

Centre universitaire Mila  
département MI  
2eme année Master STIC

## Base de Données Avancées

Par SELMANE S

# Rappel sur les Bases de Données

# Base de Données

- **Base de données** Une base de données informatique est un **ensemble de données** qui ont été stockées sur un support informatique, et **organisées et structurées** de manière à pouvoir facilement consulter et modifier leur contenu.

# Systeme de Gestion de Base de Données

- **SGBD** Un Systeme de Gestion de Base de Données (SGBD) est un **logiciel** (ou un ensemble de logiciels) permettant de **manipuler les données d'une base de données.**

# Langage SQL

- **Le langage SQL (Structured Query Language)**  
est un langage de requête utilisé pour interroger des bases de données exploitant le modèle relationnel.

# Le Modèle Relationnel

- Une base des données **relationnelle** est une base des données où chaque objet est représenté par **une relation(table)**
- Les relations sont reliées par ce qu'on appelle **clé étrangère** (Ceci permet de minimiser la redondance des données)

# Formes Normales

## 1ère Forme normale (1FN) :

toutes les données sont atomiques

## 2ème Forme normale (2FN) :

1FN + un attribut non clé ne dépend pas d'une partie de la clé

## 3ème Forme normale (3FN) :

2FN + un attribut non clé ne dépend pas d'un ou plusieurs attributs ne participant pas à la clé

# Opérateurs Relationnels

**Projection** : supprimer des attributs d'une table et éliminer les tuples en double

**Restriction** : supprimer les tuples d'une table ne satisfaisant pas une condition précisée

**Jointure** : faire le produit cartésien puis supprimer les tuples d'une table ne satisfaisant pas une condition précisée

**Union** : fusionner deux tables ayant le même schéma et éliminer les tuples en double

**Différence relationnelle** : soustraire une table (ses tuples) d'une autre ayant le même schéma

**Intersection** : prendre les tuples communs de deux tables ayant le même schéma

# Les droits dans SQL

**SELECT:** privilège qu'il faut posséder pour lire une table

**INSERT, DELETE, UPDATE:** privilèges nécessaires pour mettre à jour une table.

**INSERT(X), UPDATE(X):** privilège nécessaire pour insérer, mettre à jour l'attribut X.

# Octroi et retrait de privilèges

**GRANT** privilège ON objet TO utilisateur [WITH GRANT OPTION]

**REVOKE** [GRANT OPTION FOR] privilège ON objet FROM utilisateur RESTRICT | CASCADE

# Exemples

- GRANT ALL ON TABLE resultat TO directeur WITH GRANT OPTION;
- GRANT INSERT ON TABLE resultat TO sec\_1;
- GRANT SELECT, UPDATE(points) ON TABLE resultat TO prof\_1;

# Types de données

**Char(n)** : chaîne de caractères de taille fixe  $n$

**Varchar(n)** : chaîne de caractères de taille variable mais inférieure à  $n$

**Int** : Entier (un sous ensemble fini des entiers, dépend de la machine)

**Smallint** : Entier. Sous ensemble de int

**Numeric(p,d)** : Réel codé sur  $p$  digits et max  $d$  digits pour partie à droite de la décimale.

**Real** : Un réel flottant.

**Date** : YYYY-MM-DD (année, mois, jours)

**Time** : HH :MM :SS (heure, minute, seconde)

# Langage SQL

## **Création de base des données (MySQL):**

Create database <nom bdd>

Create database gestion;

## **Suppression de base des données (MySQL):**

DROP DATABASE <nom bdd>;

DROP DATABASE gestion;

## Changer la base des données (MySQL):

Use <nom bdd>

Use gestion

# Création des tables

On utilise la clause CREATE TABLE

```
CREATE TABLE compte (  
  Num int NOT NULL,  
  Solde int,  
  NomTit varchar( 20))
```

## Rajout de contraintes

```
CREATE TABLE compte (  
  Num int NOT NULL,  
  Solde int DEFAULT 0,  
  NomTit varchar(20), PRIMARY KEY (Num), CHECK (Num > 1) )
```

# Suppression de table

La commande DROP TABLE permet de supprimer une table.

**DROP TABLE compte.**

Si une vue est définie sur la table compte alors il faut utiliser

**DROP TABLE compte CASCADE**

# Modification de table

La commande **ALTER TABLE** permet de modifier le schéma d'une relation.

Exemple :

## Ajout d'une colonne

```
ALTER TABLE compte ADD Date_ouverture date
```

## Suppression d'une colonne

```
ALTER TABLE compte DROP Solde CASCADE
```

## **Modification d'une colonne**

alter table compte

modify (NomTit varchar(30))

# Clé étrangère

Soient Personne (NSS, Nom) et Voiture (Matricule, modèle, Proprio).

« Proprio » correspond au NSS du propriétaire. C'est une *clé étrangère* dans le schéma Voiture car c'est une clé dans un autre schéma.

# Clé étrangère

```
CREATE TABLE Voiture (  
  Matricule CHAR(8), Modele CHAR(10),  
  Proprio CHAR(3), PRIMARY KEY(Matricule),  
  FOREIGN KEY(Proprio) REFERENCES  
  Personne(NSS)  
  ON [DELETE | UPDATE]  
  CASCADE | RESTRICT | SET NULL  
)
```

# Clé étrangère

**CASCADE:** Si une personne est supprimée, alors les voitures qu'elle possède sont supprimées.

**RESTRICT :** Le système refuse la suppression d'une personne s'il y a des voitures qui lui sont rattachées. C'est l'option par défaut.

**SET NULL:** Si une personne est supprimée, alors l'attribut Proprio prend la valeur NULL.

# Langage de Manipulation de Données

Trois commandes SQL permettant d'effectuer les trois types de modification des données :

- ajout de lignes
- mise à jour de lignes
- suppression de lignes

# Insertion

- Syntaxe :

**INSERT INTO** nomtable (col1,..., coln )

**VALUES** (val1,...,valn )

- Ou

**INSERT INTO** nomtable (val1,..., valn )

# Modification

- Syntaxe:

**UPDATE** nomtable

**SET** col1 = exp1, col2 = exp2, ...

**WHERE** prédicat

# Suppression

- L'ordre DELETE permet de supprimer des lignes d'une table.
- Syntaxe:

**DELETE FROM** nomtable

**WHERE** prédicat

# La clause SELECT

La clause SELECT correspond à la projection de l'algèbre.

Les titres des films :

```
SELECT Titre FROM film
```

L'utilisation de l'astérisque permet de sélectionner tous les attributs :

```
SELECT * FROM film
```

SQL autorise par défaut les doublons. Pour le forcer à les éliminer, on utilise la clause DISTINCT

```
SELECT DISTINCT Titre FROM film
```

SELECT peut contenir des expressions arithmétiques ainsi que le renommage d'attributs :

```
SELECT Prix_HT * 1.206 AS Prix TTC FROM produit
```

# La clause WHERE

Elle correspond au prédicat de sélection dans l'algèbre. La condition porte sur des attributs des relations qui apparaissent dans la clause FROM

```
SELECT DISTINCT Titre FROM film WHERE  
Réalisateur "Bergman" AND Acteur = "Stewart"
```

SQL utilise les connecteurs **AND, OR et NOT**.  
Pour simplifier la clause WHERE, on peut  
utiliser la clause **BETWEEN**.

```
SELECT Num FROM compte WHERE Solde  
BETWEEN 0 AND 10000
```

# La clause FROM

Elle correspond au produit cartésien de l'algèbre.

Le titre et le réalisateur des films programmés cinéma UGC.

```
SELECT Titre, Réalisateur
```

```
FROM film, programme WHERE film.titre =  
programme.titre AND programme.NomCiné =  
"UGC"
```

# La clause ORDER BY

SQL permet de trier les résultats de requête

```
SELECT *
```

```
FROM programme
```

```
WHERE NomCiné="UGC"
```

```
ORDER BY Horaire ASC, Titre DESC
```

# Opérateurs SQL

**NOT** : inverse la valeur de tout autre opérateur booléen

**IS NULL** : vérifie si une valeur est NULL ou non

**ALL** : compare une valeur à toutes les valeurs d'un ensemble

**ANY** : vérifie si n'importe quel élément d'un ensemble est TRUE

**SOME** : vérifie si certains éléments d'un jeu de comparaisons sont TRUE

**EXISTS** : vérifie si le résultat d'une sous-requête contient des lignes

**IN** : vérifie si l'opérande est égal à un élément d'une liste d'expressions

**LIKE** : vérifie si l'opérande correspond à un modèle utilisant des métacaractères ( %, \_ ).

# Opérateurs ensemblistes

SELECT ... ..

UNION/ INTERSECT/ EXCEPT

SELECT ...

# Les fonctions d'agrégats

Ce sont des fonctions qui agissent sur des ensembles (multi-ensembles) de valeurs :

**AVG** : la valeur moyenne de l'ensemble

**MIN** : la valeur minimale

**MAX** : la valeur maximale

**SUM** : le total des valeurs de l'ensemble

**COUNT** : le nombre de valeur dans l'ensemble

# La clause group by et having

SQL permet de grouper les résultats de requête

Si on a un condition concernant le groupage en utilise la clause having

```
Select nom, prenom, count(*) from etudiant
```

```
Where note >= 10
```

```
Group by specialite
```

```
Having count(*) > 20
```