

Introduction aux bases de données

Sommaire

I.	Généralités.....	2
1.	Notion de base de données.....	2
2.	Modèles de base de données.....	2
3.	Système de gestion de base de données (SGBD)	4
II.	Modèle relationnel	5
1.	Présentation	5
2.	Notions de base (table, relation; attribut; clés; domaine; ...) ;	5
3.	Conception de schémas d'une base de données	7

I. Généralités

1. Notion de base de données

Il est difficile de donner une définition exacte de la notion de base de données. Une définition très générale pourrait être :

Définition 1 :

Base de données : *Un ensemble organisé d'informations avec un objectif commun.*

Peu importe le support utilisé pour rassembler et stocker les données (papier, fichiers, etc.), dès lors que des données sont rassemblées et stockées d'une manière organisée dans un but spécifique, on parle de base de données.

Plus précisément, on appelle base de données un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données). Bien entendu, dans le cadre de ce cours, nous nous intéressons aux bases de données informatisées.

Définition 2 :

Base de données informatisée : *Une base de données informatisée est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur (disque dur, cd-rom, clé USB, ...), représentant des informations du monde réel et pouvant être interrogées et mises à jour par plusieurs utilisateurs.*



Avantages d'utilisation des bases de données :

- **stocker** de **gros volumes** d'informations
- **partager** des informations par une communauté de personnes
- **gérer l'accès** à ces informations
- gérer des **informations cohérentes et non-redondantes**

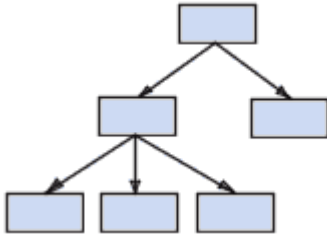
Une fois la base de données spécifiée, on peut y insérer des données, les récupérer, les modifier et les détruire. C'est ce qu'on appelle manipuler les données. Les données peuvent être manipulées non seulement par un Langage spécifique de Manipulation des Données (LMD), mais aussi par des langages de programmation classiques.

2. Modèles de base de données

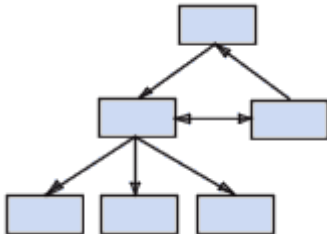
Un modèle de données est un ensemble de concepts et de règles de composition de ces concepts permettant de décrire des données.

Il existe principalement quatre types de modèles :

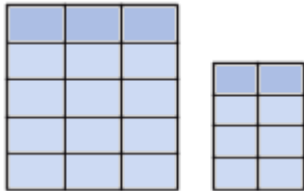
- **Le modèle hiérarchique** : les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de BD



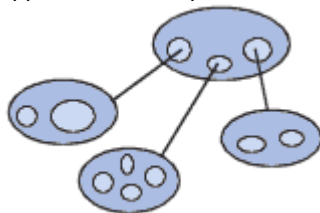
- **Le modèle réseau** : comme le modèle hiérarchique ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant



- **Le modèle relationnel** : les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations



- **Le modèle objet** : les données sont stockées sous forme d'objets, c'est-à-dire de structures appelées classes présentant des données membres. Les champs sont des instances de ces classes



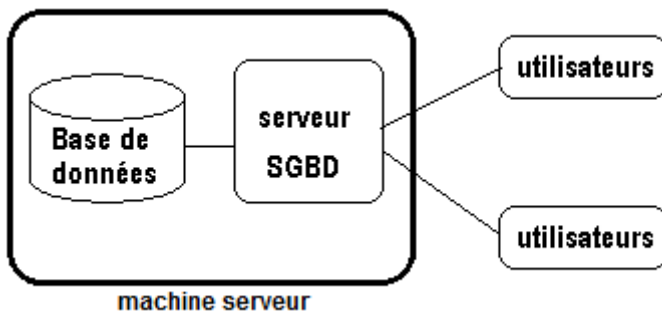
A la fin des années 90 les bases relationnelles sont les bases de données les plus répandues (environ trois quarts des bases de données).

3. Système de gestion de base de données (SGBD)

Un SGBD est un ensemble de logiciels chargés d'assurer les fonctions minimales suivantes :

- Le maintien de la cohérence des données entre elles,
- le contrôle d'intégrité des données accédées,
- les autorisations d'accès aux données,
- les opérations classiques sur les données (consultation, insertion, modification, suppression)

Actuellement, la plupart des SGBD fonctionnent selon un mode client/serveur. Le serveur (sous-entendu la machine qui stocke les données) reçoit des requêtes de plusieurs clients et ceci de manière concurrente. Le serveur analyse la requête, la traite et retourne le résultat au client.



Quelques SGBD connus et utilisés :

- MS Access
- Microsoft SQL
- MySQL
- PostgreSQL
- Oracle
- IBM DB2
- Sybase
- Informix

II. Modèle relationnel

1. Présentation

Le modèle relationnel est fondé sur la théorie mathématique de l'algèbre relationnelle. Il permet de produire une représentation simple des données sous forme de tables constituées de lignes et de colonnes.

La manipulation des données se fait selon le concept mathématique de relation de la théorie des ensembles, c'est-à-dire l'algèbre relationnelle. L'algèbre relationnelle a été inventée en 1970 par E.F. Codd, le directeur de recherche du centre IBM. Elle est constituée d'un ensemble d'opérations formelles sur les relations. Les opérations relationnelles permettent de créer une nouvelle relation (table) à partir d'opérations élémentaires sur d'autres tables (par exemple l'union, l'intersection, ou encore la différence).

En termes savants, Codd voulait assurer l'indépendance entre l'organisation logique des données et la technique informatique utilisée pour les stocker. En termes simples, il cherchait une méthode permettant de stocker des données (structurées) de toute nature, sans recourir chaque fois à de la programmation spécifique. Codd est considéré comme le créateur de l'algèbre relationnelle (l'aspect théorique des bases de données), qui utilise la théorie des ensembles.

2. Notions de base (table, relation; attribut; clés; domaine; ...) :

- Domaine :

Un domaine est un ensemble de valeurs caractérisé par nom.

Exemple :

L'ensemble $N = \{0, 1, 2, \dots\}$ est le domaine des entiers.

L'ensemble couleurs = {blanc, rouge, vert, jaune, noir, ...} est le domaine des couleurs.

- Produit cartésien :

Rappelons que le produit cartésien d'un ensemble de domaines D_1, D_2, \dots, D_n est l'ensemble des vecteurs $\langle v_1, v_2, \dots, v_n \rangle$ où, pour i variant de 1 à n , v_i est une valeur de D_i

Exemple :

$D_1 = \{\text{Pierre, Paul, Jacques}\}$ et $D_2 = \{1, 2\}$

$D_1 \times D_2 =$

Pierre	1
Pierre	2
Paul	1
Paul	2
Jacques	1
Jacques	2

- **Relation :**

Une relation est un sous ensemble nommée du produit cartésien d'une liste de domaines.

- Exemple :

$D1 = \{\text{Youssef, Rachid, Nabil}\}$

$D2 = \{1,2,3, \dots, 20\}$

On peut composer la relation " Notes " :

Youssef	8
Rachid	13
Nabil	6

Une relation peut être vue comme un tableau à 2 dimensions dont les colonnes correspondent aux domaines et les lignes contiennent les tuples.

Une relation est parfois appelée " table ".

- **Attribut :**

Un attribut est une colonne d'une relation caractérisée par un nom.

Exemple :

Nom	Moyenne
Youssef	8
Rachid	13
Nabil	6

- **Schéma de relation :**

Un schéma de relation est le nom de la relation suivi de la liste des attributs avec leurs domaines.

Représentation textuelle :

Notes (Nom : D1, Moyenne : D2)

Afin de simplifier, on ne précise en général pas les domaines :

Notes (Nom, Moyenne)

Représentation graphique :

Notes
Nom : D1 Moyenne : D2

- **Schéma de base de données**

Un schéma de base de données relationnel S est un ensemble de schémas de relation

$S = \{R_1, R_2, \dots, R_p\}$ et un ensemble de contraintes d'intégrité.

Les contraintes d'intégrité :

- Contraintes d'unicité
- Contraintes de références

3. Conception de schémas d'une base de données

- **Les contraintes d'unicité :**

Par définition, une relation est un ensemble de tuples. Un ensemble n'ayant pas d'élément en double, il ne peut exister deux fois le même tuple dans une relation. Afin d'identifier les tuples d'une relation sans donner toutes les valeurs et d'assurer simplement l'unicité des tuples, la notion de clé est utilisée.

Notion : Clé (Key)

Ensemble minimal d'attributs dont la connaissance des valeurs permet d'identifier un tuple unique de la relation considérée.

De manière plus formelle, une clé d'une relation R est un ensemble d'attributs K tel que, quels que soient les tuples t1 et t2 d'une instance de R, $t_1(K) \neq t_2(K)$, c'est-à-dire que t1 et t2 ont des valeurs de K différentes.

- **Contraintes de références :**

Les **clés étrangères**, permettent de gérer des **relations entre plusieurs tables**, et garantissent la **cohérence des données**.

Exemple de deux tables professeurs et matières, avec chaque professeurs enseigne une seule matière, l'attribut **Code_Mat** est une clé étrangère :

Matières :

Code	Désignation
IF	Informatique
MATH	Mathématique
PH	Physique

Professeurs :

Num	Nom	Code_Mat
1	Khalid	IF
2	Youssef	IF
3	Asmaa	PH
4	Laila	MATH

Exemple de deux tables professeurs et matières, avec chaque professeurs peut enseigner plusieurs matières :

Matières :

Code	Désignation
IF	Informatique
MATH	Mathématique
PH	Physique

Professeurs :

Num	Nom
1	Khalid
2	Youssef
3	Asmaa
4	Laila

Enseigner :

Num	Code
1	IF
1	MATH
2	PH
3	PH