

# Bases de Données 2

## #3 - Gestion d'une BD

Matthieu Nicolas  
Polytech S5 - II

Slides réalisées à partir de celles de Claude Godart et Malika Smaïl

# SQL

- Comporte 3 parties :
  - **Langage de Définition de Données (LDD)** pour créer, modifier et supprimer des définitions de tables (*create table, alter table, drop table*)
  - **Langage de Manipulation de Données (LMD)** pour rechercher, ajouter, modifier et supprimer des données (*select, insert, update, delete*)
  - **Langage de Contrôle de Données (LCD)** pour gérer les droits d'accès aux données (*grant, revoke*)

# Plan

- Définitions de tables en SQL
- Gestion de tuples en SQL

# Définition de tables en SQL

Base de Données 2  
#3

# Exemple

- On re-considère les relations suivantes pour les exemples suivants :

**Produits**(prod\_id, nom, pu)

prod_id	nom	pu
1	A3	10.0
2	crayon	9
3	stylo	15
4	A4	10.0

**Depots**(depot\_id, adr, volume)

depot_id	adr	volume
1	Nancy	100
2	Laxou	200
3	Vandoeuvre	115
4	Nancy	220
5	Nancy	1000

**Stock**(prod\_id, depot\_id, qte)

prod_id	depot_id	qte
1	1	0
3	2	9
1	3	15
2	4	20
3	5	0
1	4	5
2	5	2
3	3	30
1	2	10

# Définition de schémas de relation en SQL

- **CREATE TABLE** : Création de schéma de relation
- **ALTER TABLE** : Modification de schéma de relation
- **DROP TABLE** : Suppression de schéma de relation

# Création d'un schéma de relation

- Permet de définir :

- Noms des attributs
- Types des attributs
- Contraintes d'intégrité

```
CREATE TABLE <nom table> (  
    <nom attribut 1> <type attribut 1> [...],  
    <nom attribut 2> <type attribut 2> [...],  
    ...  
)
```

# Types

- **VARCHAR**(*size*) : chaîne à taille variable (0 à *size* caractères)
- **INT**(*size*) : entier signé (-2147483648 à 2147483647) ou non-signé (0 à 4294967295). *size* ne sert qu'à **gérer l'affichage**
- **FLOAT**(*p*) : nombre décimal. *p* représente la **précision** du nombre (0 à 53)
- **DATE** : date au format YYYY-MM-DD
- Plus d'infos : [https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)



# Création d'un schéma de relation - exemple

```
CREATE TABLE produits (  
    prod_id INT,  
    nom VARCHAR(30)  
)
```



**produits**(prod\_id, nom)

prod_id	nom

# Contraintes d'intégrité

- Règles qui définissent la cohérence d'une donnée ou d'un ensemble de données
- Contraintes de base...
  - **non nullité** d'un attribut
  - **valeurs restreintes** pour un attribut
  - **unicité** de la valeur d'un attribut ou groupe d'attributs
  - **clé primaire**

# Création avec contraintes d'intégrité - exemple

```
CREATE TABLE depots (  
    depot_id INT PRIMARY KEY  
        AUTO_INCREMENT,  
    adr VARCHAR(40) NOT NULL,  
    volume INT CHECK(volume >= 0)  
);
```

**depots**(depot\_id, adr, volume)

depot_id	adr	volume

# Disclaimer - CHECK

- **CHECK** est parsé et accepté par MySQL...
- ... Mais pas pris en compte jusqu'à MySQL 8.0.16
- Voir : <https://dev.mysql.com/doc/refman/8.0/en/create-table-check-constraints.html>

# Clés étrangères - 1

- **Permettent de relier les différentes tables** de notre base de données
- Aussi appelées **contraintes de référence**

# Création avec clés étrangères - exemple

**CREATE TABLE** stocks (

prod\_id **INT**,

depot\_id **INT**,

qte **INT DEFAULT 0**,


**PRIMARY KEY** (prod\_id, depot\_id),

**FOREIGN KEY** (depot\_id) **REFERENCES** depots(depot\_id),

**FOREIGN KEY** (prod\_id) **REFERENCES** produits(prod\_id)

)

**stocks**(prod\_id, depot\_id, qte)



prod_id	depot_id	qte

# Clés étrangères - 2

[CONSTRAINT nom\_constr]

FOREIGN KEY (nom\_attr) REFERENCES nom\_table(nom\_attr)

[ON DELETE mode\_reference]

[ON UPDATE mode\_reference]

- Permet de **spécifier le comportement** à adopter pour les tuples de la table **en cas de suppression ou de mise à jour** d'une valeur de la clé étrangère **dans sa table d'origine**
- Plusieurs modes de références

# Clés étrangères - 3

- **RESTRICT/NO ACTION** : rien ne se passe, requête de mise à jour ou de suppression **annulée**
- **CASCADE** : les **tuples** concernés sont **modifiés/supprimés**
- **SET NULL** : les **attributs** des tuples concernés sont **mis à la valeur NULL**
  - **Incompatible si l'attribut** fait partie de la **clé primaire** de la table
- **SET DEFAULT** : les **attributs** des tuples concernés sont mis à la **valeur définie par défaut**



# Modification d'un schéma de relation

- Permet de :
  - Ajouter/modifier/retirer des attributs
    - `ALTER TABLE <nom table>`
      - `[ADD <nom_attr> <type> <contraintes>]`
      - `[ADD CONSTRAINT <nom_constr> <constr>]`
    - Ajouter/modifier/retirer des contraintes d'intégrité
      - `[CHANGE <nom_attr> <nouveau_nom_attr> <nouveau_type> <nouvelles_contraintes>]`
      - `[DROP <nom_attr>]`
      - `[DROP CONSTRAINT <nom_constr>]`

# Modification d'un schéma de relation - exemple

```
ALTER TABLE produits
```

```
ADD pu FLOAT;
```

```
ALTER TABLE produits
```

```
ADD CONSTRAINT PK_produits
```

```
PRIMARY KEY (prod_id)
```



**produits**(prod\_id, nom, pu)

prod_id	nom	pu

```
ALTER TABLE produits
```

```
CHANGE prod_id prod_id INT AUTO_INCREMENT
```

# Suppression d'un schéma de relation

**DROP TABLE** <nom table> [, <nom table, ...]

- Suppression du **schéma** de la relation
- Suppression des **tuples** de la relation
- Suppression des **index** de la relation
- **Exemple** : **DROP TABLE** stocks, depots, produits

# Gestion de tuples en SQL

Base de Données 2  
#3

# Insertion de tuples

- Insertion d'un seul tuple

**INSERT INTO** produits **VALUES** (5, "A1", 23)

**INSERT INTO** produits(nom, pu) **VALUES** ("A1", 23)

- Insertion d'un ensemble de tuples

**CREATE TABLE** papiers(prod\_id **INT**, nom **VARCHAR**(30), pu **FLOAT**);

**INSERT INTO** papiers **SELECT** \* **FROM** produits **WHERE** nom **LIKE** "A%"

# Modification de tuples

```
UPDATE produits  
SET pu = 88, nom = "stylo++"  
WHERE nom = "stylo"
```

```
UPDATE produits  
SET pu = pu * 1.1
```

```
UPDATE stocks  
SET qte = qte + 10  
WHERE prod_id IN (  
    SELECT prod_id  
    FROM produits  
    WHERE nom = "crayon"  
)
```

# Suppression de tuples

**DELETE** [**FROM**] <nom table> [, <nom table, ...>]  
[**WHERE** <condition>]

- Supprimer tous les tuples de *produits* ?

**DELETE** produits

- Supprimer le produit ayant pour identifiant 5 ?

**DELETE** produits **WHERE** prod\_id = 5

# Références

- Documentation MySQL
  - <https://dev.mysql.com/doc/>
- Listes des commandes avec exemples
  - <https://www.w3schools.com/sql/default.asp>



Des questions ?