

# SYSTÈME DE GESTION DE BASES DE DONNÉES

(SGBD)

## Table des matières

I) Définition.....	1
A) Qu'est-ce qu'une base de données ?.....	1
B) Qu'est-ce qu'un Système de gestion des bases de données (SGBD) ?.....	1
C) Relation.....	2
D) clé étrangère.....	3
E) Clé étrangère.....	4
II) Le langage SQL.....	4
A) définir une relation.....	4
B) coder en SQL.....	4

## I) Définition

### ***A) Qu'est-ce qu'une base de données ?***

Une base de données est un ensemble structuré d'informations, organisé pour être facilement consulté, modifié et mis à jour.

Exemple : un fichier contenant les informations des élèves d'un lycée (nom, prénom, classe, notes).

Il existe différents types de bases de données, par exemple, les bases de données hiérarchiques, les bases de données objet, les bases de données nosql ou bien encore les bases de données relationnelles. Les bases de données relationnelles sont le plus utilisées au monde, c'est ce type de base de données qui seront étudiées.

### ***B) Qu'est-ce qu'un Système de gestion des bases de données (SGBD) ?***

Un Système de Gestion de Base de Données (SGBD) est un logiciel qui permet :

- de créer une base de données
- de stocker les données
- de gérer les accès et les droits
- d'interroger et de modifier les données (via un langage comme SQL)

### C) Relation

La notion de relation est au cœur des bases de données relationnelles. Une relation peut être vue comme un tableau à 2 dimensions, composé d'un en-tête et d'un corps. Le corps est lui-même composé de t-uplets (lignes) et d'attributs (colonnes). L'en-tête contient les intitulés des attributs, le corps contient les données proprement dites. À noter que l'on emploie aussi le terme "table" à la place de "relation".

id	titre	auteur	ann_publi	note
1	1984	Orwell	1949	10
2	Dune	Herbert	1965	8
3	Fondation	Asimov	1951	9
4	Le meilleur des mondes	Huxley	1931	7
5	Fahrenheit 451	Bradbury	1953	7
6	Ubik	K.Dick	1969	9
7	Chroniques martiennes	Bradbury	1950	8
8	La nuit des temps	Barjavel	1968	7
9	Blade Runner	K.Dick	1968	8
10	Les Robots	Asimov	1950	9
11	La Planète des singes	Boulle	1963	8
12	Ravage	Barjavel	1943	8
13	Le Maître du Haut Château	K.Dick	1962	8
14	Le monde des Â	Van Vogt	1945	7
15	La Fin de l'éternité	Asimov	1955	8
16	De la Terre à la Lune	Verne	1865	10

Voici un exemple de relation .

Autre contrainte très importante dans les bases de données relationnelles, une relation ne peut pas contenir 2 t-uplets identiques. Ce qui est la raison pour laquelle on définit une clé primaire .

Ici, les attributs de cette relation sont « id », « titre », « auteur », « ann\_publi », « note ».

« id » est la clé primaire de cette relation.

Une clé primaire est un attribut dont la valeur permet d'identifier de manière unique un t-uplet de la relation. Autrement dit, si un attribut est considéré comme clé primaire, on ne doit pas trouver dans toute la relation 2 fois la même valeur pour cet attribut.

#### **D) clé étrangère**

id	titre	nom_auteur	prenom_auteur	date_nai_auteur	langue_ecriture_auteur	ann_public	note
1	1984	Orwell	George	1903	anglais	1949	10
2	Dune	Herbert	Frank	1920	anglais	1965	8
3	Fondation	Asimov	Isaac	1920	anglais	1951	9
4	Le meilleur des mondes	Huxley	Aldous	1894	anglais	1931	7
5	Fahrenheit 451	Bradbury	Ray	1920	anglais	1953	7
6	Ubik	K.Dick	Philip	1928	anglais	1969	9
7	Chroniques martiennes	Bradbury	Ray	1920	anglais	1950	8
8	La nuit des temps	Barjavel	René	1911	français	1968	7
9	Blade Runner	K.Dick	Philip	1928	anglais	1968	8
10	Les Robots	Asimov	Isaac	1920	anglais	1950	9
11	La Planète des singes	Boulle	Pierre	1912	français	1963	8
12	Ravage	Barjavel	René	1911	français	1943	8
13	Le Maître du Haut Château	K.Dick	Philip	1928	anglais	1962	8
14	Le monde des Â	Van Vogt	Alfred Elton	1912	anglais	1945	7
15	La Fin de l'éternité	Asimov	Isaac	1920	anglais	1955	8
16	De la Terre à la Lune	Verne	Jules	1828	français	1865	10

Ci-dessus ont été ajoutés 3 attributs différents.

Ici il y a plusieurs d'informations dupliquées, par exemple, on retrouve 3 fois "K.Dick Philip 1928 anglais", de même pour "Asimov Isaac 1920 anglais"... Cette duplication est-elle indispensable ?

Non ! Est-elle souhaitable ? Non plus ! En effet, dans une base de données, on évite autant que possible de dupliquer l'information (sauf à des fins de sauvegarde, mais ici c'est toute autre chose).

Si autant de données sont dupliquées inutilement, cela témoigne d'une mauvaise structure de

relation.

Mais alors, comment faire pour avoir aussi des informations sur les auteurs des livres ?

### ***E) Clé étrangère***

Cette notion est simple à comprendre. La seule chose importante à savoir est qu'une clé étrangère est un attribut qui correspond à la clé primaire d'une relation autre. Ce qui permet d'attribuer simplement une valeur numérique à une chaîne de caractères (par exemple)

## **II) Le langage SQL**

### ***A) définir une relation***

Cette section sera complète mais non inutilement surcomplète. Elle traitera essentiellement les lignes de code à connaître en SQL pour le baccalauréat.

Il est IMPORTANT de connaître la définition d'un schéma relationnel.

On schématisera par un exemple :

LIVRES(id : INT, titre : TEXT, #id\_auteur : INT, ann\_publi : INT, note : INT)

Ici on a plusieurs informations importantes :

- INT/TEXT représente le type d'attribut. (INT pour un entier numérique, TEXT pour une chaîne de caractères. Ce ne sont que des exemples.)
- id, titre, id\_auteur et ann\_publi sont des clés.
- le soulignage d'une clé annonce qu'il s'agit d'une clé primaire
- le dièse (#) annonce qu'il s'agit d'une clé étrangère.

### ***B) coder en SQL***

On utilisera le tableau du I)D), qu'on appellera relation LIVRES

Pour afficher un attribut, de tous les t-uplets de la relation, par exemple titre, on utilise :

**SELECT titre**

**FROM LIVRES**

en outre, on peut aussi imposer une condition. Par exemple les titres des livres écrits par Bradbury :

```
SELECT titre  
FROM LIVRES  
WHERE nom_auteur = « Bradbury »
```

on peut aussi choisir d'afficher plusieurs attributs avec de multiples conditions :

```
SELECT titre, ann_publi, note  
FROM LIVRES  
WHERE langue_ecrite_auteur = « anglais »
```

Cela affichera la date de publication , le titre et la note des ouvrages écrits initialement en anglais.

On peut aussi changer la valeur d'un attribut d'un t-uplet :

```
UPDATE LIVRES  
SET note = 0  
WHERE ann_publi = 1968
```

ici on donne une note de 0 aux livres écrits en 1968.

on peut également faire des liens entre plusieurs relations. On donnera ici une méthode de faire, la plus simple. La notion de jointure est souvent inutile quand on connaît la méthode qu'on abordera maintenant.

Soit une relation VIE\_AUTEUR qui comporte le nombre d'enfants de chacun des auteurs des livres, et leur nombre d'appartements. Soit une autre relation ANNEE qui comporte la liste de tous les livres écrits entre 1950 et 1980 et leur titre.

on peut écrire :

```
SELECT nb_enfants, liste_livres, note  
FROM VIE_AUTEUR as v, ANNEE as a, LIVRES as l  
WHERE nb_enfants=2  
WHERE liste_livres = « la belle et la bête »  
WHERE note = 8
```

Attention : il s'agit ici d'un exemple loin de rigoureux de l'utilisation de « as ».

Il est aisé de constater qu'il s'agit d'avoir une « variable »(terme non rigoureux) qui va prendre pour valeur une relation.

Cela conclut ce cours.

