**Le langage SQL**

Table des matières

[1. Introduction SQL 2](#_Toc93571071)

[2. Le langage 2](#_Toc93571072)

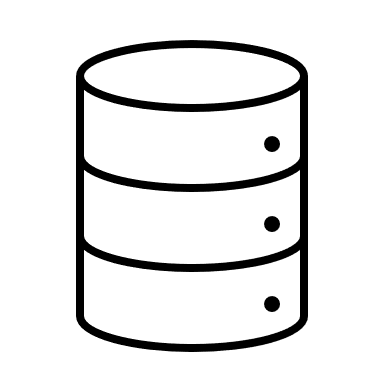
[2.1 Langage de définition de donnée LDD 2](#_Toc93571073)

[2.2 Langage de manipulation de données LMD 3](#_Toc93571074)

# 

# Introduction SQL

* SQL structured Query Language (langage de requête structurée) est un langage de commande, constitué d’instruction (tâche), ‘operand’ (valeur sur laquelle effectuer la tâche) et des opérateur (calcul et comparaison).
* Il y existe 4 grands types de donnée : chaine de caractère, numérique, booléen et date.



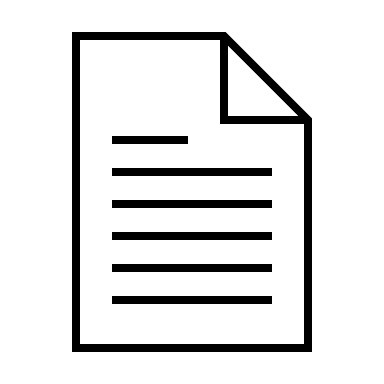
Requête

BDD

-Espace de stockage - SGBD

Interactions

SQL



Résultat

Réponses

**Client**

**Serveur**

* L’outils client et le serveur peuvent être sur le même PC.
* La base de données est d’abord établie avec le MCD ensuite traduit en MLD selon les règles de passe des 2 modèles. A partir du modèle physique (Tables), on pourra entrer des commandes pour créer, manipuler et contrôler la base de données

# Le langage

## Langage de définition de donnée LDD

En terme générale ce langage nous permet de créer et modifier l’organisation des données. A l’aide des différentes commandes ce langage permet de créer les tables, de modifier la structure d’une table existante, de créer les contraintes d’intégrité, les clés étrangères et primaires.

* **Création de table**

**Create Table** Nom\_table\_

(

Nom\_1ere\_colonne Type\_Variables Contrainte (si besoin),

Nom\_2e\_Colonne Type\_variables Contrainte (si besoin),

) ;

* **Modification et suppression d’une table**

**Alter Table :** commande permettant d’ajouter une colonne (ADD COLUMN), de supprimer une colonne (DROP COLUMN), de modifier une colonne (ALTER COLUMN) ainsi que d’ajouter, modifier et supprimer une contrainte (ADD ou ALTER ou DROP CONSTRAINT)

**Alter Table** nom\_table **ADD** nom\_colonne Type **;**

**Alter Table** nom\_table **DROP** nom\_colonne

* **Les contraintes**

Les contraintes d’intégrité servent à contrôler les informations dans les structures. Ce sont des conditions qui doivent être respectées par l’utilisateur pour que les données soient enregistrées dans la BDD. En voici des exemples :

**UNIQUE**: pour refuser les doublons dans les valeurs de la même colonne.

**NOT NULL** : pour refuser les valeurs nulles dans une colonne

**DEFAULT** : pour remplir une valeur par défaut dans une cellule s’il y a aucune valeur.

## Langage de manipulation de données LMD

Ce langage permet d’afficher les résultats des requêtes (LID) mais également d’insérer, modifier et supprimer de nouvelles lignes.

#### Langage d’interprétation de Données

* **Affichage des données (table)**

La commande **SELECT** permet de ressortir les données pour les traitements, très utilisé par les développeurs notamment lors des tests unitaires (entre autres). L’opérateur de sélection **WHERE**, permet de spécifier l’identifiant des lignes ou le nom des colonnes sur lesquelles effectuer la sélection pour le traitement demandé. Syntaxiquement, la condition après WHERE se fait à l’aide des opérateurs de comparaison.

**SELECT** nom\_colonne **FROM** nom\_table

**WHERE** nom\_col ou id\_ligne (comparé à quelque chose)

* **Renommage à l’affichage des résultats :**

**SELECT** nom\_table **AS** nouveau\_nom\_table

Lister les valeurs de deux colonnes dont les noms sont modifiés :

**SELECT** colonne1 **AS** NOM1, colonne2 **AS** NOM2

Cela permet de changer le résultat que à l’affichage et non dans la BDD,

* **Tri des colonnes et lignes**

On peut faire le tri par ordre croissant (ASC) ou décroissant (DESC) :

**SELECT** colonne 1, colonne 2, colonne3 **from** table\_nom

**WHERE** [conditions]

**ORDER BY** nom\_colonne 1 **ASC,** nom\_colonne2 **ASC,** nom\_clonne3 **DESC** **;**

Sélectionner les mots de la colonne qui commence par A.

**SELECT \*** **FROM** table\_nom **WHERE** nom\_colonne **LIKE** **‘**A**%’** (

* **Les jointures**

Les jointures permettent de joindre deux tables différentes contenant les clés étrangères.

**SELECT** nom\_table.nom\_col **FROM** nom\_table INNER JOIN nom\_table2

ON nom\_table.ID\_col = nom\_table2.ID\_col

#### Manipulation de données

* **Insérer**

**INSERT INTO** nom\_table (nom1erecolonne**,** nom2emecolonne**,** nom3emecolonne**)**

**VALUES** **(‘**valeur1**’, ‘**valeur2**’, ‘**valeur3**’)**

* **Mise à jour**

On utilise cette fonctionnalité pour modifier des lignes ou des colonnes, et elle permet aussi de supprimer et/ou de modifier des cellules précises dans une table.

**UPDATE** nom\_table **SET** nom de la colonne 1= ‘Nouvelle\_valeur’,

nom colonne 2 **=** **‘**Nouvelle\_valeur’

**WHERE** id\_colonne = 3

🡺Modifie une ligne **3** entière pour une table à 2 colonnes :

On peut également changer toutes les lignes d’une colonne.