

Structured Query Language

SQL

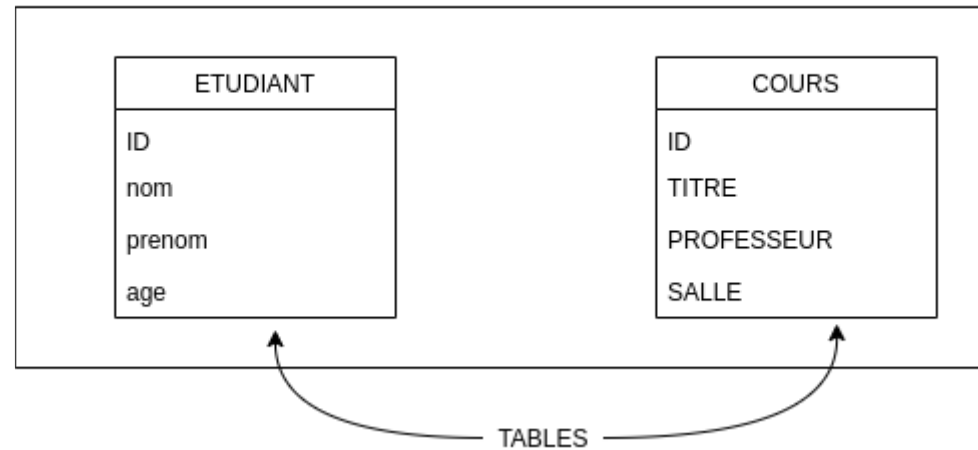
Qu'est-ce que SQL ?

Langage de requête structurée. C'est un langage qui permet d'administrer des bases de données (créer, ajouter, modifier, supprimer ou récupérer des informations).

Comment ça fonctionne ?

Schéma de présentation d'une BDD

BASE DE DONNEE: ECOLE



Comment ça fonctionne ?

Instructions

le langage SQL est basé sur des "instructions", des "mots" qui vont permettre d'effectuer des actions. Une suite d'instruction est nommé requête. exemple: **SELECT, UPDATE, DROP, FROM, ORDER BY**, etc...

ATTENTION ! une requête SQL doit obligatoirement finir par un ";", sinon la requête ne s'effectuera pas

Exemple

imaginons que j'ai une base de donnée et que dans cette bdd, j'ai une table nommé ETUDIANT et que je veux récupérer le nom de tous les étudiants de cette table, en sql on pourrait faire:

```
SELECT nom  
FROM ETUDIANT;
```

3

Quels outils

Pour apprendre et utilisé SQL vous devez choisir un système de gestion de base de données, les plus connus étant:

- MySQL
 - MariaDB
 - PHPMyAdmin (MySQL mais avec interface)
-

3

Notions de base : CREATE, ALTER, DROP

Avant de pouvoir commencer, il faut d'abord créer sa base de donnée et les tables qui la composent. Pour cela ces instructions sont indispensable:

- **CREATE** : créer base de données ou table
 - **ALTER**: modifie une table existante
 - **DROP**: supprime base de données ou table
-

Exemple

```
// Je souhaite créer une base de données ECOLE
// puis créer la table ETUDIANT vu précédemment:

CREATE DATABASE ECOLE; // Création de la bdd ECOLE
USE ECOLE; //permet d'accéder à la bdd ECOLE
CREATE TABLE ETUDIANT; // Création de la table ETUDIANT dans la bdd
```

Exercices

Installer PHPMyAdmin, puis aller dans la partie SQL en haut (l'objectif est de faire un maximum de chose en SQL et ne pas s'aider de l'interface pour la création) et faites en sorte de pouvoir créer une bdd

PREMIEREBDD et créez la table **USER** dans cette bdd. Le **USER** devra contenir un id qui sera sa clé primaire, un nom, un prenom et un age.

Correction

```
CREATE DATABASE PREMIEREBDD;  
CREATE TABLE USER (id INT AUTO_INCREMENT, nom varchar(20) NOT NULL, prenom varchar(20) NOT NULL, age INT NOT NULL, PRIMARY KEY(id));
```

Notions de base : INSERT, UPDATE, DELETE

Après avoir créer notre database et notre première table, il faudrait remplir la table **USER** de nouveaux utilisateurs:

- **INSERT INTO** : ajoute un utilisateur dans une table
 - **UPDATE**: modifie les infos d'utilisateur(s)
 - **DELETE**: supprime l'utilisateur(s)
-

Exemple

```
// Ajout d'un utilisateur nommé Henri dans la table ETUDIANT
INSERT INTO ETUDIANT(nom, prenom, age) VALUES("Roi de France", "Henri II", 474);

// Je souhaite supprimer mon chère Henri
DELETE FROM USER WHERE nom="Henri II";
```

Exercice

Ajouter 5 nouveaux utilisateurs: qui auront respectivement pour prénom: Victor, Elodie, Romain, Guillaume, Emmanuelle. Les autres informations seront à votre choix. Après avoir Créer ces 5 utilisateurs, on s'est rendu compte que Victore s'écrivait avec un e, nous ne voulons pas supprimer l'utilisateur que nous venons de créer. Faites en sortes de corriger l'erreur sur le prénom de Victore.

Base des requêtes

Instructions	Explication
SELECT	Sélectionner les champs à sélectionner(* signifie 'tout')
FROM	Les champs sélectionner précédemment proviennent des tables suivantes
WHERE	Début des conditions supplémentaires

Les autres

Instructions	Explication
AND/OR/NOT	Après le WHERE obligatoirement, autres conditions
GROUP BY	Regrouper les données en fonction de champ
ORDER BY	Trier les résultats en fonction de champs
COUNT/SUM/AVG	Permet de faire des calcul (compter, additionner, moyenne)

Exemples

```
// Nous souhaitons afficher les élèves dont l'âge est supérieur ou égal à 18ans
```

```
SELECT age
FROM ETUDIANT
WHERE age >= 18;
```

```
// Nous souhaitons récupérer toutes les informations
// des cours que le professeur Patrick a effectué
```

```
SELECT *
FROM COURS
WHERE professeur="Patrick";
```

Exercice

Vous êtes développeur dans une mairie de France, votre supérieur vous donne un fichier .sql contenant les informations sur toutes les villes de France. Vous avez plusieurs informations à récupérer:

- Toutes les informations de la ville de Paris
 - Récupérer les nom et départements de toute les villes se trouvant dans le 77 groupé par nom de ville
 - Afficher les villes qui ont plus de 25 000 habitants en 2012 ordonné par nombre d'habitants
-



Correction

```
//1
SELECT * FROM villes_france_free WHERE ville_nom="Paris";
//2
SELECT ville_nom, ville_departement
FROM villes_france_free
WHERE ville_departement=77 GROUP BY ville_nom;
//3
SELECT ville_nom, ville_population_2012
FROM villes_france_free
WHERE ville_population_2012>250000 ORDER BY ville_population_2012;
```



Création et gestion d'utilisateur

MySQL, MariaDB

```
// Permet de créer un nouvel utilisateur
CREATE USER 'test'@'localhost' IDENTIFIED BY 'newpassword';
// Permission d'utiliser les SELECT sur toutes les databases
GRANT SELECT ON * . * TO 'test'@'localhost';
// Permissions pour tout
GRANT ALL PRIVILEGES ON *.* TO 'test'@'localhost';
// Supprimer tout les privileges pour la database ECOLE
REVOKE ALL PRIVILEGES ON ECOLE TO 'test'@'localhost';
// Supprimer l'utilisateur
DROP USER 'test'@'localhost'
```

Liens entre les tables

Exercice

Durant toute votre enfance vous aviez une passion pour la chaîne de télévision Nickelodeon, et cela persiste encore aujourd'hui, vous avez retrouvé une vieille base de données que vous aviez fait il y a 5 ans sur l'univers de Bob l'éponge. Vous décidez de voir si tous ce que vous aviez à l'époque, fonctionne encore aujourd'hui.

- afficher le nom, prénom de l'utilisateur et le modèle, dont la marque de voiture est "burger"
- afficher le nom et prénom des utilisateurs qui pratique le sport "musculatation"

- créer un nouveau USER nommé garry l'escargot, et faite en sorte qu'il participe au sport "chasse à la méduse"
-



Exercice

Dans cet exercice, le but est de créer la base de données avec les informations ci-après. Sujet: En plus d'être développeur, vous êtes passionnés par le monde de la musique et voulez répertorier tous ce qui tournent autour des musiciens. Vous créez une table musicien contenant un id, un nom et un nom d'artiste. Le Musicien peut créer un ou plusieurs album, à l'inverse, un album de peut être créer que par un ou plusieurs musiciens (ex: les enfoirés). Un album est constitué d'une ou plusieurs chanson. Une chanson possède un id, un nom, une durée, et un style (reggae, rock, etc...). L'album est également lié à une maison de disque car c'est la maison de disque qui produit les musiciens et vend les albums de ces derniers. Un album possède donc un id, un nom, un style (varié, rock, etc...), un prix et un nombre de vente. Enfin, un Musicien peut participer à aucun ou plusieurs concert.

Structured Query Language

SQL

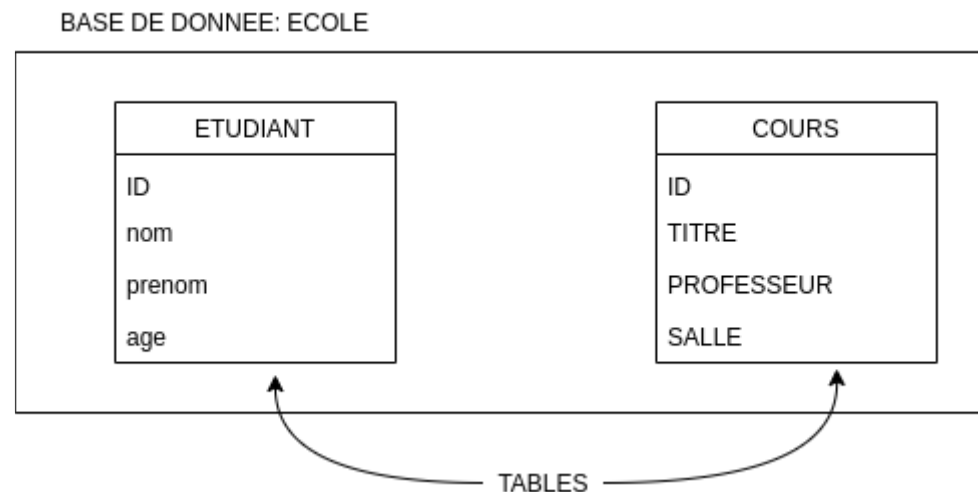


Qu'est-ce que SQL ?

Langage de requête structurée. C'est un langage qui permet d'administrer des bases de données (créer, ajouter, modifier, supprimer ou récupérer des informations).

Comment ça fonctionne ?

Schéma de présentation d'une BDD



Comment ça fonctionne ?

Instructions

le langage SQL est basé sur des "instructions", des "mots" qui vont permettre d'effectuer des actions. Une suite d'instruction est nommé requête. exemple: SELECT, UPDATE, DROP, FROM, ORDER BY, etc...
ATTENTION ! une requête SQL doit obligatoirement finir par un ";", sinon la requête ne s'effectuera pas

Exemple

imaginons que j'ai une base de donnée et que dans cette bdd, j'ai une table nommé ETUDIANT et que je veux récupérer le nom de tous les étudiants de cette table, en sql on pourrait faire:

```
SELECT nom  
FROM ETUDIANT;
```

Quels outils

Pour apprendre et utilisé SQL vous devez choisir un système de gestion de base de données, les plus connus étant:

- MySQL
 - MariaDB
 - PHPMyAdmin (MySQL mais avec interface)
-



Notions de base : CREATE, ALTER, DROP

Avant de pouvoir commencer, il faut d'abord créer sa base de donnée et les tables qui la composent. Pour cela ces instructions sont indispensable:

- **CREATE** : créer base de données ou table
 - **ALTER**: modifie une table existante
 - **DROP**: supprime base de données ou table
-



Exemple

```
// Je souhaite créer une base de données ECOLE
// puis créer la table ETUDIANT vu précédemment:

CREATE DATABASE ECOLE; // Création de la bdd ECOLE
USE ECOLE; //permet d'accéder à la bdd ECOLE
CREATE TABLE ETUDIANT; // Création de la table ETUDIANT dans la bdd
```

Exercices

Installer PHPMyAdmin, puis aller dans la partie SQL en haut (l'objectif est de faire un maximum de chose en SQL et ne pas s'aider de l'interface pour la création) et faites en sorte de pouvoir créer une bdd PREMIEREBDD et créez la table USER dans cette bdd. Le USER devra contenir un id qui sera sa clé primaire, un nom, un prenom et un age.

Correction

```
CREATE DATABASE PREMIEREBDD;
```

```
CREATE TABLE USER (id INT AUTO_INCREMENT, nom varchar(20) NOT NULL, prenom varchar(20) NOT NULL, age INT NOT NULL, PRIMARY KEY(id));
```

Notions de base : INSERT, UPDATE, DELETE

Après avoir créer notre database et notre première table, il faudrait remplir la table USER de nouveaux utilisateurs:

- **INSERT INTO** : ajoute un utilisateur dans une table
 - **UPDATE**: modifie les infos d'utilisateur(s)
 - **DELETE**: supprime l'utilisateur(s)
-

Exemple

```
// Ajout d'un utilisateur nommé Henri dans la table ETUDIANT
INSERT INTO ETUDIANT (nom, prenom, age) VALUES ("Roi de France", "Henri II", 474);

// Je souhaite supprimer mon chère Henri
DELETE FROM USER WHERE nom="Henri II";
```

Exercice

Ajouter 5 nouveaux utilisateurs: qui auront respectivement pour prénom: Victor, Elodie, Romain, Guillaume, Emmanuelle. Les autres informations seront à votre choix. Après avoir Créer ces 5 utilisateurs, on s'est rendu compte que Victore s'écrivait avec un e, nous ne voulons pas supprimer l'utilisateur que nous venons de créer. Faites en sortes de corriger l'erreur sur le prénom de Victore.

Base des requêtes

Instructions	Explication
SELECT	Sélectionner les champs à sélectionner(* signifie 'tout')
FROM	Les champs sélectionner précédemment proviennent des tables suivantes
WHERE	Début des conditions supplémentaires

Les autres

Instructions	Explication
AND/OR/NOT	Après le WHERE obligatoirement, autres conditions
GROUP BY	Regrouper les données en fonction de champ

Instructions	Explication
ORDER BY	Trier les résultats en fonction de champs
COUNT/SUM/AVG	Permet de faire des calcul (compter, additionner, moyenne)



Exemples

```
// Nous souhaitons afficher les élèves dont l'âge est supérieur ou égal à 18ans
```

```
SELECT age
FROM ETUDIANT
WHERE age >= 18;
```

```
// Nous souhaitons récupérer toutes les informations
// des cours que le professeur Patrick a effectué
```

```
SELECT *
FROM COURS
WHERE professeur="Patrick";
```



Exercice

Vous êtes développeur dans une mairie de France, votre supérieur vous donne un fichier .sql contenant les informations sur toutes les villes de France. Vous avez plusieurs informations à récupérer:

- Toutes les informations de la ville de Paris
 - Récupérer les nom et départements de toute les villes se trouvant dans le 77 groupé par nom de ville
 - Afficher les villes qui ont plus de 25 000 habitants en 2012 ordonné par nombre d'habitants
-



Correction

```
//1
SELECT * FROM villes_france_free WHERE ville_nom="Paris";
//2
SELECT ville_nom, ville_departement
FROM villes_france_free
WHERE ville_departement=77 GROUP BY ville_nom;
//3
SELECT ville_nom, ville_population_2012
FROM villes_france_free
WHERE ville_population_2012>25000 ORDER BY ville_population_2012;
```



Création et gestion d'utilisateur

MySQL, MariaDB

```
// Permet de créer un nouvel utilisateur
CREATE USER 'test'@'localhost' IDENTIFIED BY 'newpassword';
```

```
// Permission d'utiliser les SELECT sur toutes les databases
GRANT SELECT ON *.* TO 'test'@'localhost';
// Permissions pour tout
GRANT ALL PRIVILEGES ON *.* TO 'test'@'localhost';
// Supprimer tout les privileges pour la database ECOLE
REVOKE ALL PRIVILEGES ON ECOLE TO 'test'@'localhost';
// Supprimer l'utilisateur
DROP USER 'test'@'localhost'
```

Liens entre les tables

Exercice

Durant toute votre enfance vous aviez une passion pour la chaîne de télévision Nickelodeon, et cela persiste encore aujourd'hui, vous avez retrouvé une vieille base de données que vous aviez fait il y a 5 ans sur l'univers de Bob l'éponge. Vous décidez de voir si tous ce que vous aviez à l'époque, fonctionne encore aujourd'hui.

- afficher le nom, prénom de l'utilisateur et le modèle, dont la marque de voiture est "burger"
 - afficher le nom et prénom des utilisateurs qui pratique le sport "musculatation"
 - créer un nouveau USER nommé garry l'escargot, et faite en sorte qu'il participe au sport "chasse à la méduse"
-

Exercice

Dans cet exercice, le but est de créer la base de données avec les informations ci-après. Sujet: En plus d'être développeur, vous êtes passionnés par le monde de la musique et voulez répertorier tous ce qui tournent autour des musiciens. Vous créez une table musicien contenant un id, un nom et un nom d'artiste. Le Musicien peut créer un ou plusieurs album, à l'inverse, un album de peut être créer que par un ou plusieurs musiciens (ex: les enfoirés). Un album est constitué d'une ou plusieurs chanson. Une chanson possède un id, un nom, une durée, et un style (reggae, rock, etc...). L'album est également lié à une maison de disque car c'est la maison de disque qui produit les musiciens et vend les albums de ces derniers. Un album possède donc un id, un nom, un style (varié, rock, etc...), un prix et un nombre de vente. Enfin, un Musicien peut participer à aucun ou plusieurs concert.
