**SQL**

Table des matières

[Chapitre 1 DEFINITION & OBJECTIFS du SQL 2](#_Toc92998229)

[Chapitre 2 : LDD (LANGAGE DE DIFINITION DE DONNEES) 3](#_Toc92998230)

[2. 1 Création de table : 3](#_Toc92998231)

[2. 2 Modifier la structure de la table : 3](#_Toc92998232)

[Chapitre 3 : LMD (LANGAGE DE MANIPULATION DE DONNEES) 4](#_Toc92998233)

[3. 1 INSERT : 4](#_Toc92998234)

[3. 2 UPDATE (Mise à jour) : 4](#_Toc92998235)

[3.3 DELETE : 4](#_Toc92998236)

[Chapitre 4 : LID (Langage d’Interprétation de Données) : 5](#_Toc92998237)

[4. 1 : SELECT 5](#_Toc92998238)

[4. 2 : WHERE 5](#_Toc92998239)

[Titre 2 5](#_Toc92998240)

[Chapitre 2 5](#_Toc92998241)

[Titre 1 5](#_Toc92998242)

[Chapitre 3 6](#_Toc92998243)

[Titre 1 6](#_Toc92998244)

[Chapitre 4 6](#_Toc92998245)

[Titre 1 6](#_Toc92998246)

[Chapitre 5 6](#_Toc92998247)

[Titre 1 6](#_Toc92998248)

[Chapitre 6 6](#_Toc92998249)

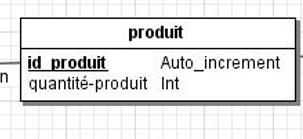
[Titre 1 6](#_Toc92998250)

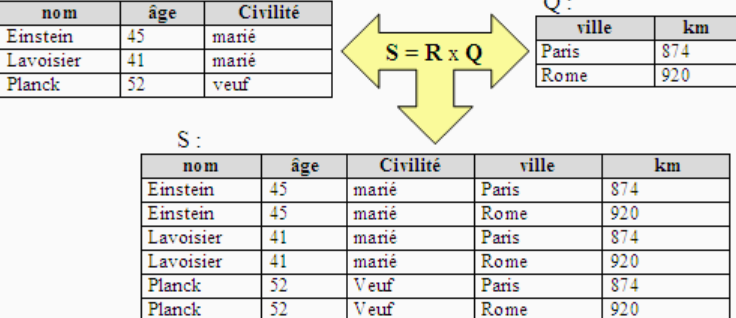
# 

# DEFINITION & OBJECTIFS du SQL

SQL = Structured Language Query, donc sql comme son l’indique c’est un langage de communication avec le SGBD.

L’objectif de SQL est de créer (passage du MLD vers le MPhysiqueD) et structurer les tables d’une base de données relationnelle, modifier, manipuler la BDD

 SQL



MDL TABLES

Résultats= table(affichage)

BDD

Requête SQL

Demandeur du service (client) Serveur (c’est lui qui contient le SGBD) Affichage

Exemple de SGBD : Postgress, Mysql (c’est eux qui gèrent la BDD qui se retrouve en zone mémoire)

Exemple d’outils pour envoyer des requêtes : Pgadmin, sqllite (servent à envoyer des requêtes)

En fait il existe 5 types de langages SQL :

1. **LDD** (Langage Définition des Données) : son but est de modifier la structure des données(colonnes) et création de nouvelles tables. Ses commandes principales ses mots réservés sont : **Create, Alter, Drop.**
2. **LID** (Langage d’Interrogation de Données) : son objectif est de rechercher les valeurs. Son mot réservé est : **SELECT**
3. **LMD** (Langage de Manipulation de données) : comme son nom l’indique il sert à manipuler les données. Ses mots réservés sont :

**INSERT :** insérer des lignes dans une table

**UPDATE :** modifier les valeurs des cellules de la table **DELETE :** supprimer des lignes d’une table

1. **LCT** (Langage de contrôle de transactions) : (transactions= requêtes=échanges) Très utilisé par les développeurs car il aide à valider ou annuler les transactions qu’on a appliquée à la BDD. Ses commandes principales sont :

**COMMIT** : Valider une requête

**Rollback :** revenir en arrière (c’est l’équivalent du rôle de Ctrl Z en Word.) revenir en arrière.

1. **LCD** (Langage de Contrôle des données) : C’est le langage des **administrateurs** de bases de données, le but est de limiter l’accès aux données. Car il existe trois niveaux d’accès à des bases de données : niveau **visiteur** (accès à une partie ou toute la BDD pour juste la consulter), niveau **utilisateu**r (accès à consulter et modifier une partie de la BDD) et puis le niveau **administrateur** (Open-Bar), ses principales commandes sont :

**GRANT :** c’est pour donner l’accès à quelqu’un

**REVOKE :** c’est pour supprimer l’accès à quelqu’un.

Dans ce cours on revient en détail sur 4 premières fonctionnalités (type de langage) à savoir : LDD, LID, LMD et LCT.

# Chapitre 2 : LDD (LANGAGE DE DIFINITION DE DONNEES)

C’est avec langage, avec ces différentes commandes (mots clés), qu’on crée les tables, on modifie la structure d’une table, on crée les contraintes d’intégrité, les clé étrangères et primaires du modèle MLD (Modèle Logique de Données). Une fois que notre MLD est réalisé avec une optimisation optimale on commence alors à créer physiquement (MLP=modèle physique) grâce à ce langage. Le plus souvent ici on utilise les contraintes d’intégrité pour contrôler l’information de la structure.

UNIQUE : c’est pour refuser les doublons dans la même colonne.

NOT NULL : c’est pour refuser les valeurs nulles dans une colonne

DEFAULT : pour dire que s’il y a aucune valeur, on remplit alors avec une valeur par défaut dans une colonne.

## 2. 1 Création de table :

**Create Table** Nom\_de\_la\_table\_

**(**

Nom\_de\_1ere\_colonne Type de variables des lignes de la colonne Définition de la contrainte**,**

Nom\_de\_2eme\_colonne Type de variables des lignes de la colonne Définition de la contrainte**,**

**) ;**

## 2. 2 Modifier la structure de la table :

**Alter Table** Cette commande nous permet de :

* D’ajouter une colonne (ADD COLUMN)
* Supprimer une colonne (DROP COLUMN)
* Modifier une colonne (ALTER COLUMN)
* Ajouter une contrainte (ADD CONSTRAINT)
* Supprimer une contrainte (DROP CONSTRAINT)

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

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

Exemple : on peut ajouter une colonne avec type décimal après modifier le type de colonne, ajouter et supprimer les contraintes et suppression de colonne :

\*Ajout de la colonne « type\_produit » : **ALTER TABLE** produit **ADD COLUMN** type\_produit **INTEGER,**

\*Modification type de colonne : **ALTER TABLE** produit **ADD COLUMN** type\_produit **CHAR(20),**

\*Ajout de la contrainte: **ALTER TABLE** produit **ADD CONSTAINT**CK\_type\_produit**CHECK**(type\_produit == fruit)**,**

\*Suppression de la contrainte : **ALTER TABLE** produit **DROP CONSTAINT** CK\_type\_produit

# Chapitre 3 : LMD (LANGAGE DE MANIPULATION DE DONNEES)

Il permet essentiellement d’insérer de nouvelles lignes, les modifier et les supprimer

## 3. 1 INSERT :

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

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

## 3. 2 UPDATE (Mise à jour) :

Cette fonctionnalité modifie des lignes, 1 ligne == 1 ou plusieurs colonnes, 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’, 🡺 ça modifie une cellule

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

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

**WHERE** id = **3** (id= identifiant de la colonne)

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

De la même façon on peut changer toutes les lignes d’une colonne.

## 3.3 DELETE :

Cette commande sert à supprimer une ligne ou plusieurs dans une table.

**DELETE FROM** nom\_table

**WHERE** [conditions]

Conditions : sont des précisions des valeurs qu’il faut supprimer.

**EXEMPLE** : supprimer les valeurs de cellules ayant une age > 18 ans

**DELETE FROM** Etudiant **WHERE** age > 18

## Chapitre 4 : LID (Langage d’Interprétation de Données) :

## 4. 1 : SELECT

La commande **SELECT** permet de ressortir les données pour les traitements, c’est un langage très utilisé par les développeurs (dans les tests unitaires par exemple) pour tester leurs logiciels qu’ils ont développés.

**SELECT** …no de la colonne. **FROM** …nom de la table.

**WHERE [**conditions**]**

C’est une commande d’affichage de lignes et/ou de colonnes.

## 4. 2 : WHERE

C’est un opérateur de sélection, il permet de ressortir les lignes et/ou les colonnes, les valeurs dont on a besoin pour faire de l’analyse par exemple.

## 4.3 RENOMMER LES TABLES, COLONNES, LIGNES AFFICHEES PAR LES REQUETES :

Ca permet de changer le résultat de l’affichages mais pas la BDD,

**SELECT** nom\_table **AS** « nouveau\_donné à la table ».

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

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

**\*TRIS DES RESULTATS :** On peut faire le tri 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** **;**

**SELECT \*** **FROM** table\_nom **WHERE** nom\_colonne **LIKE** **‘**A**%’** (sélectionner les mots de la colonne qui commence par A.

# Chapitre 5 LES JOINTURES :

Les jointures on les appliques sur les tables qui contiennent les clé étrangères pour afficher les différentes tables ensemble.