



LA MANU

L'ÉCOLE DES MÉTIERS DU NUMÉRIQUE
Manufacture de compétences



SUPPORT APPRENANT

SQL – MySQL



CONFIDENTIEL

*Ce document est strictement confidentiel et ne doit pas être diffusé
sans accord préalable écrit*

SQL – MySQL

Ce que l'on a vu...

On parle de base de données à partir du moment où des données vont être récoltées, organisées et stockées afin de répondre à un besoin spécifique, tel que la mise en place d'un système d'informations par exemple. Une base de données peut utiliser différents supports tels que du papier ou des fichiers informatiques. Ainsi, la création d'une base de données pourra se faire soit manuellement, soit par un système automatisé. Les bases de données manuelles, comme les archives d'entreprises par exemple, sont toujours utilisées. Cependant, leur consultation est lente et fastidieuse, il y a également des risques de détérioration, de perte ou encore au niveau de la confidentialité. Alors que les bases de données utilisant un système automatisé permettent une manipulation des données rapide et efficace, un partage de ressources entre plusieurs utilisateurs sans difficulté et une meilleure sécurisation des données grâce à des systèmes d'authentification notamment. L'utilisation de bases de données permet la sauvegarde de données qui seront stockées sur un serveur par exemple, c'est que l'on appelle la persistance des données.

Les **Systèmes de Gestion de Base de Données** (SGBD), ou en anglais DataBase Management System (DBMS), sont des logiciels qui permettent d'administrer des bases de données, stocker des données, les traiter ou les consulter. Ces logiciels sont des intermédiaires entre les utilisateurs et les fichiers de données. Il existe différents types de SGBD, les 2 principaux étant les :

- **SGBDR** : Système de Gestion de Bases de Données Relationnelle. Les SGBDR organisent le stockage des données sur le principe de tables reliées entre elles. Les données de différentes tables peuvent alors communiquer entre elles par le biais de jointures. Dans les SGBDR la redondance des données stockées est limitée. La structure et le type des données sont définis dès la création de la base de données. L'intégrité des données peut aussi être vérifiée par la définition de contraintes à respecter (NULL/NOT NULL, UNIQUE, PRIMARY KEY, AUTO_INCREMENT...). Les instructions pour interagir avec la base de données, dans le cas d'un SGBDR, sont faites avec un langage normalisé, le SQL. Cependant, il y a quelques différences dans la syntaxe et la fonctionnalité des différents SGBDR.
Ex : MySQL, SQL Server, Oracle, PostgreSQL...
- **SGBD NoSQL** : Système de Gestion de Bases de Données Not Only SQL. Dans un SGBD NoSQL les données ne sont pas structurées, la base de données est extensible sans limite et ne vérifie pas l'intégrité des données selon des contraintes définies. Il n'y a pas de langage normalisé à tous les SGBD NoSQL. Ce qui fait que ce système est particulièrement adapté pour le big data et pour des projets dont les exigences en termes de données sont difficiles à déterminer. Les SGBD NoSQL organisent le stockage des données notamment sous forme de collections d'objets dans des documents (ressemblance avec le JSON).
Ex : MongoDB, CouchDB, Big Table, Voldemort ...

SQL



SQL est l'acronyme de **Structured Query Language**, ce qui signifie en français « langage d'interrogation structurée ». C'est un langage de communication avec les bases de données conçu par IBM dans les années 70.

SQL est le langage standard des Systèmes de Gestion de Bases de Données Relationnelles. Il a été normalisé à la fin des années 80-début années 90 pour devenir le langage normalisé de communication avec les bases de données.

Par convention, les mots-clés SQL s'écrivent en majuscules et les noms de bases de données, tables, champs (ou colonnes)... s'écrivent en minuscules. En ajoutant à cela une bonne indentation de son code, cela permet de rendre les requêtes plus lisibles et de dissocier rapidement les différents éléments d'une requête.

Les procédures stockées sont un ensemble de requêtes précompilées qui vont s'exécuter plus rapidement. Elles permettent également de garder une trace des commandes effectuées ce qui peut faciliter la gestion et la maintenance de la base de données. Lors de la création d'une base de données on va d'abord écrire un script d'installation dans un fichier .sql, qui contiendra les requêtes relatives à la création de la base de données. Puis de la même manière on pourra écrire un script de population qui contiendra des requêtes pour remplir la base de données avec des données.

MySQL



MySQL est un Système de Gestion de Bases de Données Relationnelle créé en 1995. Ce SGBDR est libre, gratuit et très populaire, en particulier chez les concepteurs de sites internet en PHP. En effet, MySQL faisant parti des packages LAMP (Linux, Apache, MySQL, PHP), WAMP (Windows, Apache, MySQL, PHP), MAMP (Mac, Apache, MySQL, PHP), il est largement utilisé pour les applications utilisant du PHP. MySQL est souvent combiné au PHP, mais il peut être combiné avec d'autres langages de programmation (Java, Python, C++...) ou s'utiliser seul.

Pour utiliser MySQL on peut passer par le biais d'une interface graphique telle que PhpMyAdmin. Sur MySQL, on utilise des magic quotes (') autour des noms de champs, de tables, de bases de données, pour éviter les confusions et les différencier des mots-clés ou fonctions SQL.

Quelques commandes SQL – MySQL...

Commande	Description
CREATE DATABASE `nomBaseDeDonnées`;	Crée une base de données.
CREATE DATABASE `nomBaseDeDonnées` CHARACTER SET 'utf8';	Crée une base de données avec l'encodage UTF-8.
CREATE DATABASE IF NOT EXISTS `nomBaseDeDonnées`;	Crée une base de données si elle n'existe pas déjà.
DROP DATABASE `nomBaseDeDonnées`; ⚠ Commande dangereuse	Supprime la base de données.
DROP DATABASE IF EXISTS `nomBaseDeDonnées`; ⚠ Commande dangereuse	Supprime la base de données si elle existe.
USE `nomBaseDeDonnées`;	Utilise la base de données, se positionne dessus.
CREATE TABLE `nomTable`(`nomColonne` typeColonne contraintesEventuelles, `nomColonne` typeColonne contraintesEventuelles);	Crée une table avec 2 colonnes.
CREATE TABLE IF NOT EXISTS `nomTable`(`nomColonne` typeColonne contraintesEventuelles, `nomColonne` typeColonne contraintesEventuelles);	Crée une table avec 2 colonnes si elle n'existe pas.
AUTO_INCREMENT	Contrainte d'auto-incrément.

PRIMARY KEY	Clé primaire, permet de s'assurer que chaque entrée est unique.
DROP TABLE `nomTable`; ⚠ Commande dangereuse	Supprime la table.
ALTER TABLE `nomTable` ADD `nomColonne` `typeColonne`;	Ajoute une colonne dans la table.
ALTER TABLE `nomTable` CHANGE `ancienNomColonne` `nouveauNomColonne` `typeColonne`;	Renomme une colonne d'une table.
ALTER TABLE `nomTable` MODIFY `nomColonne` `nouveauTypeColonne`;	Change le type d'une colonne.
INSERT INTO `nomTable` (`nomColonne1`, `nomColonne2`) VALUES (`donnéeLigne1Colonne1`, `donnéeLigne1Colonne2`), (`donnéeLigne2Colonne1`, `donnéeLigne2Colonne2`), (`donnéeLigne3Colonne1`, `donnéeLigne3Colonne2`);	Insert 3 lignes (ou tuples) de données dans une table à 2 colonnes.
SELECT `colonne1`, `colonne2` FROM `nomTable`;	Affiche les données de 2 colonnes d'une table.
SELECT `colonne1`, `colonne2` FROM `nomTable` WHERE condition;	Affiche les données de 2 colonnes d'une table respectant une condition.
LIKE modèle	Permet de rechercher les données correspondant à un modèle particulier.
ORDER BY `nomColonne` ASC	Ordonne les données par ordre alphabétique/croissant sur une colonne. DESC pour ordonner dans l'ordre non-alphabétique/décroissant.
DELETE FROM `nomTable` WHERE condition;	Supprime les lignes (ou tuples) de données de la table répondant à la condition.
UPDATE `nomTable` SET `nomColonne` = `nouvelleValeur` WHERE condition;	Modifie les données d'une table selon une condition.

Liens utiles

<https://sql.sh/>
<https://sql.sh/categorie/bonnes-pratiques>
<https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql>
<https://openclassrooms.com/en/courses/4449026-initiez-vous-a-lalgebre-relationnelle-avec-le-langage-sql>
<https://sql.developpez.com/>
<https://postgresql.developpez.com/cours-complet/sql-conception-mise-en-oeuvre/?page=introduction-et-premiers-select>
<http://deptinfo.unice.fr/~grin/messupports/sql.pdf>
<https://laurent-audibert.developpez.com/Cours-BD/?page=langage-sql>
<https://docs.phpmyadmin.net/fr/latest/intro.html>
<https://dev.mysql.com/doc/refman/8.0/en/introduction.html>
<http://oseox.fr/sql/>
<https://www.w3schools.com/sql/default.asp>

Bonnes pratiques

Rappel des principales bonnes pratiques	
<input type="checkbox"/>	L'utilisation des magic quote pour les noms de base de données, des tables, des champs et des alias.
<input type="checkbox"/>	Ecrire les instructions en majuscules.
<input type="checkbox"/>	Pour une meilleure compréhension du code, il faut indenter son code.
<input type="checkbox"/>	Utiliser les commentaires.
<input type="checkbox"/>	Toujours mettre un WHERE quand on utilise un UPDATE ou un DELETE pour éviter de tout modifier ou tout supprimer.
<input type="checkbox"/>	Utiliser la commande IF NOT EXISTS ou IF EXISTS avant de créer ou supprimer une base de données ou table.
<input type="checkbox"/>	Ne pas utiliser de SELECT * . Il est important de maîtriser ce que nous affichons.
<input type="checkbox"/>	Vocabulaire spécifique : <ul style="list-style-type: none"> • Requête • Base de données • Table • Champ • Auto-incrément • Clé primaire • Script