

# Notes de cours SQL

Thibaut Marmey

September 15, 2018

## 1 Commandes SQL

### 1.1 Utiliser une database déjà construite

- Lien du tutorial
- Connecter à au serveur MySQL : `mysql -u root *p`
- Pour Charger la base de données *SOURCE world.sql*;
- Pour obtenir les informations de la BdD (nom des colonnes, type de données...) :  
*DESCRIBE nom\_du\_tableau* ou *SHOW COLUMNS FROM nom\_du\_tableau*
- Pour sélectionner une colonne du tableau : *SELECT nom\_du\_champ FROM nom\_du\_tableau*
- Pour compter le nombre d'éléments dans une colonne : *SELECT COUNT(\*) FROM nom\_du\_champ*
- La commande *SELECT* permet entre autres de joindre un autre tableau aux résultats, filtrer, classer, grouper les résultats. Une utilisation presque complète est de la forme :  
*SELECT a FROM table WHERE condition GROUP BY expression HAVING*  
*condition {UNION — INTERSECT — EXCEPT} ORDER BY expression LIMIT*  
*count OFFSET start*
- Pour sélectionner les éléments sans les doublons : *SELECT DISTINCT*. Privilégier *GROUP BY* quand cela est possible pour optimiser les performances.
- Renommer une colonne lors d'une requête SQL : *AS*.  
Exemple d'utilisation seulement sur colonne 1: *SELECT colonne1 AS c1, colonne2 FROM nom\_de\_la\_table*
- Récupérer une colonne d'une base de donnée : en mode orienté objet : *tableau.colonne*
- Extraire des lignes de la base de données qui respectent une condition *WHERE condition*
- Combiner plusieurs condition avec *AND* & *OR* : *WHERE cd1 AND cd2*

- Pour récupérer les éléments d'une colonne par leur nom utiliser la fonction *LIKE* :  
WHERE colonne LIKE 'A%'.  
'A%' : mot commençant par A  
'%Z' : mot finissant par Z  
'A%Z' : mot commençant par A et finissant par Z et n'importe quelle chaîne de caractères entre  
'A\_Z' : unique caractère possible entre A et Z
- *SELECT \* FROM city AS c WHERE (c.CountryCode = 'FRA' AND c.population > 300000) OR (c.CountryCode = 'ESP' AND c.Name LIKE 'T%');*  
Permet d'afficher les villes françaises de plus de 300000 habitants et les villes espagnoles commençant par la lettre T via la fonction *LIKE*. Tout en changeant l'appel de la table 'city' par 'c' via *AS*.
- Simplifier l'utilisatin de *OR*, utiliser la fonction *IN* :  
*WHERE colonne1 IN (val1, val2, val3)* permet de renvoyer les éléments qui on les valeurs val1 ou val2 ou val3.  
Il était possible d'utiliser la fonction *OR* mais la fonction *IN* rend la lecture plus facile.
- Obtenir un intervalle de données : *WHERE col (NOT) BETWEEN start AND end*
- Filtrer les données de type NULL : *WHERE col IS (NOT) NULL*
- Regrouper des données similaires et utiliser des fonctions pour les traiter : *GROUP BY*.  
*SELECT Continent, COUNT(\*) FROM country GROUP BY Continent;*  
Permet de récupérer le nombre de pays dans chaque continent  
Autres fonctions qui peuvent être utilisées : *AVG()*, *MAX()*, *MIN()*, *SUM()*
- Filtrer ces fonctions avec la : *HAVING*.  
*SELECT Continent, COUNT(\*) AS Total FROM country GROUP BY Continent HAVING Total BETWEEN 10 AND 50;*
- Trier les colonnes dans l'ordre ascendant ou descendant :  
*ORDER BY col1 ASC, col2 DSC*
- Pour filtrer le nombre de résultat dans la table utiliser les fonctions *LIMIT* et *OFFSET*.  
*LIMIT 5 OFFSET 10* prendra les valeurs de la 11eme à la 15eme ligne.  
MySQL accepte la syntaxe que nous venons de voir. Cependant pour une faciliter la migration vers un autre SGBD (système de gestion de bases de données) ex PostgreSQL privilégier la syntaxe suivante :  
*LIMIT 10, 5* Le premier est l'OFFSET le suivant la limite.  
D'où *LIMIT 5 OFFSET 10 == LIMIT 10, 5*
- Retourner un résultat entre plusieurs conditions : *CASE ... END*  
*SELECT col, CASE*  
*WHEN cd1 THEN output*  
*ELSE output2*  
*END FROM table*

- Réunir deux tables en excluant les doublons : *UNION*  
*SELECT table1 UNION SELECT table2*
- Réunir deux tables en incluant les doublons : *UNION ALL*
- Appliquer condition sur une autre table : *IN*  
*SELECT col1 FROM tab1 WHERE col2 IN (SELECT col3 FROM tab2 ...)*
- Modifier les données d'une BdD et la mettre à jour : *UPDATE*  
Attention faire un back-up de la BdD en cas de problème car ce n'est pas possible de revenir en arrière.  
Sinon utiliser les commandes *TRANSACTION* pour utiliser *BEGIN TRANSACTION*, *ROLLBACK TRANSACTION*, *COMMIT TRANSACTION*

## 1.2 MySQL Workbench

- Importer BdD avec *Data Import/Restore* et double cliquer sur la BdD dans la section *SCHEMAS*

## 1.3 Autres

- Changer Mot de passe : *sudo mysql \_secure\_installation*
- Lien pour listes de base de données
- *sudo mysql*  
*ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'root';*