# Introduction base de données

## Objectifs :

* Découvrir les principales caractéristiques des bases de données (modèle relationnel / modèle entité relation
* Cours orienté **Conception** -> Être capable d’effectuer une modélisation de donnée puis de créer une base de données équivalente
* Applications TP (SQL)

## Premières définitions :

**Base de données (BD) :** Collection de données cohérentes et structurées. Ensemble cohérent, intégré, partagé de données structurées défini pour les besoins d’une application.

Une base de données est un ensemble de données modélisant les objets d’une partie du réel et servant de support à une application informatique. Elle est utilisée par de nombreuses personnes et son organisation est régie par un **modèle de données**.

**Système de gestion de base de données (SGBD) :** Logiciel(s) assurant structuration, stockage, maintenance, mise à jour et consultation des données d’une BD.

-Un SGBD est un ensemble de logiciels coordonné permettant de décrire, mémoriser, manipuler, traiter et interroger efficacement des données spécifiques dans une grande masse d’informations partagées par plrs utilisateurs.

**PLAN :**

**Chapitre 1 :** Introduction

**Chapitre 2 :** Conception de BD

**Chapitre 3 :** Le modèle Entité-Association

**Chapitre 4 :** Le modèle relationnel

**Chapitre 5 :** Algèbre relationnel et SQL

# Chapitre 1 : Introduction

## Rôle essentiel des BD :

* **Assurer le stockage des données :**
  + Enregistrer sur la mémoire secondaire
  + Garantie de pérennité des données
* **Prendre en compte la structure des données**
* **Permettre des utilisations simultanées et autorisées :**
  + Contrôle d’accès et gestion de la concurrence des opérations
  + Garantie de confidentialité et d’intégrité des données

## Défis des BD aujourd’hui

* Multiplicités des types de données (classiques, multimédia, données, géographique, données semi structurées…)
* Informations incorrectes ou incomplètes. Nous sommes submergés d’info souvent de mauvaise qualité
  + Comment les exploiter avec rigueur et précision ?
  + Comment produire de l’info correcte à partir de données imprécises ou incorrectes
* Volumes et performances – Les volumes des BD explosent (croissance constance), elles sont réparties sur de nbreux serveurs
  + Comment mémoriser, gérer et exploiter ces données en toute sécurité en offrant aux utilisateurs des temps de réponse acceptables ?
* Accès aisé par des non informaticiens
  + Développent de techno, d’interfaces et de compétences pour amener les utilisateurs vers l’autonomie nécessaire
* Données distribuées et nomades
  + Modèle centralisé n’est plus d’actualité
* BD et Web
  + Comment unifier Web et BD
  + Comment interfacer le Web et le BD ?
* Données décisionnelles (BD traditionnelles / BD d’aide à la décision - > Big Data)
* Emergence des BD Objets…

## Cycle de vie d’un BD – 4 phases :

1. **Conception de la base** (schéma conceptuel)
2. **Implantation des données** (schéma logique)
3. **Utilisation** (interrogation, mises à jour)
4. **Maintenance** (correction, évolution)

### Une base de données doit satisfaire 5 principaux critères :

* Une bonne représentation du monde réel (fidélité)

La base de données doit fournir une image fidèle de la réalité avec des informations toujours fiables et à jour.

* Une non-redondance de l’information (unicité de l’info)

L’information contenue dans la base doit être unique d’un point de vue sémantique et d’implantation physique.

* Une indépendance des données par rapport aux traitements

Les données constituent une image du monde réel, les programmes de traitement doivent être conçus par rapport à cette structure.

* La sécurité et la confidentialité des données

La sécurité et la confidentialité des données doivent être assurées.

* La performance des applications

Toute application (requêtes…) doit avoir une réponse dans des temps convenables (optimisation).

La réalisation d’un logiciel est précédée d’une **phase d’analyse**.

Mise en service - Utilisation

Analyse

Réalisation informatique (programmation, tests, validation)

## Phase d’analyse :

Etape 1 : Evaluation des besoins actuels (Entretiens)

Lister en vrac les doc à éditer, les statistiques à obtenir, les éléments à calculer, les informations à stocker.

Etape 2 : Evaluation des besoins futurs

Anticiper :

* Quelles sont les évolutions possibles de la BD
* La taille des rubriques prévue est-elle suffisante ?

Etape 3 : Structuration des informations

Une fois définis et les informations à stocker énumérées, il faut classer les informations par thème (tables).

# Chapitre 2 : Conception de BD :

### Indépendant du SGBD :

* Besoin des utilisateurs
* **Analyse** **conceptuelle** (modélisation des données)
* Schéma conceptuel

### Spécifique à un SGBD :

* **Conception** **logique** (traduction dans le modèle de BD)
* Schéma logique
* **Mise** **en** **place** (selon le SGBD)
* Schéma physique + code interne

## Architecture à 3 niveaux :

Vue externe

Vue externe

Vue externe

Vue externe

Schéma logique

Schéma physique

**Niveau externe**

**Niveau logique**

**Niveau physique**

## Pourquoi a-t-on besoin d’un modèle de données ?

### Pour savoir en quels termes décrire la structure des données :

Tables ? Entités ? Attributs ? Champs ? Etc…

Ex : modèle relationnel

Modèle de données

(Ensemble de concepts + règles d’intégrité)

Schéma de données

Univers réel

## Modélisation des données :

* Principe : Séparer la description des données et manipulation par des programmes.
* Description : Spécification des structures des données et de leurs types.
* Manipulation : Opération d’interrogation, d’insertion, mise à jour, suppression.
* Réalisation : norme ANSI-SPARC.

Univers réel

Analyse/conception

Schéma conceptuel

Traduction

Schéma logique

Implémentation

Schéma physique

## Niveau Conceptuel :

### Description des besoins -> modèle conceptuel

* + Dialogue concepteurs / utilisateurs
  + Indépendant de la solution informatique
  + Prise en compte des contraintes d’intégrité

## Niveau logique :

* Traduction du schéma conceptuel en un schéma logique dans les concepts du modèle utilisé par le SGBD choisi

## Niveau physique :

* Choix de structure de stockage des données par les admins système
* Schéma interne : description des choix d’enregistrement des données dans les fichiers.
* Fait appel à un nouveau modèle, le modèle interne, où les concepts sont ceux de fichier, organisation de fichier, index, chemin d’accès, clé…

## SGBD Architecture Ansi/Sparc d’un SGBD

### Objectifs des SGBD

* Masquer les aspects de stockage :
  + Indépendance logique
  + Indépendance physique
* Gérer efficacement les données
* Faciliter l’extraction et l’ajout d’information
* Optimiser le traitement d’information
* Assurer la sécurité de données
* Eviter les conflits

### Spécificités d’un SGBD

* Très grande quantité de données à gérer
* Besoin d’interroger, maj souvent, rapidement et efficacement ces données
* Contrôler la redondance d’information
* Partage des données / Accès concurrents
* Gérer les autorisations d’accès / Sécurité des données
* Offrir des interfaces d’accès multiples
* Vérifier les contraintes d’intégrité
* Assurer la reprise après panne

### Fonctions des SGBD

* Description des données : *Langage de Définition de Données (LDD)*

Langage de Manipulation de Données (LMD)

}

* Recherche des données
* Maj des données
* Transformation des données
* Contrôle de l’intégrité des données
* Gestion de transactions et sécurité

# Chapitre 3 : Le modèle Entité-Association

**PLAN :**

1. Intro
2. Entité – Propriété – Identifiant
3. Relations (ou associations) entre entités
4. Cardinalité d’une relation
5. Règles de Normalisation
6. Construction du MCD

## I – Introduction

* Le modèle Entité – Relation repose dur une perception du monde réel sous forme d’un ensemble d’objets appelés entités, associés au moyen d’un ensemble de relations.
* Cette modélisation a été développée de façon à faciliter la conception des architectures de BD

Modèle Conceptuel des Données (MCD)

Univers réel

* Le modèle E/A est un Formalisme graphique pour la modélisation de données
* Objets du monde réel -> Entités / Liens entre les objets -> Associations
* Le modèle entité-association est un modèle de données de type conceptuel.
* Il constitue une base pour plrs méthodes et outils d’aide à la conception des BD

#### Entité (ou Individu ou Objet)

* C’est un type d’objet ou de concept (concret ou abstrait) manipulé par l’organisation étudiée. On peut la définir sans faire référence à d’autres entités (ex : entités client, Produit, Commande, …)
* Représentation de l’entité Etudiant

#### Propriété (ou Attributs) d’une entité

* Une entité est caractérisée par un ensemble de propriétés (ou attributs).
* Exemple : des propriétés possibles de l’entité Etudiant sont le CodeEtudiant, le nom, le prénom, l’adresse, le numéro de téléphone.

## II – Entité – Propriété – Identifiant

Etudiant

CodeEtudiant

NomEtudiant

PrénomEtudiant

* Représentation :
* Pour chaque propriété il existe un ensemble de valeurs autorisées que l’on appelle Domaine de la propriété.

#### Contrainte d’intégrité sur les propriétés :

Elles permettent de limiter les valeurs possibles d’une propriété.

Exemple : un âge doit être compris entre 0 et 120 ans, un salaire doit être compris entre 1 000 € et 50 000 €.

#### Occurrence d’une entité

C’est un élément individualisé et unique appartenant à cette entité.

Si on considère une propriété comme étant une variable, les valeurs qui lui seront affectées représentent les occurrences de cette propriété. Si on affecte une valeur à chacune des propriétés composant une entité, on obtient une occurrence de celle-ci.

Représentation : l’étudiant numéro 101 est

Etudiant

101

Dupont

Martin

une occurrence de l’entité Etudiant.

#### Identifiant (ou clé) d’une entité

C’est une propriété particulière de l’entité, telle qu’à chaque valeur de cette propriété corresponde une et une seule occurrence de l’entité.

Toute entité a un identifiant qui est constitué d’une ou plrs de ses propriétés et qui permet d’identifier sans ambiguïté chaque occurrence de l’entité.

L’identifiant figure en première position dans la liste des propriétés et il est souligné.

## III - Relations (ou associations) entre entités

#### Définition

Une relation permet de relier plrs entités. Elle est généralement caractérisée par un verbe.

La **dimension** d’une relation est le nb d’entités qu’elle relie (binaire, ternaire, quaternaire, …)

La **collection** d’une relation est la liste des entités qui participent à cette relation.

#### Occurrence d’une relation

C’est une relation individualisée constituée d’une et une seule occurrence des entités qu’elle relie.

#### Propriété (ou attribut) d’une relation

Une relation peut posséder des propriétés. Il s’agit d’informations qui ne peuvent prendre de sens qu’avec la présence des entités de la relation qu’elle relie.

## IV – Cardinalités d’une relation

C’est le nombre minimum et maximum d’occurrence d’une relation pour une occurrence d’une entité.

|  |  |  |
| --- | --- | --- |
| **Cardinalité** | **Simple** | **Multiple** |
| Obligatoire | 1,1 | 1,N |
| Facultative | 0,1 | 0,N |

## V – Règle de normalisation – Entités

Première forme normale : chaque entité doit posséder un identifiant qui caractérise ses individus (occurrences) de manière unique.

Deuxième forme normale : l’identifiant peut être composé de plrs attributs mais les autres attributs de l’entité doivent être dépendant de l’identifiant en entier (et non une partie de cet identifiant)

Troisième forme normale (importante) : les attributs d’une entité doivent dépendre directement de son identifiant.

(Par exemple, le date de fête d’un client ne dépend pas de son identifiant numéro de client mais plutôt de son prénom. Il faut donc créer une entité calendrier à part.)

## VI – Règles de normalisation

* Normalisation de relations (importante) : les attributs des associations doivent dépendre des identifiants de toutes les entités en association.
  + Par exemple, la quantité commandée dépend à la fois du numéro de client et du numéro d’article,
  + Par contre la date de commande non. Il faut donc créer une entité commandes à part.

## VII – Construction du MCD

* Pour la construction du MCD, on a à notre disposition :
  + Un dictionnaire des Données (liste des attributs + différentes caractéristiques)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nom symbolique | Description | Type | Commentaires | Contraintes |
| Ex : NoClient | N° du client | N (entier) | N° séquentiel | Automatique |

* + Des règles de Gestion (ensemble de règles qui vont permettre notamment de fixer les cardinalités)
  + Une liste des résultats attendus

#### Phase A : Epuration de Dictionnaire des données

* Suppression des synonymes (ex : N°Salarié et CodeSalarié ; RefProduit et CodeProduit)
* Suppression des polysèmes (on ne doit pas avoir la même propriété Nom pour NomClient et NomSalarié)

#### Phase B

* Repérer les Entités du domaine étudié qui apparaissent le plus naturellement (ex : Client, Fournisseur, Dépôt, Commande, …)
* Et leur attribuer des propriétés.

#### Phase C

* Etablir les relations entre entités

#### Phase D

* Fixer les cardinalités

#### Phase E

* Vérifier la troisième forme normale et la normalisation des relations
* Effectuer les corrections nécessaires

# Chapitre 4 : Le modèle relationnel

## I – Introduction

#### Objectifs :

* Permettre un haut degré d’indépendance Données – Programmes d’application.
* Fournir une base solide pour traiter de pb de cohérences et de redondance de données
* Permettre le développement de LMD (langages de manip de données non-procéduraux basés sur des théories solides
* Etre un modèle ensemble permettant de modéliser et de manipuler des données complexes. C’est u objectif important car il a permis d’intégrer de nouveaux concepts (ex : les objets)
* Devenir un standard pour la description et la manipulation de bdd. Cet objectif a été réalisé en particulier grâce à IBM (langage SQL)

## II – Domaine

**Définition :** Ensemble de valeurs caractérisé par un nom.

#### Relation

* La relation est le concept central du modèle relationnel
* Plus simplement, une relation est un tableau à deux dimensions
* Une ligne est un n-uplet ou tuple ou enregistrement
* Nom associé à chaque colonne afin de la repérer indépendamment de l’ordre = attribut
  + Associé à un domaine

#### Attribut

**Définition :** Colonne d’une relation caractérisée par un nom

Les attributs prennent leurs valeurs dans les domaines, plrs attributs peuvent être définis sur le même domaine.

#### Tuple

**Définition :** Ligne d’une relation correspondant à un enregistrement (assignation de valeurs à chacun des attributs d’une relation).

#### Clé d’une relation

**Définition :** Une clé est un groupe d’attributs minimum qui détermine un n-uplet unique dans une relation (à tout instant)

**Contrainte d’intégrité :** Toute relation doit posséder une clé renseignée (sans valeur inconnue)

#### Schéma d’une relation

**Définition :** Le schéma d’une relation décrit :

* Son nom
* La liste des attributs qu’elle comporte et des domaines associés
* La liste des attributs composant la clé
* Le schéma d’une relation représente son **intention**, c’est-à-dire les attributs communs et invariants des tuples qu’elle va contenir au cours du temps.
* Une table représente une **extension** d’une relation, càd une vue des tuples qu’elle contient à un instant donné.
* Le **degré** ou **arité** d’une relation : nb d’attributs de la relation.
* La **cardinalité** d’une relation : nb de tuples.

**Donc :**

Le schéma d’une relation représente son **intention**,

La table avec tous les tuples représente une **extension** de la relation.

L’intention est le résultat de la **description des données**

L’extension fait suite à des **manipulations des données** et représente l’état de la base

#### Base de données relationnelle

**Définition :** BDD dont le schéma est un ensemble de schémas de relations et dont les occurrences sont des tuples de ces relations.

## III – Dépendances fonctionnelles

#### Définitions

* La notion de dépendance fonctionnelle (dépendance entre données) fut introduite pour caractériser des BD cohérentes.
* Un attribut (ou grp d’attributs) y dépend fonctionnellement d’un attribut (ou grp d’attributs) x, su étant donnée une valeur de x, il lui correspond une valeur unique y et ceci quel que soit l’instant considéré.
* On dit aussi que **x détermine y** ou y est fonctionnellement dépendant de x et on note x 🡪 y.
* X est appelé le **Source** et y le **But**.

#### Transitivité

**X** 🡪 **Y** et **Y** 🡪 **Z** => **X** 🡪 **Z**

#### Dépendance fonctionnelles élémentaires

* Dépendance fonctionnelle de la forme < 6< y, où y est un attribut unique non inclus dans x et où il n’existe pas x’ c x tel que x’ 🡪 y. (x est l’ensemble minimum d’attributs déterminant y)
* Autrement dit, on dit qu’il y a dépendance fonctionnelle élémentaire entre les attributs x et y si aucune partie de x ne détermine y.

#### Dépendance fonctionnelles directes

On dit que l’attribut y dépend fonctionnellement de x par une dépendance fonctionnelle élémentaire directe si cette dépendance est élémentaire et s’il n’existe pas de propriété z telle que : x 🡪 z et z 🡪 y.

**On élimine toute transitivité.**

## IV – Clé d’une relation

* L’ensemble minimum d’attributs d’une relation pouvant déterminer tous les autres est appelé Clé.
* La clé détermine un n-uplet de façon unique.

#### Normalisation

* **BUT : Eliminer les redondances et incohérences**
* La normalisation constitue un ensemble de règles introduites dans le modèle relationnel dès son origine
* Le concept de forme normale d’une relation a été introduit par CODD en proposant une classification en trois formes normales.
* Cette classification repose sur la notion de dépendance fonctionnelle.

#### Première forme normale (1FN)

Une relation est une première forme normale si :

* Elle possède une clé
* Contient une valeur atomique (non multiple, non composée)

#### Deuxième forme normale (2FN)

Une relation est une deuxième forme normale si :

* R est en 1FN
* Tout attribut n’appartenant pas à une clé ne dépend pas d’une partie de cette clé (tout attribut doit dépendre fonctionnellement e la totalité de la clé) – DF Elémentaire

#### Troisième forme normale (3FN)

Une relation est une troisième forme normale si :

* R est en 2FN
* Tout attribut n’appartenant pas à une clé ne dépend pas d’un attribut non clé