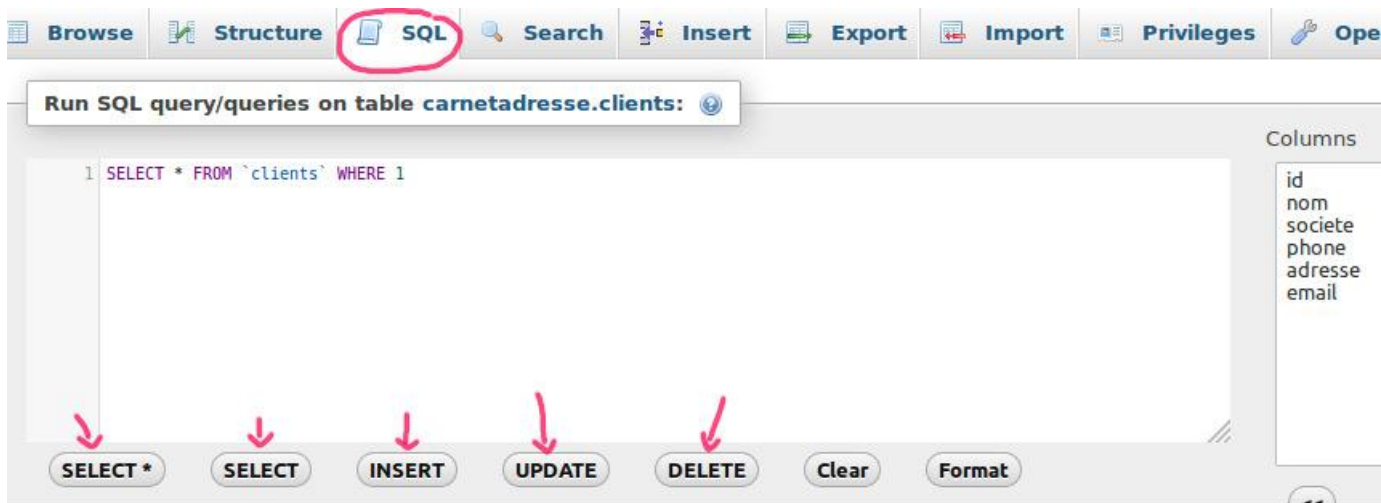
Le SQL

Comment parler ce langage magique ?
Les bases...

Parler le SQL

Les bases

Vous avez de la chance, dans PHPmyAdmin, dans l'onglet SQL, il y a déjà des boutons qui génère la syntaxe de base.



Parler le SQL

select

Structure de base de la requête

```
SELECT  
colonne1, colonne2, colonne3, colonne4  
FROM  
nom_de_la_table  
WHERE  
condition  
;
```

Pour sélectionner toutes les colonnes, on utilise *.

```
SELECT  
*  
FROM  
nom_de_la_table  
;
```

Parler le SQL

Pour les conditions (where)

=	"égal à"
>	"plus grand que"
<	"plus petit que"
>=	"plus grand ou égal à"
<=	"plus petit ou égal à"
<>	"différent de"
LIKE	"qui ressemble à"

Parler le SQL

Pour les conditions (like)

LIKE, associer à %, va permettre de retrouver des résultats dans la DB selon 3 critères :

- "qui **commence** par *chaine_de_caracteres*". (LIKE 'Er%' : tout mot commençant par "Er"),
- "qui **termine** par *chaine_de_caracteres*" (LIKE '%Er' : tout mot terminant par "Er"),
- ou "qui **contient** cette *chaine_de_caracteres*" (LIKE '%Er%' : tout mot contenant "Er").

Parler le SQL

Comment trier

Avec **ORDER BY**.

On écrira :

ORDER BY ASC pour que les données d'une colonne soient triées de A à Z ou du plus petit au plus grand.

ORDER BY DESC pour que les données soient triées de Z à A ou du plus grand au plus petit.

Parler le SQL

Limiter les résultats

Avec **LIMIT**, on limitera le nombre de résultat.

Parler le SQL

insert

Structure de base de la requête

```
INSERT INTO  
nom_de_la_table  
(colonne1, colonne2, colonne3...  
dernière_colonne)  
VALUES  
(value_colonne_1, value_colonne2, ...  
value_dernière_colonne);
```

Exemple :

```
INSERT INTO  
clients  
(nom, societe, phone, adresse, email)  
VALUES  
(  
'Jen barber', 'Reynholm Industries',  
'05 67 89 90 12', 'IT', 'jen@ri.uk');
```

Parler le SQL

update

Structure de base de la requête

ATTENTION : Si on met pas WHERE,
toutes les rangées seront mises à
jour.
#caca

Exemple :

```
UPDATE nom_de_la_table  
SET nom_de_colonne = "nouvelle valeur"  
WHERE nom_de_colonne OPERATEUR "valeur"  
      [and|or nom_de_colonne OPERATEUR  
      "value"];
```

[] = optionnel

```
UPDATE clients  
SET societe = "Restaurattec"  
WHERE societe = "COGIP";
```


Parler le SQL

delete

Structure de base de la requête

ATTENTION : Si on met pas WHERE,
toutes les rangées seront effacées
#caca

```
DELETE from  
nom_de_la_table  
WHERE nom_de_colonne OPERATEUR "valeur"  
[and|or "nom_de_colonne" OPERATEUR  
"valeur"];  
  
[ ] = optionnel
```

Exemple :

```
DELETE FROM  
clients  
WHERE  
societe = "restauratec";
```

Parler le SQL

AS (petite astuce d'alias)

Alias sur une colonne

```
SELECT nom_de_colonne_a_rallonge AS c1  
FROM `table`
```

Alias sur une table

```
SELECT *  
FROM `nom_table_a_rallonge` AS t1
```

Parler le SQL

AS (petite astuce d'alias)

Exemple

```
SELECT f.numero_facture AS nun, p.nom_personne,  
p.prenom_personne AS personne, s.nom_societe AS societe  
  
FROM factures AS f, personnes AS p, societes AS s  
  
WHERE f.fk_societe = s.id_societe AND f.fk_personne =  
p.id_personne
```

Jouons un petit jeu...



Toujours dans phpMyAdmin mais via l'onglet SQL, on va apprendre à afficher, ajouter, modifier et supprimer des données en SQL.

- Reprends la base de données becode.
- Affiche toutes les données.
- Affiche uniquement les prénoms.
- Affiche les prénoms, les dates de naissance et l'école de chacun.
- Affiche uniquement les élèves qui sont de sexe féminin.
- Affiche uniquement les élèves qui font partie de l'école Andy.

- Affiche uniquement les prénoms des étudiants, par ordre inverse à l'alphabet (DESC). Ensuite, la même chose mais en limitant les résultats à 2.

- Ajoute Ginette Dalor, née le 01/01/1930 et affecte-la à Central, toujours en SQL.

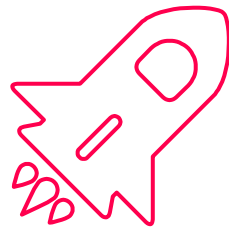
- Modifie Ginette (toujours en SQL) et change son sexe et son prénom en "Omer".

- Supprimer la personne dont l'ID est 3.

- Faire d'autres manipulations pour voir si t'es bien compris.

Ressources

pour commencer ou aller plus loin



- [Petit quizz](#) pour vérifier tes connaissances
- [Se familiariser avec le SQL](#) (Sololearn)
- [S'entraîner au SQL](#) (w3schools)
- [Les types de données](#) (OpenClassrooms)
- [Les bons types de colonnes](#) (OpenClassrooms)