

# Conception de Base de Données

interro shema entite-association : 25%  
clé cours : DB18

## Contents

<b>1 Qu'est ce qu'une base de donnée ?</b>	<b>2</b>
1.1 Historique . . . . .	2
Systèmes transactionnels . . . . .	2
1.2 inconvénients des systèmes de fichier . . . . .	2
1.3 Lien entre les données . . . . .	3
1.4 Caractérisation d'une Base de Données . . . . .	3
1.5 Difficultés d'identifiant des concepts et liens . . . . .	3
1.6 Système de gestion de base de données (S.G.B.D.) . . . . .	3
<b>2 Le modèle entités-associations</b>	<b>4</b>
2.1 Introduction . . . . .	4
2.2 Entités et types d'entités(T.E.) . . . . .	4
2.3 Associations-Type d'associations (T.A) . . . . .	4
Obligatoire et Facultatif . . . . .	5
2.4 Construction d'un Schéma conceptuel . . . . .	5
2.5 Intérêt de prévoir un type d'association plutôt qu'un attribut . . . . .	5
2.6 Attributs . . . . .	5
2.6.1 Identifiant . . . . .	5
2.6.2 Mono/Multi-Valués . . . . .	5
2.6.3 Atomique ou Décomposable . . . . .	6
2.7 Non Redondance dans le schémas entités-association . . . . .	6

# 1 Qu'est ce qu'une base de donnée ?

## 1.1 Historique

Le stockage de données n'est pas nouveau en informatique. On utilisa ensuite des cartes perforées puis des bandes magnétiques pour stocker l'information. Le problème avec ce type de fichiers est que les accès ne peuvent se faire que séquentiellement. Les évolutions techniques ont permis de réduire les temps d'accès, mais pas les accès concurrents aux fichiers pour différentes applications. Ce besoin d'accès concurrent s'est vu obligé de faire appel à un nouveau concept : les **systèmes transactionnels** (*suite d'instructions qui forment un tout*).

Le principe de transaction n'est pas la seule méthode pour accéder à une ressource partagée sans corrompre tout le fichier. On peut notamment bloquer le fichier ou bloquer un seul enregistrement à la fois. Même si ces méthodes sont plus faciles à implémenter, elles restent moins optimisées dans le cadre d'une gestion de base de données.

69' : Apparition de banques de données (faiblement structurées, classés par mots clés et dictionnaires)

70' : IBM lance le SGBD (Système de gestion de Bases de Données)

## Systèmes transactionnels

différentes façon de gérer un fichier par un user

- Bloquer un fichier
- bloquer l'enregistrement sur lequel on écrit
- adopter le principe de transactions
  - suite logique d'instructions considérée comme formant un tout (transfert d'argent (soit les 2 instructions (+1000 -1000) s'effectuent soit aucune))

fichier : lire , modifier , fin de transaction ( 3 copies ) si copie 1 != copie 3 alors erreur car collision

## 1.2 inconvénients des systèmes de fichier

- Dépendance des applications
- Redondance des données
- Risques d'incohérences des données
- Impossibilité d'interroger les données
- Sécurité (auto-sauvegarde)

- Pas de modification de structure possible
- Pas de relation entre les fichiers (et les données dans ces fichiers)

### 1.3 Lien entre les données

Une **Donnée** est un enregistrement dans un code en vue de transmettre ou stocker de l'information.

Une **information** (Subjectif) : sens ou signification que l'on attache ou déduit d'un ensemble de données. Un **Fichier** est un ensemble de données.

Un **objet** du *type livre* à un **lien** avec un **objet** du *type auteur*.

### 1.4 Caractérisation d'une Base de Données

- **Relation** entre les données
- **Sauvegarde** des données sur un support (disque)
- **Partage** des données entre plusieurs utilisateurs
- **Indépendance** des données par rapport aux applications
- **Sans redondance** inutile
- Contrôle de **cohérence**
- Exploitation des données par **interrogations**
- **Longueur** d'enregistrement **variable**
- **Modification** possible de la structure des enregistrements

### 1.5 Difficultés d'identifiant des concepts et liens

Les types d'objets ont des caractéristiques et donnent des types d'enregistrements.

### 1.6 Système de gestion de base de données (S.G.B.D.)

**DBMS** : Database management system

SGBD - programmation interface - user

On peut interroger avec la base de données, en utilisant le langage SQL pour poser des questions (Query).

- \* DQL : Data Query Language : requête d'accès sous la forme déclarative
- \* DDL : Data Definition Language : ajoute et modifie des schémas de BDD
- \* DCL : Data Control Language : Traite les permissions
- \* DML : Data Manipulation Language : Ajoute et modifie des données

Une DataBase est gérée par un Administrateur. Il est responsable de l'ensemble du système :

- Création de la structure de originale de la BDD et organisation fichiers (DDL)
- Modifiacion de la structude de la BD
- Gestion des accès
- Backup et entretien de la BD

Le but d'un SGBD :

- **Accès optimal** à toute donnée
- **Traitement simultané** des données
- **Validité et cohérence** des données
- **Sécurité** ( droit d'accès )
- **Sauvegarde et Récupération**

## 2 Le modèle entités-associations

### 2.1 Introduction

Il existe 3 étapes dans la création d'une base de données. On débute avec un schéma conceptuel (**Entité association**). Puis, un schéma logique (**Shéma conforme ou SGBD**) et on termine avec un schéma physique ou **SQL**

### 2.2 Entités et types d'entités(T.E.)

Une **Entité** est une chose qui existe dans le monde réel, à propos de laquelle on veut enregistrer des informations. Un **Type d'entité** est une abstraction générale de ce qui caractérise plusieurs entités communes.

Ainsi, une entité deviendra une occurrence d'un type d'entité. (ex : Bob est une occurrence de Humain)

> un Type d'entité ne prends pas de 's' ( ex : nom et ~~noms~~ )

On définira donc un Type d'entité par son **Nom** et sa **Liste d'attributs**

### 2.3 Associations-Type d'associations (T.A)

Une **association** est une correspondance, un lien entre 2 ou plusieurs entités.

> Une Occurence d'un T.A. est toujours reliée à une et une seule occurrence de chaque T.E. associé.

- 1 à 1
- 1 à N
- N à N

Section-[inscription]-Etudiant

Une cardinalité est le nombre de liens entre 2 types d'entités.

### Obligatoire et Facultatif

La cardinalité minimum d'un type d'entités donne son caractère obligatoire(1) ou facultatif(0).

## 2.4 Construction d'un Schéma conceptuel

Pour former un schéma, on cherche 2 concepts et un Lien qui les unis

## 2.5 Interêt de prévoir un type d'association plutôt qu'un attribut

Lors de la création d'un schéma, on pourrait penser qu'utiliser un attribut plutôt qu'un type d'association peut être une bonne idée. En pratique, ce n'est jamais le cas. En effet, si vous laissez à vos utilisateurs le soin de rentrer (par exemple leur code de cours) ils se peut que ceux-ci se trompent et indiquent **n'importe quelle chaîne de caractère**. Il n'y a pas non plus de **vérification de cohérence** que peut offrir un SGBD par l'intermédiaire de **contraintes**.

## 2.6 Attributs

Un **Attribut** est défini par un *nom*, un *type* et le *domaine de valeurs admises*.

> Un Attribut Booleen possède 3 valeurs possibles : Vrai, Faux **ET** **NULL** quand on ne sait pas.

### 2.6.1 Identifiant

Un **identifiant** est un type particulier d'*attribut* qui prends une valeur différente pour chaque entité dans la Base de Données.

### 2.6.2 Mono/Multi-Valués

Un *attribut* peut être :

- **Mono-valué** (ex : nom)
- **Multi-valué** (ex : prenom[1..6])

- *Obligatoire* prenom[1..3]
- *Facultatif* prenom[0..3]

### 2.6.3 Atomique ou Décomposable

**Atomique** : Attribut non décomposable **Décomposable** : Attribut que l'on peut décomposer (ex : une adresse)

## 2.7 Non Redondance dans le schémas entités-association

Pour des raisons de cohérence et d'économie sur les disques on évitera toute forme de redondance pour toutes les **données déjà présentes** ou les **données calculables**