

SQL – Structured Query Language



- > **INFRASTRUCTURES** SYSTÈMES & RÉSEAUX
- > **CYBERSÉCURITÉ** INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > **TRANSFORMATION NUMERIQUE DES ENTREPRISES**

www.adrar-numerique.com

SQL – Structured Query Language

Pré-requis – Outils et logiciels

Pour la suite de cette formation, vous allez devoir installer quelques outils

1. Serveur de Base de Données

- La solution « simple » sur **Windows** : Installer **WampServer** ou **XAMPP**
Cela vous fournira un serveur LAMP fonctionnel (Apache, MySQL, PHP),
- La solution d'avenir : utiliser **Docker**, un gestionnaire de conteneurs.
C'est la solution que vous aurez le plus de chances de croiser en entreprise.

La courbe d'apprentissage est un peu plus ardue que pour la solution précédente, mais lorsque vous avez une config fonctionnelle, c'est beaucoup plus efficace et souple.

Docker est une solution **multi-plateformes**, fonctionnant aussi bien sous Windows, Linux ou Mac.

SQL – Structured Query Language

Installation de Docker sous Windows

1. Installer Docker pour Windows (<https://www.docker.com/>)
2. Installer WSL 2 (Windows Subsystem for Linux)
 - Ouvrir un terminal powershell et entrer les commande suivante :
wsl.exe --update
Puis :
wsl --set-default-version 2
 - Ajouter votre utilisateur local au groupe docker-users
 - Ouvrez un terminal PowerShell en tant qu'administrateur et entre la commande suivante :
net localgroup docker-users "your-user-id" /ADD
 - Fermez votre session et rouvrez là

SQL – Structured Query Language

Installation de Docker sous Windows

- Créez un répertoire qui sera votre répertoire de travail, puis clonez-y le dépôt git suivant :
git clone <https://github.com/guirod/docker-lamp.git>
Ce dépôt contiendra l'essentiel pour la suite de votre formation (serveur MySQL, serveur Apache, PHP 8.1, serveur SMTP + mailcatcher) et sera mis à jour au besoin
- Quelques commandes utiles :
 - **Lancer et couper vos containers (serveurs) :**
docker-compose up -d
docker-compose down
 - **Lister les containers**
docker container ls
 - **Obtenir une connexion en root sur le container**
docker exec -it nom_du_container executable
 - **Exemple pour se connecter au container apache et lancer l'exécutable "bash"**
docker exec -it docker-lamp-docker-lamp-php-1 bash

SQL – Structured Query Language

Installation d'une GUI (Graphic User Interface) pour ses bases de données

- 2 principaux outils, je vous conseille d'installer les 2.
 - MySQL Workbench
 - <https://www.mysql.com/fr/products/workbench/>
 - Graphiquement plus agréable
 - Parfois un peu lourd et buggué pour certaines tâches (import et export notamment)
 - HeidiSQL
 - <https://www.heidisql.com/>
 - Un peu moins convivial, mais très puissant.
 - Compatible avec de nombreux SGBDr (MySQL/MariaDB, PostgreSQL, SQL Server)
- Pour chacun de ces outils, configurez et sauvegarder votre connexion :
 - Host : localhost / 127.0.0.1
 - Username : root
 - Password : p@ssw0rd pour les utilisateurs Docker (configuré dans le fichier compose.yaml)

SQL – Structured Query Language

Connexion en ligne de commande

- Se connecter au container MySQL

```
docker exec -it docker-lamp-docker-lamp-mysql-1 bash
```

- Se connecter avec son utilisateur

```
mysql -hlocalhost -uroot -pp@ssw0rd (problème, le pass sera dans votre bash history)  
mysql -h localhost -u root -p (un prompt vous demandera votre mdp)
```

- Enjoy (Non. Mais parfois vous n'aurez pas le choix)

SQL – Structured Query Language

SQL Introduction

SQL – Structured Query Language

SQL – Structured Query Language

- Langage informatique normalisé
- N'est pas un langage procédural
- Créé par IBM en 1974
- Normalisé depuis 1986
 - Recommandation de l'ANSI
 - Norme ISO/CEI 9075 - Technologies de l'information - Langages de base de données SQL

SQL – Structured Query Language

SQL – Structured Query Language

Le langage SQL permet principalement de faire du CRUD :

- Create : Ecrire, insérer des données
- Read : Lire des données
- Update : Modifier des données
- Delete : Supprimer des données

Il permet aussi de modifier la structure de la base de données :

- Créer, modifier, supprimer des bases de données
- Créer, modifier, supprimer des tables
- Créer, modifier, supprimer des utilisateurs et gérer leurs privilèges

SQL – Structured Query Language

Base De Données Relationnelle

- 3 objectifs :
 - **Garantir l'intégrité des données**
 - Eviter l'altération des données (usure, pannes, erreurs, malveillance)
 - Eviter l'incohérence des données
 - La duplication des données : exemple 2 adresses différentes pour un utilisateur
 - Les valeurs aberrantes : exemple Age < 0
 - **Garantir l'indépendance entre données et traitements**
 - L'information correspondant à une donnée doit être compréhensible indépendamment
 - Si une donnée est calculée, on évitera de la stocker en BDD
 - **Traitements rationalisés**
 - Une fois les données définies, les traitements consistent principalement à ajouter, modifier, supprimer et consulter des données

SQL – Structured Query Language

SGBDr – Système de Gestion de Base de Données Relationnelles

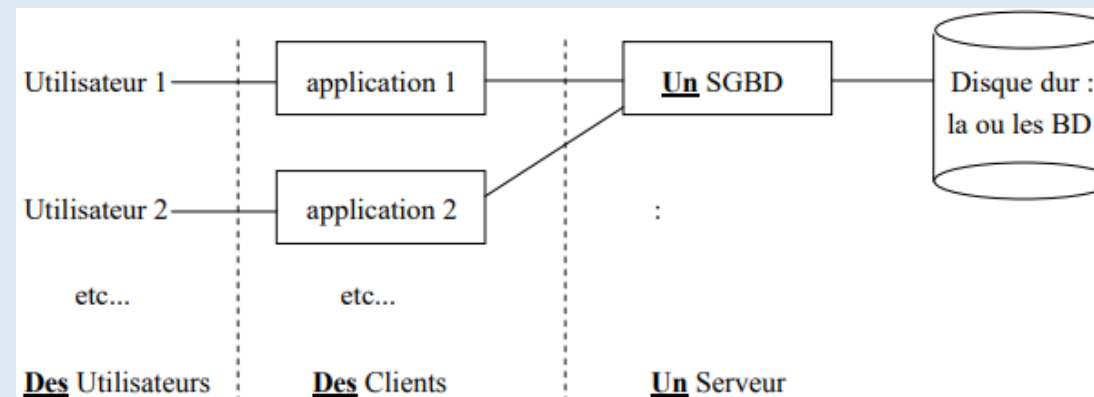
- Ensemble de logiciels faisant l'interface entre l'utilisateur et la BDD
- **Objectifs :**
 - Stockage et requêtage des données
 - Multi-utilisateurs => Gestion des droits d'accès, des collisions
 - Limiter la redondance (duplication de données) => Intégrité des données

SQL – Structured Query Language

SGBDr – Système de Gestion de Base de Données Relationnelles

■ Architecture

- Le plus souvent on utilise une architecture client/serveur



- Une BDD est stockée sur 1 ou plusieurs disques durs
- 1 seul SGBD par BDD
- Plusieurs applications / API (clients) peuvent communiquer avec le SGBD

SQL – Structured Query Language

SGBDr – Système de Gestion de Base de Données Relationnelles

- L'objectif majeur du SGBD étant d'assurer l'intégrité, voici quelques moyens mis en œuvre
 - Conformité au modèle : Vérifier que les données saisies soient cohérentes
 - Accès concurrents : Eviter les incohérences dues à des modifications multiples au même moment (transactions)
 - Gestion sur panne : Garantir le maintien de la cohérence même quand une panne intervient
 - Autorisations d'accès

SQL – Structured Query Language

SGBDr – Système de Gestion de Base de Données Relationnelles

- Les SGBDr les plus connus :
 - MySQL (Oracle) / MariaDB (Fork open source de MySQL)
 - OracleDB
 - PostgreSQL : alternative open source à OracleDB
 - SQL Server (Microsoft)

SQL – Structured Query Language

SQL

Administration d'une BDD

SQL – Structured Query Language

Création de BDD

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name [create_option]...  
create_option: [DEFAULT] {  
    CHARACTER SET [=] charset_name |  
    COLLATE [=] collation_name |  
    ENCRYPTION [=] {'Y' | 'N'}  
}
```

IF NOT EXISTS : Si cette option est présente, cette commande ne lèvera pas d'exception si la BDD existe déjà. Il n'y fera aucune modification.

create_option permet de définir des paramètres de la BDD.

Pour plus d'informations : <https://dev.mysql.com/doc/refman/8.0/en/create-database.html>

SQL – Structured Query Language

Suppression et modification de BDD

- SUPPRESSION

DROP {**DATABASE** | **SCHEMA**} [**IF EXISTS**] *db_name*;

- MODIFICATION

ALTER {**DATABASE** | **SCHEMA**} [*db_name*] *alter_option* ...

alter_option: { [**DEFAULT**] **CHARACTER SET** [=] *charset_name* |
[**DEFAULT**] **COLLATE** [=] *collation_name* |
[**DEFAULT**] **ENCRYPTION** [=] {'Y' | 'N'} |
READ ONLY [=] {**DEFAULT** | 0 | 1} }

- SELECTIONNER UNE BDD : Nécessaire pour pouvoir effectuer des opérations (création ou consultation)

USE *db_name*;

- DIVERS

SHOW DATABASES;

SQL – Structured Query Language

Gestion des droits et utilisateurs

■ Sécurité

- Limiter au strict nécessaire les droits des utilisateurs exposés (application web)
- Utiliser des mots de passe forts et uniques
- Maintenir ses logiciels à jour

SQL – Structured Query Language

Gestion des droits et utilisateurs

- Création d'un utilisateur

```
CREATE USER 'alice'@'localhost' IDENTIFIED BY 'P@ssw0rd';
```

- Assigner les privileges

```
GRANT PRIVILEGE ON database.table TO 'username'@'host';
```

Exemple :

```
GRANT CREATE, ALTER, DROP, INSERT, UPDATE, DELETE, SELECT, REFERENCES, RELOAD  
ON *.* TO 'alice'@'localhost' WITH GRANT OPTION;
```

- Révoquer un privilege

```
REVOKE type_of_permission ON database_name.table_name FROM 'username'@'host'
```

SQL – Structured Query Language

Gestion des droits et utilisateurs

- Appliquer les modifications:
FLUSH PRIVILEGES;
- Afficher les privileges d'un utilisateur
SHOW GRANTS FOR 'username'@'host';
- Supprimer un utilisateur
DROP USER 'username'@'localhost';

SQL – Structured Query Language

Exercice

- Coder un script pour :
Créer un autre utilisateur 'bob'
Créer une BDD pour bob lui assignant tous les privileges
- Ressource :
<https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql>

SQL – Structured Query Language

SQL

Structure des données

SQL – Structured Query Language

Les Tables

- Les tables sont les objets qui contiennent toutes les données d'une BDD.
- Données organisées dans un tableau à deux dimensions
 - Les lignes sont des tuples, enregistrements ou n-uplets
 - Les colonnes sont des attributs

SQL – Structured Query Language

Les Tables

| id | nom | prénom | profession | code postal | ville |
|----|---------|----------|----------------------|-------------|------------|
| 1 | Durand | Michel | Directeur | 75016 | Paris |
| 2 | Dupond | Karine | Secrétaire | 92000 | Courbevoie |
| 3 | Mensoif | Gérard | Commercial | 75001 | Paris |
| 4 | Monauto | Alphonse | Commercial | 75002 | Paris |
| 5 | Emarre | Jean | Employé | 75015 | Paris |
| 6 | Abois | Nicole | Secrétaire | 95000 | St Denis |
| 7 | Dupond | Antoine | Assistant commercial | 75014 | Paris |

SQL – Structured Query Language

Les types de données

- Numériques exacts :
 - Entiers signés :
 - INTEGER 4 octets => -2147483648 à +2147483647
 - SMALLINT (2 octets), BIGINT (8 octets)
 - Décimaux signés :
 - NUMERIC, DECIMAL
- Numériques approximatifs
 - REAL, DOUBLE PRECISION, FLOAT
- Chaînes de caractères
 - Longueur fixe : CHAR
 - Longueur variable : VARCHAR
 - Objets de type caractère : CLOB

SQL – Structured Query Language

Les types de données

- Chaînes binaires
 - Nombre d'octets fixe : BINARY
 - Longueur variable : VARBINARY
 - Binary Large Object : BLOB
- Booléens
 - 3 valeurs : true, false, unknown
- Dates et heures
 - DATE
 - TIME/TIMESTAMP WITH/WITHOUT TIMEZONE
- DIVERS
 - XML, ARRAY, INTERVAL

SQL – Structured Query Language

Opérations sur les tables

- Création

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name  
    (create_definition,...)  
    [table_options]  
    [partition_options]
```

Exemple :

```
CREATE TABLE users (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    name VARCHAR(100),  
    email VARCHAR(255) NOT NULL,  
    birthdate DATE  
)
```

<https://dev.mysql.com/doc/refman/8.0/en/create-table.html>

SQL – Structured Query Language

Opérations sur les tables

- Modification :
 - Ajouter/Modifier/Supprimer un attribut
 - Ajouter/Modifier/Supprimer une contrainte
 - Modifier des propriétés de la table

ALTER TABLE *tbl_name*
 [*alter_option* [, *alter_option*] ...]
 [*partition_options*]

Exemple :

ALTER TABLE *users*
 ADD *firstname* VARCHAR(100) **AFTER** *name*;

<https://dev.mysql.com/doc/refman/8.0/en/alter-table.html>

SQL – Structured Query Language

Indexes

- Dans chaque table, on peut définir un ou plusieurs indexes.
- Un ROWID est créé pour chaque enregistrement
- Avantage : Requêtes plus rapides
- Inconvénient : Peuvent surcharger le système donc à utiliser avec précaution
- Exemple :

Création d'un index : Notez que l'on peut le définir sur une ou plusieurs colonnes

CREATE INDEX index_name **ON** table_name (column1, column2, ...);

Suppression d'un index

ALTER TABLE table_name **DROP INDEX** index_name;

SQL – Structured Query Language

Contraintes

Les contraintes sont des règles appliquées aux attributs d'une table. Elles permettent d'améliorer l'intégrité des données de la base.

- Les contraintes les plus communes sont :
 - **NOT NULL** : Contraint à avoir une valeur différente de NULL.
 - **DEFAULT** : Fournit une valeur par défaut à la colonne.
 - **UNIQUE** : Ne permet pas que 2 enregistrements aient la même valeur pour une colonne (email)
 - **CHECK** : Active une condition pour qu'un enregistrement soit valide.
 - **INDEX**
 - **PRIMARY KEY** : La clé primaire permet d'identifier de manière unique chaque enregistrement. Une clé primaire peut être composée, elle fait dans ce cas référence à plusieurs attributs de la table.
 - **FOREIGN KEY** : Une clé étrangère permet de relier 2 tables.

SQL – Structured Query Language

Contraintes - Clés

■ Clé primaire :

Une clé primaire permet d'identifier chaque enregistrement d'une table.

Par définition, elle doit être UNIQUE et NOT NULL.

Une clé primaire peut être simple (définie par un seul attribut) ou composée de plusieurs attributs.

■ Clé étrangère :

Une clé étrangère est une contrainte qui permet d'associer plusieurs tables entre elles tout en assurant l'intégrité des données.

Une clé étrangère fait référence à une clé primaire d'une autre table.

A noter :

- Il est impossible de faire référence à une clé primaire si celle-ci n'existe pas dans la BDD.
- Il n'est par défaut pas possible de supprimer un enregistrement si sa clé primaire est référencée comme clé étrangère dans une autre table.

SQL – Structured Query Language

Contraintes – ON DELETE / ON UPDATE

- Clé étrangère :

Si un enregistrement est référencé dans une autre table, le SGBD empêchera la modification ou suppression de la clé primaire.

Si l'on souhaite forcer la modification / suppression, il est nécessaire de le renseigner lors de la création de la clé secondaire.

Pour cela on doit utiliser les instructions ON DELETE et ON UPDATE

- ON DELETE : A la suppression, effectue une action sur tous les enregistrements enfants.
- ON UPDATE : A la modification, reporte le changement sur tous les enregistrements enfants.

L'usage le plus courant est d'utiliser ON DELETE CASCADE ou ON UPDATE CASCADE.

Toutefois, c'est un comportement assez destructeur ce qui n'est pas toujours souhaitable.

Dans ce cas, on peut utiliser des alternatives, telles que :

ON DELETE | UPDATE [SET NULL | SET DEFAULT | NO ACTION | CASCADE]

SQL – Structured Query Language

Contraintes - Exemple

```
CREATE TABLE employes(  
    id INT NOT NULL,  
    nom VARCHAR (100) NOT NULL,  
    email VARCHAR (100) NOT NULL UNIQUE,  
    age INT NOT NULL CHECK (age >= 18),  
    salaire DECIMAL (18, 2) DEFAULT 3000.00,  
    PRIMARY KEY (id)  
);  
CREATE TABLE congés (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,  
    date_debut DATE NOT NULL,  
    date_fin DATE NOT NULL,  
    id_employe INT,  
    FOREIGN KEY (id_employe) REFERENCES employes(id) ON DELETE CASCADE  
    -- Pour pouvoir nommer la contrainte, utiliser la syntaxe suivante  
    CONSTRAINT fk_id_employe FOREIGN KEY (id_employe) REFERENCES employes(id)  
);
```

SQL – Structured Query Language

Contraintes - Suppression

- En ciblant la contrainte :

ALTER TABLE table_name

ALTER COLUMN column_name **DROP CONSTRAINT**;

Ex : **ALTER TABLE** employe

ALTER COLUMN salaire **DROP DEFAULT**;

- Si la contrainte a un alias

ALTER TABLE table_name **DROP CONSTRAINT** constraint_alias;

SQL – Structured Query Language

Associations

Lorsque l'on modélise notre BDD, on va identifier différents types d'associations qui vont lier les tables entre elles.

Bien concevoir ses associations est indispensable pour satisfaire les conditions des formes normales et assurer l'intégrité des données.

On dénombre **5 types de relations** :

- One-to-One
- One-to-Many / Many-to-One
- Many-to-Many
- Self-Reference (association récursive)

SQL – Structured Query Language

Associations – One-to-One

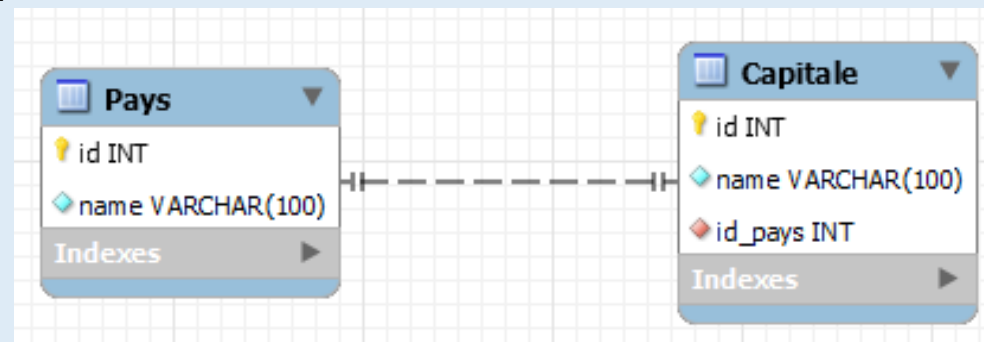
Permet d'associer un et un seul enregistrement d'une table T1 à un et un seul enregistrement d'une table T2.

Implémentée par l'utilisation d'une clé étrangère.

- Veiller à ajouter des contrôles (code et/ou BDD) pour assurer l'intégrité des données (ex : contrainte d'unicité sur la clé étrangère)
- Cette contrainte sera rarement utilisée, car généralement, la donnée peut directement être ajoutée dans la table « parent »

Exemples :

- Citoyen <-> Passeport
- Capitale <-> Pays



SQL – Structured Query Language

Associations – One-to-Many et Many-to-One

One-to-Many :

Associe un enregistrement d'une table T1 à de multiples enregistrements d'une table T2

Many-to-One :

Associe de multiples enregistrements d'une table T1 à un unique enregistrement d'une table T2

Implémentée par l'utilisation d'une clé étrangère.

- Veiller à ajouter des contrôles (code et/ou BDD) pour assurer l'intégrité des données.

Exemple :

- Pays <-> Ville : Un pays possède plusieurs villes, une ville n'appartient qu'à un seul pays



SQL – Structured Query Language

Associations – Many-to-Many

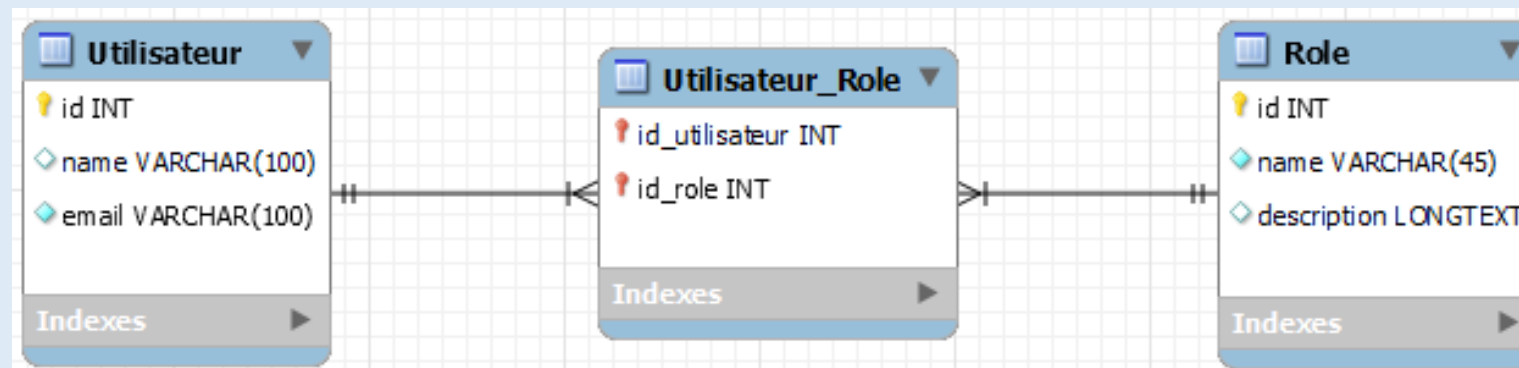
Associe un enregistrement d'une table T1 à de multiples enregistrements de la table T2 et vice versa.

Pour implémenter une association Many-to-Many, il est nécessaire de passer par une **table d'association**.

La table d'association peut aussi être **porteuse de donnée**, par exemple ci-dessous, la table Utilisateur_Role pourrait indiquer la date d'expiration du rôle.

Exemple :

- Utilisateur <-> Rôle : un utilisateur peut avoir plusieurs rôles et un rôle peut être possédé par plusieurs utilisateurs.



SQL – Structured Query Language

Associations – Self-Reference

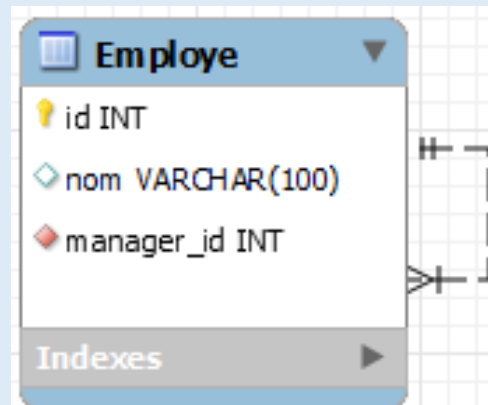
Une association Self-Reference permet d'associer un enregistrement d'une table T à un autre enregistrement d'une table T.

L'association sera souvent une association One-to-Many ou Many-to-One.

C'est notamment utilisé lorsque l'on souhaite gérer une arborescence.

Exemples :

- Catégories <-> Catégorie enfant
- Employé <-> Manager



SQL – Structured Query Language

SQL Ressources

SQL – Structured Query Language

Documentation

- Site officiel :
<https://www.mysql.com/>
- Documentation officielle :
<https://dev.mysql.com/doc/refman/8.0/en/>
- Tuto W3Schools :
<https://www.w3schools.com/sql/default.asp>
- MySQL Tutorial :
<https://www.mysqltutorial.org/>
- Slide sur l'optimisation :
<https://www.slideshare.net/ZendCon/joinfu-the-art-of-sql-tuning-for-mysql-presentation>

SQL – Structured Query Language

SQL Travaux Pratiques

SQL – Structured Query Language

Documentation

- Site officiel :
<https://www.mysql.com/>
- Documentation officielle :
<https://dev.mysql.com/doc/refman/8.0/en/>
- Tuto W3Schools :
<https://www.w3schools.com/sql/default.asp>
- MySQL Tutorial :
<https://www.mysqltutorial.org/>
- Slide sur l'optimisation :
<https://www.slideshare.net/ZendCon/joinfu-the-art-of-sql-tuning-for-mysql-presentation>