



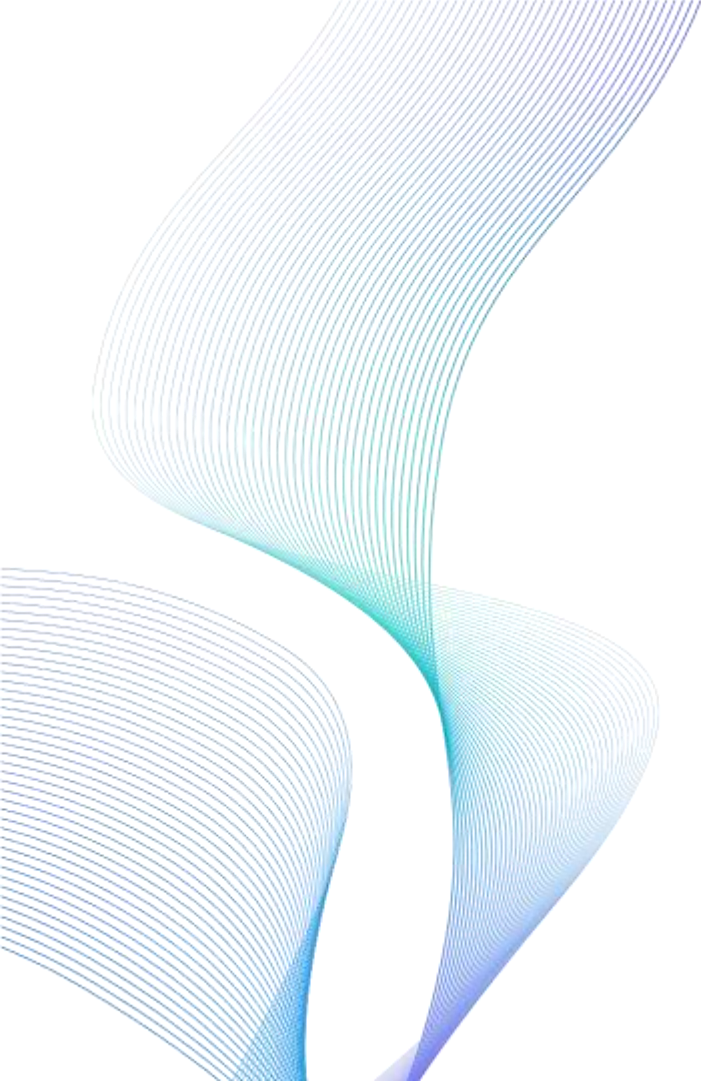
Université Abou Bakr Belkaid – Tlemcen  
Faculté des Sciences de la Nature et de la Vie et Sciences de la Terre et de l'Univers  
Département des Sciences de la Terre et de l'Univers

# Le langage SQL

GHENNANI Hind Selma  
hindselma.ghennani@univ-tlemcen.dz



**SQL**



# SQL

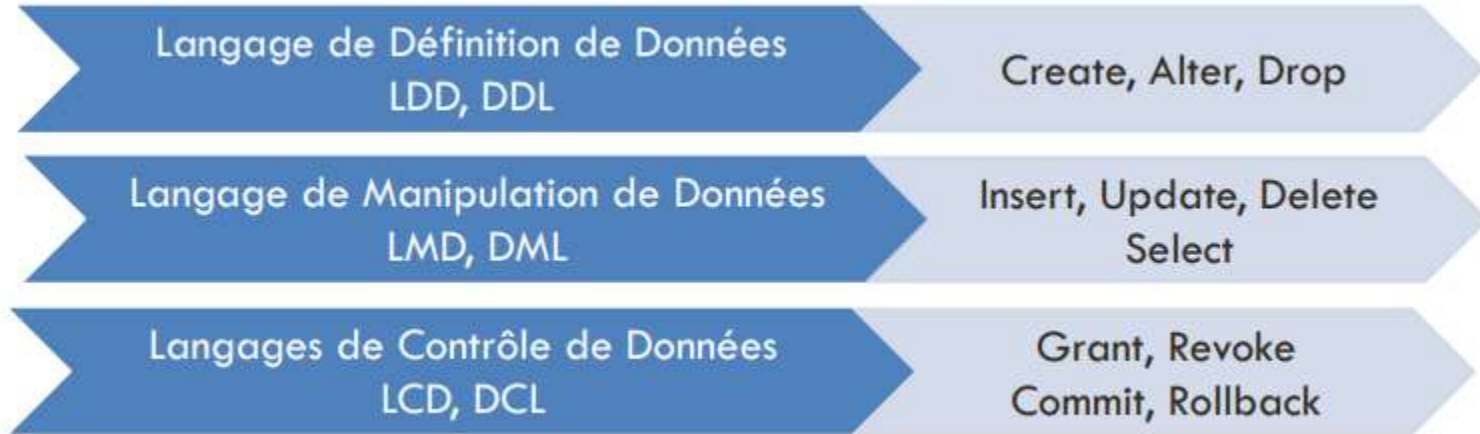
- SQL (Structured Query Language) est un langage complet de gestion de BD relationnelles
- Langage d'Interrogation **Structuré** qui consiste à **définir** , **manipuler** , **contrôler** les BDs
- Une commande s'appelle un **ordre**, une instruction s'appelle une **requête**
- Chaque requête doit terminer par un « ; »
- Une requête peut être utilisée de manière interactive ou incluse dans un programme

# SQL

<b>Date de première version</b>	1974
<b>Auteur</b>	Donald Chamberlin et Raymond Boyce
<b>Développeur</b>	IBM
<b>Dernière version stable</b>	SQL 2011
<b>Paradigme</b>	Déclaratif
<b>Dialectes</b>	SQL 86, SQL 89(SQL1), SQL 92(SQL2), SQL 99(SQL3), SQL 2003, SQL 2008, SQL 2011
<b>Système d'exploitation</b>	Multi-Plateforme

# SQL

Composé de 3 Sous langages :



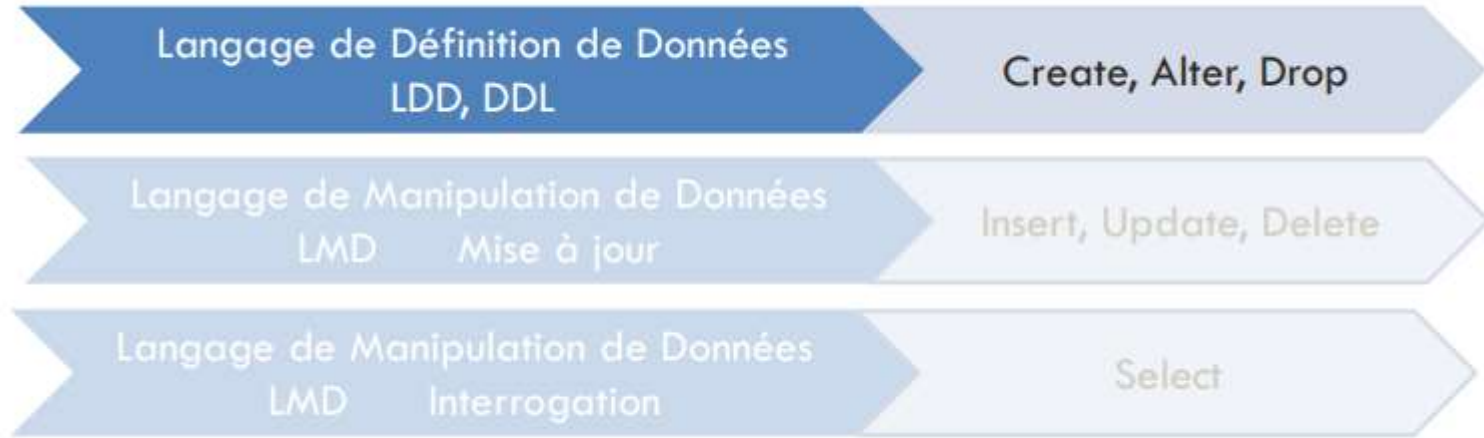


# SQL

- Programme



# SQL - Langage de Définition de Données



# SQL - Langage de Définition de Données

- Création de table

```
CREATE TABLE <Nom Table> (Attribut1 Type [Contrainte_intégrité],  
                             Attribut2 Type [Contrainte_intégrité],  
                             ..... ,  
                             ..... ,  
                             [Contrainte_intégrité], ... ) ;
```

- ✘ La requête peut être exprimée dans une ligne ou étalée sur plusieurs lignes
- ✘ Toutes les valeurs d'une colonne sont de même type
- ✘ « Describe » ou « Desc » permet d'afficher la structure de la table



# SQL - Langage de Définition de Données

- Création de table

## Contraintes d'intégrité

- ✗ NOT NULL
- ✗ DEFAULT
- ✗ UNIQUE
- ✗ CHECK
- ✗ PRIMARY KEY
- ✗ FOREIGN KEY

# SQL - Langage de Définition de Données

- Création de table

## Contraintes d'intégrité

- ✗ **NOT NULL** : Force la saisie d'une colonne (spécifiée implicitement pour les attributs qui font partie de la clé primaire)
- ✗ **DEFAULT** <Valeur> : Précise une valeur par défaut qui est incluse dans tout nouveau tuple qui n'a pas été renseigné
- ✗ **Ex** : **CREATE TABLE** Etudiant

```
(Matricule VARCHAR(3),  
Nom VARCHAR(20) NOT NULL,  
Prenom VARCHAR(15),  
Date_Naiss DATE NOT NULL,  
Annee_Insc Integer DEFAULT 2017 );
```

# SQL - Langage de Définition de Données

- Création de table

## Contraintes d'intégrité

- ✗ **UNIQUE** [(Attribut)] : Toutes les valeurs contenues dans la colonne doivent être uniques au sein de la table (spécifiée implicitement pour la clé primaire)
- ✗ **CHECK** (Condition) : Spécifie les valeurs acceptables pour une colonne
- ✗ **Ex : CREATE TABLE** Etudiant  
(Matricule VARCHAR(3),  
Nom VARCHAR(20) NOT NULL,  
Prenom VARCHAR(15) **UNIQUE**,  
Age INTEGER **CHECK** (Age BETWEEN 17 AND 25),  
Année\_Insc INTEGER **DEFAULT** 2017,  
Ville VARCHAR(15) **CHECK** (Ville IN ('Tlemcen', 'Ain Temouchent')));

# SQL - Langage de Définition de Données

- Création de table

## Contraintes d'intégrité

- ✗ **PRIMARY KEY** : Définition d'une clé primaire simple

```
CREATE TABLE <Nom Table> (Attribut1 Type PRIMARY KEY,  
.....);
```

- ✗ **Ex** : **CREATE TABLE** Etudiant

```
(Matricule VARCHAR(3) PRIMARY KEY,  
Nom VARCHAR(20) NOT NULL,  
Prénom VARCHAR(15) UNIQUE,  
Age INTEGER CHECK (Age BETWEEN 17 AND 25),  
Année_Insc INTEGER DEFAULT 2017,  
Ville VARCHAR(15) CHECK (Ville IN ('Tlemcen', 'Ain Temouchent')) );
```

# SQL - Langage de Définition de Données

- Création de table

## Contraintes d'intégrité

- ✘ **FOREIGN KEY** : Définition d'une clé étrangère

```
CREATE TABLE <Nom Table> (Attribut1 Type,  
Attribut2 Type,  
Att_Clé_Etr Type REFERENCES TablePère (Clé_Primaire),  
.....,  
FOREIGN KEY (Att_Cé_Etr) REFERENCES TablePère (attribut) );
```



# SQL - Langage de Définition de Données

- Création de table

## Contraintes d'intégrité

✘ **FOREIGN KEY** : Définition d'une clé étrangère

✘ **Ex** : **CREATE TABLE** Etudiant (Matricule ...)

**CREATE TABLE** Cours (CodeCours ...)

**CREATE TABLE** Suivre

(Matricule VARCHAR(3) **REFERENCES** Etudiant (Matricule),

CodeCours NUMBER,

**PRIMARY KEY** (Matricule, CodeCours),

**FOREIGN KEY** (CodeCours) **REFERENCES** Cours (CodeCours) );

# SQL - Langage de Définition de Données

- Modification de structure

┆ **ALTER TABLE** <Nom\_Table>

ADD Attribut Type [Contrainte\_intégrité] |

MODIFY/CHANGE Attribut Type [Contrainte\_intégrité] |

RENAME TO Nouveau\_Nom\_Table |

RENAME COLUMN Old\_Name TO New\_Name |

DROP COLUMN Nom-Col ;

✗ **COLUMN** : Facultatif dans plusieurs SGBD

✗ **MODIFY** ou **CHANGE** dans quelques SGBD permet aussi de renommer les colonnes

# SQL - Langage de Définition de Données

- Modification de structure

- ↳ **Ajouter une colonne**

- ```
ALTER TABLE Etudiant ADD Telephone INTEGER NOT NULL;
```

- ↳ **Modifier une colonne**

- ```
ALTER TABLE Etudiant MODIFY Telephone VARCHAR (12);
```

- ↳ **Supprimer une colonne**

- ```
ALTER TABLE Etudiant DROP Année_Insc;
```

- ↳ **Renommer une colonne**

- ```
ALTER TABLE Etudiant RENAME COLUMN Ville TO Adresse;
```

- ↳ **Renommer une table**

- ```
ALTER TABLE Etudiant RENAME TO Enseignant;
```

# SQL - Langage de Définition de Données

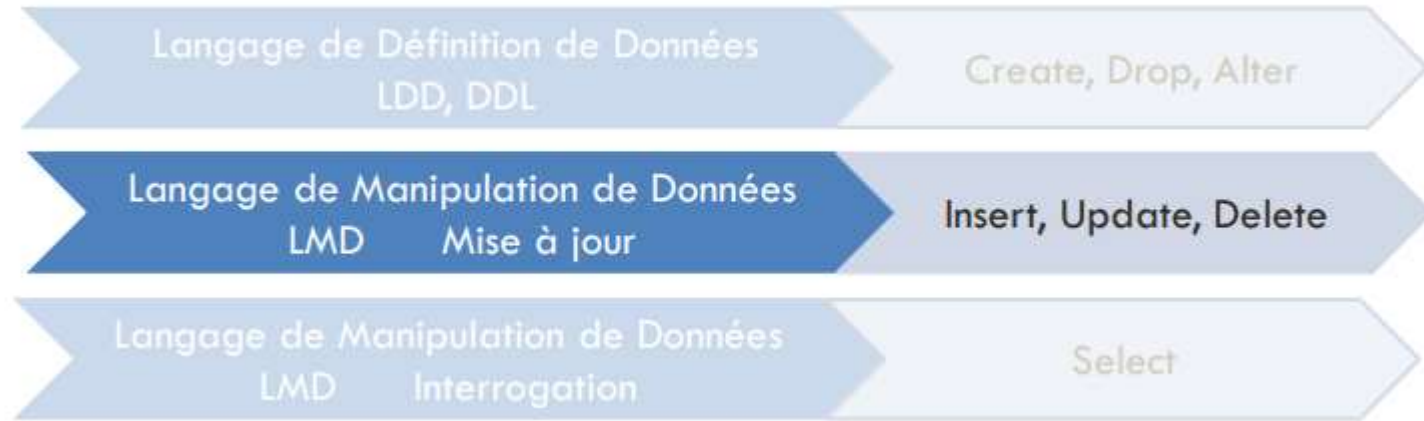
- Suppression de table

**DROP TABLE <Nom\_Table> [CASCADE CONSTRAINTS]**

- ✗ Supprime la table avec son contenu
- ✗ CASCADE CONSTRAINTS : Supprime toutes les contraintes de clé externe référençant cette table
- ✗ Si on tente à détruire une table dont certains attributs sont référencés sans spécifier CASCADE CONSTRAINTS → Le SGBD va refuser

**DROP TABLE** Etudiant ;

# SQL - Langage de Manipulation de Données





# SQL - Langage de Manipulation de Données

## Insertion de données

```
INSERT INTO <Nom Table> [(Attribut1, Attribut2, Attribut3,...)]  
      VALUES      (Valeur1, Valeur2, Valeur3,...), (Valeur1, Valeur2, Valeur3,...), ... ;
```

- ✘ Les noms de colonnes sont facultatifs si on respecte **l'ordre de définition** et **toutes les valeurs de colonnes sont fournies**
- ✘ Les attributs non spécifiés seront NULL ou à la valeur par défaut
- ✘ **Ex :** **INSERT INTO** Etudiant (Nom , Prénom, Age, Année\_Insc) **VALUES** ('SARI', 'RIHAM', 20, 2016);  
**INSERT INTO** Etudiant **VALUES** ('BENSAYAH', 'FATIMA', 21, 2015);  
**INSERT INTO** Etudiant (Age, Prénom, Année\_Insc, Nom) **VALUES** (20, 'AMAL', 2016, 'MAHI');  
**INSERT INTO** Etudiant (Prenom, Age) **VALUES** ('MERYEM', 20);  
**INSERT INTO** Etudiant **VALUES** ('BENSOUNA', Null, 20, Null);

# SQL - Langage de Manipulation de Données

## Insertion de données (Exercice)

**Etudiant (Nom, Prénom, Age, Année\_Insc)** : Parmi ces 7 requêtes, quelles qui sont justes ou fausses ?

1. INSERT INTO Etudiant VALUES ('AMARBENSABEUR', 'NADJWA', '2015'); **Fausse**
2. INSERT INTO Etudiant VALUES ('ANITER', 'HICHAME', 21, 2016), ('ARICHI', '111', 20, 2022), ('ATTAR', '222', 19, 2010), ('BAGHLI', '333', 20, 1920); **Juste**
3. INSERT INTO Etudiant (Prénom, Age) VALUES ('AMINA', 20), ('ISLAM', 20), ('BAHAR', 'AYMEN', 20); **Fausse**
4. INSERT INTO Etudiant VALUES (19, 'ROMAISSA', 2016, 'BELDJILALI'); **Fausse**
5. INSERT INTO Etudiant (Age) VALUES (20); **Juste**
6. INSERT INTO Etudiant VALUES (Null, 'SOUFYANE', Null, Null); **Juste**
7. INSERT INTO Etudiant VALUES (Nom , Prénom, Age, Année\_Insc) ('BENOMARI', Null, Null); **Fausse**