

# I- Introduction

## 1-I- Généralités

Les BD existent depuis environ 30 ans.

Principaux systèmes de la deuxième génération :  
ORACLE, INGRES, SYBASE, INFORMIX, DB2 et  
RDB.

L'architecture est répartie au moins avec des  
stations clients transmettant leurs requêtes à de  
puissants serveurs gérant les bases.

Principaux systèmes de la troisième génération :  
ORACLE, INGRES, ONTOS, OBJECTORE, ACCESS,  
VERSANT, ORION et O2.

modèles à objets, règles, ...

**Applications** : CAO, bureautique, gestion  
avancée

**Définition SGBD** : ensemble de logiciels systèmes permettant de stocker et d'interroger un ensemble de fichiers interdépendants ; cet ensemble comporte des outils pour modéliser et gérer les données d'une entreprise.

Les *données* modélisent des objets ou associations d'objets du monde réel.

Les objets sont représentés par des **articles de fichiers**.

*Les associations* sont les **liens** entre les articles.

**Interne** : Ensemble de fichiers reliés par des pointeurs.

Les données sont interrogées ou mises à jour par des programmes utilitaires ou utilisateurs.

## 2-I Langages

troisième génération : C, COBOL, FORTRAN

quatrième : NATURAL, IDEAL, MANTIS, ...

Ils accèdent à un langage unifié : SQL ou QUEL

.

### 3.I Un peu de terminologie

Type d'objets : ensemble d'objets possédant des caractéristiques similaires et manipulables par des opérateurs identiques.

#### Exemples

(1) Entier =  $\{ 0, \overset{+}{-}1, \dots, \overset{+}{-}n, \dots, \overset{+}{-} \infty \}$

Opérations : +, -

(2) Vin avec propriétés Cru, Millésime,

Qualité

Opérations : boire, produire

(3) Entité avec des propriété  $P_i$

Opérations : Créer, Consulter, Détruire, ...

(Modélisation du monde réel)

Instance d'objets : élément particulier d'un type d'objets caractérisé par un identifiant et des valeurs de propriétés.

#### Exemples

(1) 10 ;

(2) (Volnay, 1992, Excellent, 100) ;

(3)  $e(a_1, \dots, a_n)$  .

Modèle de description de données (Data model) : ensemble de concepts et de règles de composition de ces concepts permettant de décrire des données.

Langage de description de données (Data description Language) : langage supportant un modèle et permettant de décrire les données d'une BD d'une manière assimilable pour la machine.

Agrégation : Abstraction consistant à grouper des objets pour constituer des objets composés d'une concaténation d'objets composants.

Exemple (Voir figure 2)

Schéma : description au moyen d'un langage déterminé d'un ensemble de données particulier.

*Schéma conceptuel , schéma interne , schéma externe*

### Exemple

#### **Schéma conceptuel**

*types d'objets :*

Buveur (Nom, Prénom, Adresse)

VIN (Cru, Millésime, Qualité, Quantité)

*type d'associations :*

ABUS (BUVEUR, VIN, Date, Quantité)

associe un VIN et un Buveur avec une Date et la Quantité.

#### **Schéma interne**

*(voir figure 3)*

#### **Schéma externe**

*(voir figure 4)*

## I-4 Modèle ENTITE-ASSOCIATIONS

Entité : modèle d'objets identifié du monde réel dont le type est défini par un nom et une liste de propriétés.

Association (relationship) : lien logique entre entités dont le type est défini par un verbe et une liste éventuelle de propriétés

Attribut propriété d'une entité ou d'une association caractérisée par un nom et un type élémentaire.

### Exemple

entité = { BUVEUR, VIN }

association = { ABUS }

attributs = { Nom, Prénom, Adresse et Type pour BUVEUR;

Cru, Millésimes, Qualité pour VIN;

Quantité et Date pour Abus }

## I-5 Cahier des charges

1. *Persistence* : sauvegarde
2. *Gestion du disque* : amélioration des performances.  
tout en mémoire secondaire et problèmes d'efficacité
  - (a) accès des données : Hachage - Indexation
  - (b) transfert des données en secondaire et principal avec utilisation de Buffer :  
“clustering ”
  - (c) structure physique : stocker les données
3. *Partage des données*  
verrouillage ; transaction : tout ou rien  
cohérence des transactions en parallèle  
(paquets d'actions)  
exemple : pour les banques

4. *Fiabilité*

(a) cohérence des données (contraintes d'intégrités servant d'invariants)

(b) reprise sur panne

5. *sécurité* : droits d'accès

6. *langage d'interrogation* ( SQL) les requêtes de haut niveau permettent d'éviter une programmation inutile pour retrouver les données.

## II- Modèle Relationnel

Codd (1970) (voir Ulman, 1988)

1. Permettre un haut degré d'indépendance des programmes d'applications et des activités interactives à la représentation interne des données
2. Fournir une base solide pour traiter les problèmes de cohérence et redondance des données
3. Permettre le développement de langages de manipulation de données non procéduraux basés sur des théories solides (atteint avec l'algèbre relationnelle et les langages assertionnels)
4. Avoir un modèle extensible qui permet des manipulations de données complexes
5. Devenir un standard pour la description et la manipulation des B.D.

## II-1 Définitions

Domaine : *ensemble de valeurs caractérisées par un nom.*

Ce sont donc les ensembles dans lesquelles les données prennent leur valeur. Un domaine est défini en extension ou en intention (compréhension) en définissant une propriété caractéristique des valeurs du domaine.

Au départ : ENTIER, REEL, BOOLEEN,  
CARACTERES ( STRING) définis en intention

*Exemples :*

MONNAIE : réel avec 2 chiffres après la virgule

DATE : entier de 6 chiffres

COULEUR-VINS = { Rosé, Blanc, Rouge }

CRU = { Volnay, Sancerre, Chablis }

Produit cartésien Soient  $D_1, \dots, D_n$  des domaines.  
 Leur produit cartésien est

$$D_1 \times \dots \times D_n = \{(d_1, \dots, d_n) \mid d_i \in D_i\} .$$

*Exemple* : Couleur-Vins  $\times$  Cru =  
 $\{ (\text{Rosé}, \text{Volnay}), (\text{Blanc}, \text{Chablis}), \dots \}$

Relation : sous-ensemble du produit cartésien  
 d'une liste de domaines caractérisé par un nom.

*Exemple* : Couleur-Cru =  $\{ (\text{Rosé}, \text{Sancerre}),$   
 $(\text{Blanc}, \text{Sancerre}), (\text{Rosé}, \text{Chablis}) \}$

Couleur-Cru	Couleur	Cru
	Rosé	Sancerre
	Blanc	Sancerre
	Rosé	Chablis

Attribut : colonne d'une relation caractérisée par un nom.

*Exemple* : Couleur ou Cru sont des attributs de Couleur-Cru

Tuple : ligne d'une relation correspondant à un enregistrement.

*Exemple*: (Rosé, Sancerre) est un tuple de la relation Couleur-Cru.

*Remarque* : Une relation peut aussi être représentée par un diagramme. Par exemple le diagramme d'une relation du produit âges  $\times$  salaires suivant :

Nous mettrons sur les ensembles de relations une structure d'algèbre avec les opérations :  $\cup$ , diff, produit cartésien,  $\cap$

## II-2 Extensions et intentions

Les relations varient en fonctions du temps de part la modification des données. Ce qui reste invariant c'est sa structure : domaine, relation et attribut. Cette structure est capturée dans le schéma de la relation.

Schéma de la relation :  $R(A_1 : D_1, \dots, A_n : D_n)$ ,  
où  $R$  est le nom de la relation,  $A_i$  est le nom du  $i$ -ème attribut de domaine  $D_i$ .

*Exemple :*

VINS(CRU:CHAR, MILL:ENTIER, DEGRE:REEL)

Le schéma d'une relation représente son intention alors que la table représente une extension (appelée aussi une *instance* de la relation).

## II-3 Règles d'intégrité

$$\mathcal{E} = D_1 \times \cdots \times D_n$$

**Clé** : *ensemble d'attributs minimal dont la connaissance des valeurs permet d'identifier un tuple unique de la relation.*

Autrement dit : Soit  $K = D_{i_1} \times \cdots \times D_{i_m}$  et  $\pi_K$  la projection canonique de  $\mathcal{E}$  dans  $K$ . Soit  $R$  une relation de  $\mathcal{E}$ , alors l'ensemble des attributs de  $\pi_K(R)$  est une clé si  $R = \pi_K^{-1}(\pi_K(R))$  et que  $m$  est minimal pour cette propriété.

Si une clé permet d'identifier plusieurs attributs, elle est appelée une *super clé*.

Dans le cas où il existe plusieurs clés, il faut en choisir une appelée *clé primaire* (ACCESS).

*Remarque* La clé doit être la même si on étend  $R$  en extension. Donc elle est rattachée à la sémantique de la relation, elle doit en exprimer l'intention . En d'autres termes,  $\pi_K$  doit être injective.

*Exemple*

Dans la table **Assurés Sociaux**, la clé est **SS** :

SS	Nom	Prénom
----	-----	--------

Dans la table **Hopital**, la clé est **Num**

Num	Ville	Nom	spécialistes
-----	-------	-----	--------------

Une entité est identifiée par sa clé. Un type d'association est modélisé par une relation comportant les clés des entités participantes ainsi que les caractéristiques propres à l'association.

*Exemple* : Nous pouvons créer simplement l'association MALADE dont le schéma est :  
MALADE(SS , NUM, début,fin,opération) sa clé sera (SS,NUM,début).

*Exemple* : (voir Figure 6) Schémas :  
BUVEUR(NB ,NOM ,PRENOM ,ADRESSE)  
VIN(NV ,CRU ,MILL ,QUAL ,DEG)  
ABUS (NB ,NV ,DATE ,QUANTITE)  
La clé de ABUS est (NB,NV,DATE).

Il faut garantir les liens obligatoires pour les associations :

Contrainte Référentielle : *contrainte d'intégrité portant sur une relation  $R_1$  consistant à imposer que la valeur d'un groupe d'attributs apparaisse comme valeur de clé dans une autre relation  $R_2$ . (il se peut que  $R_1 = R_2$ )*

Une contrainte référentielle est représentée par une *clé étrangère*.

Clé étrangère : *groupe d'attributs qui doit apparaître comme clé dans une (autre) relation.*

*Exemple* : NB et NV sont les clés étrangères de la clé d'ABUS.

Les contraintes référentielles sont indispensables pour la mise à jour. Il faut qu'un vin existe dans VIN pour en faire référence dans ABUS.

Réciproquement, cela permet de gérer la suppression d'un tuple dans VIN s'il est impliqué dans ABUS.

Il se peut que des instanciations ne soient pas connues lors de l'insertion d'un tuple dans une relation.

Valeur Nulle : *valeur conventionnelle introduite dans une relation pour représenter une information inconnue ou inapplicable.*

Contrainte d'entité : *contrainte d'intégrité imposant que toute relation possède une clé primaire et que tout attribut participant à cette clé primaire soit non nul.*

Rien n'est imposé a priori sur les clés secondaires.

Contraintes de domaines: *contrainte d'intégrité imposant qu'une colonne d'une relation doit comporter des valeurs vérifiant une assertion logique.*

Elles sont utilisées afin de définir des domaines plus fins que ceux proposés a priori.

*Exemples* : Chomeurs  $\leq$  6 millions

$0 \leq$  Pourcentage  $\leq$  100

## II-4 Algèbre relationnelle

C'est une algèbre sur les relations. Il y a 6 opérations de base : ensemblistes traditionnelles (binaires) et spécifiques. Dans toute cette partie, nous considérerons  $R_1$  et  $R_2$  deux relations.

Union Supposons que  $R_1$  et  $R_2$  aient le même schéma. L'union  $R_1 \cup R_2$  de  $R_1$  et  $R_2$  est une relation de même schéma que  $R_1$  comportant leurs tuples sans répétition :

$$R_1 \cup R_2 = \{X \mid X \in R_1 \text{ ou } X \in R_2\} .$$

*Notations :*

$R_1 \cup R_2$ , Union( $R_1, R_2$ ), APPEND( $R_1, R_2$ )

Différence Supposons que  $R_1$  et  $R_2$  aient le même schéma. La différence  $R_1 - R_2$  de  $R_1$  et  $R_2$  est une relation de même schéma que  $R_1$  comportant les tuples de  $R_1$  n'appartenant pas à  $R_2$  :

$$R_1 - R_2 = \{X \in R_1 \mid X \notin R_2\}$$

*Notations :*  $R_1 - R_2$ , Difference( $R_1, R_2$ ),  
REMOVE( $R_1, R_2$ ), MINUS( $R_1, R_2$ )

Produit cartésien *Le produit cartésien de  $R_1$  et  $R_2$ , noté  $R_1 \times R_2$ , est une relation dont le schéma est le résultat de la concaténation de celui de  $R_1$  avec celui de  $R_2$  et comportant les tuples résultant de toutes les combinaisons de ceux de  $R_1$  et de  $R_2$  :*

$$R_1 \times R_2 = \{(x_1, \dots, x_n, y_1, \dots, y_q) \mid X \in R_1 \text{ et } Y \in R_2\}$$

*Notations :*

$R_1 \times R_2$ , PRODUCT( $R_1$ ,  $R_2$ ), TIMES( $R_1$ ,  $R_2$ )

Soit  $R$  une relation. Soient  $A_1, \dots, A_q$ ,  $q$  attributs de la relation  $R$  ( $0 < q \leq n$ ) de domaines respectifs  $D_1, \dots, D_q$ . Posons  $D := D_1 \times \dots \times D_q$  et  $A := A_1 \times \dots \times A_q$ .

Projection d'un tuple sur  $A$  Soit  $Y = (y_1, \dots, y_n)$  un tuple de  $R$  alors la projection de  $Y$  sur  $A$ , notée  $\pi_A(Y)$ , est le tuple  $(y_{i_1}, \dots, y_{i_q})$  composé des coordonnées de  $Y$  telles que  $y_{i_k} \in A_k$  pour  $1 \leq k \leq q$ .

Projection La projection de  $R$  sur  $A$ , notée  $\pi_A(R)$ , est une relation de schéma  $\pi_A(R)(A_1 : D_1, \dots, A_q : D_q)$  comportant les projections des tuples de  $R$  sur  $A$  :

$$\pi_A(R) = \{X \in D \mid (\exists Y \in R) \pi_A(Y) = X\}$$

*Notations* :  $\pi_A(R)$ ,  $\text{PROJECT}(R)$ ,  $R[A_1, \dots, A_q]$ .

Restriction Soit  $C$  une condition sur les tuples de d'une relation  $R$ . La restriction  $\sigma_C(R)$  est une relation de même schéma que  $R$  comportant les tuples de  $R$  vérifiant  $C$  :

$$\sigma_C(R) = \{X \in R \mid C(X) \text{ est vraie}\}$$

*Notations* :  $\sigma_C(R)$ ,  $R[C]$ ,  $\text{RESTRICT}(R, C)$ .

*Exemple* :  $C : x_1 \leq 3$ .

Notons  $D^i$  le domaine de  $R_i$ .

Soit  $C$  une condition sur les relations  $R_1$  et  $R_2$  qui peut-être vue comme une application  $D^1 \times D^2$  à valeur dans  $\{0, 1\}$ .

Jointure : La jointure  $R_1 \triangleright \triangleleft_C R_2$  de  $R_1$  et  $R_2$  sous la condition  $C$  est la relation  $R_3$  dont schéma est celui de  $R_1 \times R_2$  et défini comme suit en extension :

$$R_1 \triangleright \triangleleft_C R_2 = \{(x_1, \dots, x_n, y_1, \dots, y_q) \mid \\ X \in R_1, Y \in R_2, C(X, Y) = 1\}$$

Elle peut-être vue comme une extension du produit cartésien.

Nous avons :  $R_1 \triangleright \triangleleft_C R_2 = \sigma_C(R_1 \times R_2)$ .

La jointure  $R_1 \triangleright \triangleleft_{=} R_2$  est appelée une *équi-jointure*.

Si  $C$  est une inégalité, la jointure  $R_1 \triangleright \triangleleft_C R_2$  est appelée une *inéqui-jointure*.

La jointure n'évite pas les duplications des tuples :

Jointure naturelle la jointure naturelle  $R_1 \triangleright \triangleleft R_2$  de  $R_1$  et  $R_2$  est une relation dont l'ensemble attributs est l'union de ceux de  $R_1$  et  $R_2$  et dont les tuples sont ceux en composant un attribut de  $R_1$  avec un de  $R_2$  ayant même valeur pour un attribut de même nom

Notations :  $R_1 \triangleright \triangleleft_C R_2$  JOIN( $R_1, R_2, C$ )