

Chapitre 1 : Rappel sur les bases de données

1. Introduction

Historiquement, les bases de données sont apparues dans le but de répondre à plusieurs problèmes :

- Limiter le nombre d'interfaces utilisateurs.
- Diminuer les dépendances entre les données et les programmes d'application ;
- Diminuer les redondances d'informations, afin :
 - d'une part de diminuer la place nécessaire au stockage ;
 - d'autre part de diminuer les incohérences et les temps de mise à jour.



2. Base de Données

C'est un ensemble de données (sans redondance si possible) où les liaisons entre les entités sont conservées ; disponible simultanément pour plusieurs applications ; et géré par un Système de Gestion de Bases de Données S.G.B.D.

Les bases de données servent à stocker des informations sur un support informatique pendant une longue période de taille importante en autorisant des accès multiutilisateurs. Il faut donc gérer de manière efficace les accès aux disques et proposer une définition structurée des données afin d'éviter les redondances.

Les objectifs des bases de données sont :

1. **Élimination** de la redondance des données.
2. **Indépendance entre les programmes et les données** : on peut isoler le niveau utilisation (application) de l'organisation physique de données.
3. **Intégration de données** : intégration de toutes les données de l'entreprise dans un réservoir unique de données.

3. Système de Gestion de Base de Données

Un Système de Gestion de Base de Données (S.G.B.D) est d'une part un ensemble de modules qui assurent l'interface entre les usagers et les informations; d'autre part un outil qui permet la gestion d'informations (recherche, insertion, suppression, modification).

Les principales caractéristiques d'un SGBD sont :

- La confidentialité,
- Le partage des ressources,
- L'accès concurrent et
- La sécurité de fonctionnement)

Un SGBD est généralement partitionné fonctionnellement en trois parties :

1. **Interne** Ce sont les modules qui gèrent les liens entre les données manipulées, et les structures qui les stockent. Ils contiennent notamment toute l'algorithmique pour la gestion des informations.
2. **Externe** Ce sont les modules qui produisent les images des données, qui sont utilisables par les programmes d'applications, et par les utilisateurs. Ils assurent les différentes interfaces utilisateurs.
3. **conceptuel/logique** Ce sont des modules qui gèrent des fichiers d'informations propres au S.G.B.D. ainsi que des fichiers qui contiennent les schémas conceptuels des données.

Exemples de SGBD : Oracle, PostgreSQL, Access, MySQL, SQL server, DB2, Informix, etc.

4. Le modèle relationnel

Le modèle relationnel est dû à Codd en 1960. Il a été l'objet de nombreuses recherches jusque dans les années 1990. C'est le modèle le plus utilisé par les SGBDs actuellement disponibles sur le marché.

C'est un modèle très simple. Trop simple pour certains qui ont proposé d'autres modèles conceptuels permettant de mieux décrire certains aspects sémantique des données : modèles Entité-Association, relationnel-objet. . .

Numéro	Nom	Prénom	Date de naissance
65758	nom1	prénom1	16 juin 2005
7897090	nom2	prénom2	29 février 2004
7689689	nom1	prénom3	31 mars 1995
325	nom1	prénom1	16 juin 1997
45365	nom3	Prénom2	20 septembre 2002
...			
...			

Tableau 1: La relation étudiant

Le Domaine : Un domaine D est un ensemble de valeurs atomiques (non décomposable).

Quelques exemples de domaines :

- Les chaînes de caractères de longueur maximale 30.
- Les entiers positifs.
- Les entiers compris entre 1 et 12.
- Les couleurs {Rouge, Vert, Bleu}.

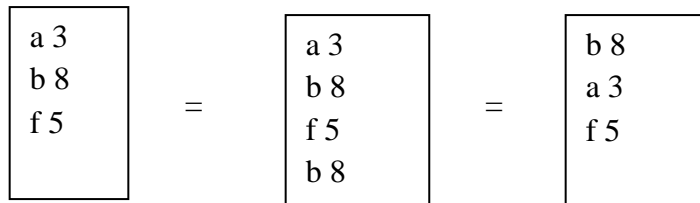
L'Attribut : Un attribut A est une caractéristique d'un objet représenté par une valeur prise dans un domaine. On note $\text{Dom}(A)$ le domaine de l'attribut A.

Instance : ou N-uplet, est une fonction totale qui associe à chaque attribut une valeur dans le domaine dom. Soit $\{A_1, A_2, \dots, A_n\}$ une liste d'attributs. Un n-uplet ou tuple est un vecteur de valeurs $V=(v_1, v_2, \dots, v_n)$ telque : $(v_1 \in \text{Dom}(A_1), v_2 \in \text{Dom}(A_2), \dots, v_n \in \text{Dom}(A_n))$.

La Relation : Soit $\{A_1, A_2, \dots, A_n\}$ une liste d'attributs. Une relation R est un ensemble de n-uplets $\{(v_1 \in \text{Dom}(A_1), v_2 \in \text{Dom}(A_2), \dots, v_n \in \text{Dom}(A_n))\}$.

Remarque : Une relation est un ensemble et non un multi-ensemble. Chaque n-uplet est donc unique, et il n'y a pas d'ordre entre les n-uplets.

Exemple :



Remarque : d'un point de vue ensembliste, $R \subseteq \prod_{i=1}^n \text{Dom}(A_i)$.

2.2 Ajout de sémantique dans le concept de relation

Identifiant ou clef : Soit une relation $R \subseteq \prod_{i=1}^n \text{Dom}(A_i)$.

L'ensemble $I = \{I_1, I_2, \dots, I_m\} \subseteq \{A_1, A_2, \dots, A_n\}$ est un identifiant si et seulement si :

- Il n'existe pas deux n-uplets dans R ayant même valeur sur l'ensemble I.
- Pour tout sous-ensemble $J \subset I$, il existe deux n-uplets dans R ayant même valeur sur l'ensemble J.

Remarques :

- Pour la relation étudiant de la table 1, le seul identifiant possible est Numéro.
- Si la relation R évolue au cours du temps, les ensembles identifiant peuvent évoluer aussi.

Soit le scénario suivant sur la relation étudiant de la table 1 :

1.1. retrait du n-uplet (325, nom1, prénom1, 16 juin 1985). Les ensembles {Numéro} et {Date de naissance} sont les identifiants.

1.2. ajout du n-uplet (65758, nom1, prénom1, 31 mars 1985). Les ensembles {Numéro, Date de naissance} et {Prénom, Date de naissance} sont les identifiants.

Ce scénario n'est pas souhaitable. L'identifiant d'une relation ne doit donc pas être un ensemble calculé dynamiquement, mais une notion statique donnée par le concepteur de la relation.

Remarque : Toute relation possède au moins une clef car la liste de tous les attributs vérifie le premier point.

Schéma d'une relation : Un schéma de relation est simplement un nom suivi de la liste des attributs, chaque attribut étant associé à son domaine. La syntaxe est donc :

$$R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$$

Le schéma d'une relation noté Schema (R) est constitué :

- d'un nom,
- d'une liste d'attributs (A_1, A_2, \dots, A_n) sur des domaines (D_1, D_2, \dots, D_n),
- d'un ensemble non vide d'identifiants (ou clefs) $\{I_i \subseteq \{A_1, A_2, \dots, A_n\}\}$,
- d'une contrainte d'intégrité P (éventuellement True).

Remarque : par abus de langage, on confond souvent le nom de la relation et le nom du schéma.

La redondance

Un attribut dans une table est redondant lorsque ses valeurs peuvent être *éliminées* de cette table *sans perte d'informations*.

5. Les formes normales

MEMBRE DU PROJET (table non normalisée)

<i>E#</i>	Nom	Ville	<i>P#</i>
E7	Humbert	Bulle	{ P1, P9 }
E1	Meier	Fribourg	{ P7, P11, P9 }

5.1. La première forme normale (1NF)

Une table satisfait à la première forme normale si les domaines de ses attributs sont constitués de valeurs atomiques. La première forme normale exige que chaque valeur d'un attribut provienne d'un domaine non structuré. Par conséquent, la définition du

domaine d'un attribut ne doit pas contenir d'ensembles, de types énumérés ou de groupes répétitifs.

La table MEMBRE DU PROJET n'est pas encore normalisée car chacun de ses tuples contient plusieurs numéros de projets auxquels participe un employé. La technique pour transformer une table non normalisée en première forme normale consiste simplement à créer un tuple distinct pour chaque engagement dans un projet. Lors de cette transformation, la clé de la table MEMBRE DU PROJET doit être redéfinie. En effet, afin d'identifier un tuple de manière unique, nous avons besoin à la fois du numéro d'employé et du numéro de projet. Les deux attributs qui forment ensemble une clé composée sont généralement (mais pas obligatoirement) placés dans les premières colonnes adjacentes de la table.

MEMBRE DU PROJET (1NF)

E#	P#	Nom	Ville
E7	P1	Humbert	Bulle
E7	P9	Humbert	Bulle
E1	P7	Meier	Fribourg
E1	P11	Meier	Fribourg
E1	P9	Meier	Fribourg

5.2. Deuxième forme normale (2NF)

Une table satisfait à la deuxième forme normale si elle est déjà en première forme normale et s'il existe une *dépendance fonctionnelle totale* reliant la clé à chaque attribut non clé.

Un attribut B est *fonctionnellement dépendant* de l'attribut A, si à chaque valeur de A correspond une et une seule valeur de B (noté conventionnellement $A \rightarrow B$). Par conséquent, la *dépendance fonctionnelle* (*functional dependency*, en anglais) de A vers B exprime le fait que chaque valeur de A détermine de manière unique une valeur de B. Une propriété connue des clés d'identification est que les attributs non clés dépendent de la clé de manière unique. Dans une table donnée, il existe donc une dépendance fonctionnelle $S \rightarrow B$ entre la clé d'identification S et un attribut quelconque B.

Dans la *dépendance fonctionnelle totale* (*full functional dependency*, en anglais) d'une clé vers un attribut, celui-ci ne doit donc pas dépendre des parties de cette clé.

Dans l'exemple précédent, l'éclatement donne lieu à deux tables, EMPLOYE et APPARTENANCE, qui sont toutes en première et deuxième formes normales. La table EMPLOYE ne contient plus de clé composée, et satisfait évidemment la deuxième forme normale. La table APPARTENANCE ne possède aucun attribut non clé, ce qui rend inutile la vérification des critères de la deuxième forme normale.

EMPLOYE (2FN)

E#	Nom	Ville
E7	Humbert	Bulle
E1	Meier	Fribourg

APPARTENANCE (2FN)

E#	P#
E7	P1
E7	P9
E1	P7
E1	P11
E1	P9

5.3. Troisième forme normale (3FN)

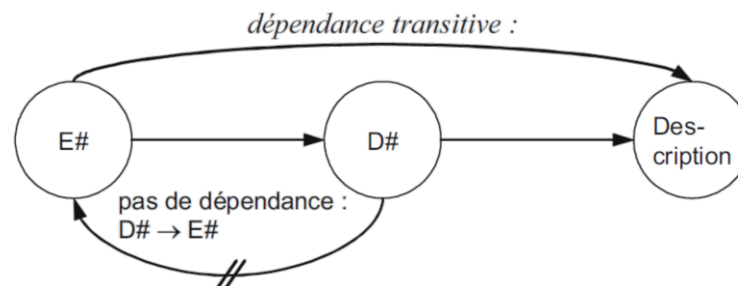
Une table est en troisième forme normale si elle est déjà en deuxième forme normale et qu'*aucun attribut non clé ne dépend d'une clé quelconque par transitivité*.

5.4. Les dépendances transitives

EMPLOYE DU DEPARTEMENT (2NF)

E#	Nom	Rue	Ville	D#	Description
E19	Savoy	avenue de la Gare	Romont	D6	Finances
E1	Meier	rue Faucigny	Fribourg	D3	Informatique
E7	Humbert	route des Alpes	Bulle	D5	Personnel
E4	Brodard	rue du Tilleul	Fribourg	D6	Finances

La dépendance transitive est une *dépendance fonctionnelle déduite des autres dépendances*. Par exemple, dans la relation EMPLOYE DU DEPARTEMENT, l'attribut Description dépend fonctionnellement du numéro d'employé (E#) par le biais du numéro de département (D#). En effet, nous savons qu'il existe une dépendance fonctionnelle entre le numéro d'employé et le numéro de département d'une part, et entre le numéro de département et sa description d'autre part. De ces deux dépendances fonctionnelles, $E\# \rightarrow D\#$ et $D\# \rightarrow \text{Description}$, nous dérivons la dépendance transitive $E\# \rightarrow \text{Description}$ par le biais du numéro de département.



Dans la relation EMPLOYE DU DEPARTEMENT ci-dessus l'attribut Description qui dépend de E# par transitivité. Donc, en vertu de la définition précédente, elle n'est pas en troisième forme normale.

Nous éliminons la dépendance transitive par éclatement de la table EMPLOYE DU DEPARTEMENT en deux tables distinctes : EMPLOYE et DEPARTEMENT. Ce

dernier contient l'attribut redondant **Description** et le numéro de département. Dans l'autre table **EMPLOYE**, nous conservons le numéro de département comme clé étrangère en l'appelant **D#_Affectation**. De cette manière, le lien entre l'employé et son département est préservé.

EMPLOYÉ (troisième forme normale)

E#	Nom	Rue	Ville	D#_Affectation
E19	Savoy	avenue de la Gare	Romont	D6
E1	Meier	rue Faucigny	Fribourg	D3
E7	Humbert	route des Alpes	Bulle	D5
E4	Brodard	rue du Tilleul	Fribourg	D6

DÉPARTEMENT (troisième forme normale)

D#	Description
D3	Informatique
D5	Personnel
D6	Finances

5.5. La forme normale de Boyce-Codd (BCFN)

La forme normale de Boyce-codd traite le cas où une partie d'une clé dépend d'un champ non clé. Donc, une table est en BCFN si tout champ appartenant à une clé ne dépend pas d'un champ non clé. Dans la relation suivante (table personne), la clé est composée du numéro de la sécurité sociale *SS#* et le *pays*.

Personne (3NF)

SS#	Pays	Nom	Région

Dans cette table, on a l'attribut **Région** qui détermine une partie de la clé, **Pays**. Donc on a **Région** → **Pays**, où une partie non clé qui détermine une partie de la clé. Donc cette table n'est pas en forme normale de Boyce-codd. Après le passage vers la forme normale de Boyce-codd on aura deux tables ; la table personne et la table région.

Personne (BCNF)

SS#	Région	Nom

Région (BCNF)

Région	Pays

6. Exemples

6.1. Exemple 1

On a les données suivantes sur des élèves avec le DFs:

ELEVE (Matricule, Nom, Age, Club, Salle)

Matricule \rightarrow Nom , AGE

Matricule \rightarrow Club

Club \rightarrow Salle

a. Que signifie chaque DFs?

b. Mettre ces informations dans un ensemble de schémas de relations normalisés.

6.2. Correction

Dans ce type d'énoncé, on donne les attributs et les DFs qui les lient. On vous demande ensuite de normaliser. Dans d'autres exemples, cf. exemple 2, il faut chercher soi-même ces attributs et/ou DFs.

Ensuite on suppose que les domaines des attributs sont sous-entendus, i.e. simple à concevoir... Ici, **Matricule** est un attribut défini sur le numéro matricule d'élève (e.g. 9867, 4 chiffres décimaux). On fera dorénavant cette hypothèse, sauf indication contraire.

1) Une dépendance fonctionnelle DF établit d'abord une relation entre donnée, en plus d'être fonctionnelle.

Matricule \rightarrow Nom, Age, signifie qu'il y a d'abord la relation "le matricule le nom et l'âge d'un certain élève" entre **Matricule**, **Nom** et **Age**. Ensuite, le nom et l'âge sont uniques pour un élève identifié par un matricule.

Matricule \rightarrow Club, signifie un élève est "inscrit" ou "participe" à un club donné. En plus, ce club est unique (un élève ne participe pas à plus d'un club).

Club \rightarrow Salle, signifie qu'un club "a un local qui est une salle". Cette salle est unique, (aucun club ne dispose de plus d'un local). Et on peut aussi déduire qu'une salle peut accueillir plusieurs clubs, car rien n'empêche cela dans les DFs.

2) Si maintenant on considère la relation : **ELEVE (Matricule, Nom , Age, Club, Salle)**, on peut dire que l'attribut **Matricule** est clé, car il détermine tous les autres attributs, y compris **Salle** (la DF **Matricule \rightarrow Salle** est transitive).

Cette relation est en 2FN, car aucun attribut non clé ne dépend d'une partie de la clé (la clé n'est pas composée d'ailleurs).

Cette relation n'est pas en 3FN, car les attributs non clés ne sont pas mutuellement indépendants, à cause de la dépendance fonctionnelle **Club \rightarrow Salle**. (Une autre façon de le dire, la DF par rapport à la clé **Matricule \rightarrow Salle**, n'est pas directe mais transitive par le fait que, par hypothèses, **Matricule \rightarrow Club** et que **Club \rightarrow Salle**). Donc on projette la relation **ELEVE** pour que cette DF (**Club \rightarrow Salle**) soit due à une clé (**Club**).

ELEVE (Matricule, Nom, Age, Club)
ACTIVITE (Club, Salle)

Ces deux relations sont en BCNF (car dans ELEVE, il n'y a plus de DFs entre attributs non clé). On retrouve la relation initiale par jointure des ces deux dernières relations, car Club est clé dans la deuxième relation

6.3. Exemple 2

Soit la relation des employés d'une société implantée sur plusieurs bâtiments.

EMPLOYES (NumE, Nom, Salaire, Département, Bâtiment)

Sachant qu'un employé travaille dans un département donné, et qu'aucun département ne possède des locaux dans plusieurs bâtiments.

1. Déterminer d'abord les DFs de la relation EMPLOYES.
2. En quelle forme normale est la relation suivante ?
3. Mettre en 3F le cas échéant.

6.4. Correction

Par rapport à l'exercice précédent, ici on doit trouver les DFs. D'après l'énoncé, on a Nume qui est un numéro est unique pour chaque employé. Donc on a :

Nume \rightarrow Nom, Salaire

On a aussi d'après l'énoncé, Nume \rightarrow Département (un employé travaille dans un département donné) et Département \rightarrow Bâtiment (un département ne possède pas des locaux dans plusieurs bâtiments. Noter ici l'expression d'une DF par une phrase négative.) D'où Nume est clé de la relation. (Nume \rightarrow Bâtiment étant transitive)

La relation est en 2FN, car la clé n'est pas composé, toutes les DFs sont totales. Mais il y a cette dernière DF qui est transitive. On normalise en 3FN par projection par rapport à cette DFs :

Employés (Nume, Nom, Salaire, Département)
Départements (Département, Bâtiment)

Ces deux relations sont en 3FN. Les attributs non clé sont mutuellement indépendants. Il n'y a pas une dépendance entre des attributs non clé avec des parties de la clé, alors les deux relation sont aussi en BCNF.