

Le modèle relationnel de base de données  
Concepts

## 1 - Définition

- **Représentation des données sous forme de table :**

Livre	Nolivre	Titre_livre
	1	Le rouge et le noir
	2	Axtérix le Gaulois
	3	Germinal
	4	Les systèmes de gestion de base de données

- **Principe simple** : 1 seul concept (relation ou table) pour décrire les données et les liens entre ses données.

- Ce modèle est issu des recherches de CODD pour IBM à San José dans les années 1970.

- La majorité des SGBD actuels sont basés sur ce modèle.

## 2 - Les Concepts de base du modèle

Le concept de relation découle directement de la théorie des ensembles

- **Rappel de quelques définitions**

**Domaine :**

Un domaine est un ensemble de valeurs

- Exemple :
  - Domaine des nombres entiers
  - Domaine des nombres rationnels
  - Domaine des couleurs du drapeau français :
    - { Bleu, Blanc, Rouge }

## 2 - Les Concepts de base du modèle - Rappel des définitions

**Produit cartésien :**

Le produit cartésien d'un ensemble de domaines  $D_1, D_2, \dots, D_n$ , noté  $D_1 \times D_2 \times \dots \times D_n$  est l'ensemble des n-uplets ou t-uples  $\langle V_1, V_2, \dots, V_n \rangle$  tels que  $V_i$  appartient à  $D_i$ .

- Exemple : Le produit cartésien des domaines :  
 $D_1 = (\text{BLEU}, \text{BLANC}, \text{ROUGE})$   
 $D_2 = (0, 1)$   
est composé des 6 t-uples suivants :

Bleu	0
Bleu	1
Blanc	0
Blanc	1
Rouge	0
Rouge	1

## 2 - Les Concepts de base du modèle - Rappel des définitions

### Relation :

Une relation est un sous-ensemble du produit cartésien d'une liste de domaine

Une relation est caractérisée par un nom.

### Exemple :

A partir des domaines

D1 = ( BLEU, BLANC, ROUGE ) et D2 = ( 0, 1 ),

on peut composer la relation CIEL suivante :

Ciel	D1	D2
Bleu		0
Bleu		1
Blanc		0
Blanc		1

## 2 - Les Concepts de base du modèle - Rappel des définitions

- Plus simplement, une relation peut être vue comme un **tableau à deux dimensions** dont **les colonnes** correspondent aux **domaines** et les **lignes** contiennent des **t-uples**. On parle parfois de **Table**.
- Chaque relation a donc une structure de table et peut constituer un fichier.
- L'ordre des colonnes est sans importance.
- On associe un nom à chaque colonne.
- Extension d'une table** : le contenu de la table; tous ses tuples
- Cardinalité** : nombre de tuples de la relation. Par exemple, la cardinalité dans CIEL est 4.

## 2 - Les Concepts de base du modèle - Base de données relationnelles

### Base de données relationnelles :

#### Attribut :

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

#### Schéma de relation :

Un schéma de relation est le nom de la relation suivi de la liste des attributs avec leurs domaines. Ainsi, à toute relation, il est possible d'associer un schéma de relation qui représente **l'intention** de la relation, alors que le tableau avec les tuples représente une **extension** possible.

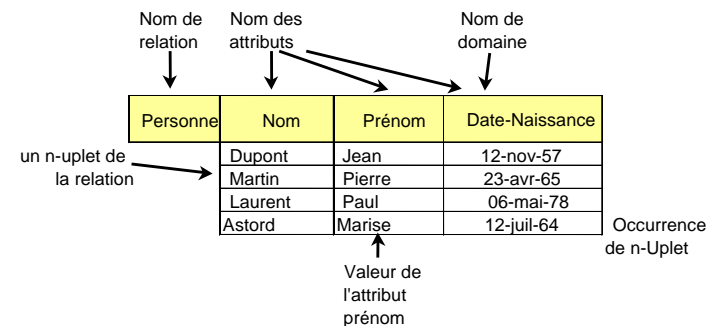
**Exemple** Dans un club, on désire gérer des adhérents, les cours proposés et les inscriptions à ces cours. Un schéma de relations pourrait être le suivant :

ADHERENT ( NADH : entier, NOM : caractère(15), PRENOM : caractère(15), DATE-NAISS : date )

COURS ( DISCIPLINE : caractère(10), NBP : entier, DATE-DEB : date, DATE-FIN : date, NADH : entier )

## 2 - Les Concepts de base du modèle - Base de données relationnelles

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



### Base de données relationnelle :

Base de données dont le schéma est un ensemble de schémas de relations et dont les occurrences sont des tuples de ces relations.

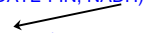
## 2 - Les Concepts de base du modèle - Base de données relationnelles

- Dans le modèle relationnel, tout est traduit explicitement par des données.
- Si l'on veut associer deux relations, un domaine commun doit y apparaître pour faire le pont explicitement entre les relations.

Par exemple, pour indiquer qu'un adhérent suit un cours on pourra répéter son numéro dans la relation COURS déclarée comme suit :

COURS ( DISCIPLINE, NBP, DATE-DEB, DATE-FIN, NADH)

ADHERENT (NOM, PRENOM, DATE-NAISS, NADH)



## 2 - Les Concepts de base du modèle – Clé d'une relation

### Clé primaire :

Groupe d'attributs minimum qui détermine un tuple d'une manière unique dans la table.

- Exemple de clés :
  - le numéro de la SECU
  - Le numéro étudiant
- La *clé primaire* de la table ADHERENT est l'attribut «**NADH**», car il permet de déterminer de façon unique une ligne de la table.
- **ATTENTION** : la clé se détermine par rapport à *toutes les valeurs possibles de l'attribut* (ou les attributs) formant la clé primaire et surtout pas par rapport aux valeurs déjà saisies.
- **Toute table doit obligatoirement avoir une clé primaire.**

## 2 - Les Concepts de base du modèle – Clé d'une relation

### Clé étrangère :

Nous appelons clé étrangère toute clé primaire apparaissant dans une autre table.

- L'attribut «NADH» est une clé étrangère dans la table COURS; il permet d'associer les deux relations (ADHERENT et COURS).
- **ATTENTION** : toutes les valeurs d'attributs de la clé étrangère sont des valeurs d'attributs de la clé primaire de la seconde relation.

## 2 - Les Concepts de base du modèle - Un exemple

### Un exemple de description et de stockage de données

La description d'une gestion de commandes peut se présenter sous la forme de trois relations :

- La relation fournisseur comprenant les attributs Code fournisseur, nom du fournisseur, situation et ville, et dont la clé est le code fournisseur.

On écrira : **F** ( CF, FNOM, ZONE, VILLE )

- La relation article comprenant les attributs Code article, nom de l'article, couleur et poids, et dont la clé est le code article.

On écrira : **A** ( CA, ANOM, COULEUR, POIDS )

- La relation fournisseur-article comprenant les attributs Code fournisseur, Code article et quantité, et dont la clé est le groupe : code fournisseur, code article

On écrira : **FA** ( CF, CA, QUANTITE )

## 2 - Les Concepts de base du modèle - Un exemple

- Au niveau physique, on pourra avoir trois fichiers représentés sur le schéma suivant :

- Le fichier Fournisseur contenant 5 tuples (ou enregistrements)
- Le fichier Article contenant 6 tuples (ou enregistrements).
- Le fichier Fournisseur-article contenant 14 tuples.

F	CF	Fnom	Zone	Ville
F1	Dupont	20	Londres	
F2	Durand	10	Paris	
F3	Dumont	30	Paris	
F4	Dubois	20	Londres	
F5	Martin	30	Athènes	

FA	CF	CA	Qté
F1	A1	3	
F1	A2	2	
F1	A3	4	
F1	A4	2	
F1	A5	1	
F1	A6	1	
F2	A1	3	
F2	A2	4	
F3	A3	4	
F3	A5	2	
F4	A2	2	
F4	A4	3	
F4	A5	4	
F5	A5	5	

A	CA	Anom	Poids	Couleur
A1	Ecrou	12	Rouge	
A2	Verrou	17	Vert	
A3	Vis	17	Bleu	
A4	Vis	14	Rouge	
A5	Lame	12	Bleu	
A6	Dent	19	Rouge	

## 3 - Les anomalies d'un mauvais schéma

- Exemple de relations :

TRANSPORTS (VOL, AVION, CAPACITE)

TRANSPORTS	VOL	AVION	CAPACITE
	IT5033	AIRBUS	314
	AF2401	AIRBUS	314
	AF2409	BOIENG 727	150
	IT5133	AIRBUS	314
	IT5035	MERCURE	150
	AF2802	AIRBUS	314

- La relation précédente possède plusieurs anomalies :

- Redondances des données :

- Par exemple, la capacité apparaît autant de fois pour un même type d'avion
- On apprend 4 fois qu'un AIRBUS a une capacité de 314 places

- Risques d'incohérence :

- Par exemple, si modification de la capacité des AIRBUS à 320 places, alors il faut répercuter cette modification sur tous les n-uplets.
- Le remplacement d'un AIRBUS par un BOEING 727 sur le vol AF2802 doit être accompagné de la modification de la capacité, en introduisant bien 150.

- Valeurs nulles :

- L'impossibilité pour la compagnie de pouvoir introduire un nouveau type d'avion dans sa flotte sans le faire voler car l'attribut vol est indispensable.

## 3 - Les anomalies d'un mauvais schéma – Éliminer les redondances

- Pour éliminer les répétitions nous allons construire la table AVIONS :

→ AVIONS (AVION, CAPACITE)

- La table TRANSPORTS se réduit à une autre table VOLS :

→ VOLS (VOL, AVION)

TRANSPORTS	VOL	AVION	CAPACITE
	IT5033	AIRBUS	314
	AF2401	AIRBUS	314
	AF2409	BOIENG 727	150
	IT5133	AIRBUS	314
	IT5035	MERCURE	150
	AF2802	AIRBUS	314

### Décomposition

VOLS	VOL	AVION
	IT5033	AIRBUS
	AF2401	AIRBUS
	AF2409	BOIENG 727
	IT5133	AIRBUS
	IT5035	MERCURE
	AF2802	AIRBUS

AVIONS	AVION	CAPACITE
	AIRBUS	314
	BOEING 727	150
	MERCURE	150

## 4 - La conception de schéma relationnel

### L'approche par décomposition

- L'approche par décomposition permet de concevoir des schémas relationnels à partir d'une relation composée de tous les attributs (**Relation Universelle**) en la décomposant en sous-relations qui n'ont pas les défauts signalés sur l'exemple précédent.
- Cette théorie utilise deux opérations de manipulation des relations :

#### Projection :

La projection de la relation R (A1, A2, ... An) sur les attributs Ai1, Ai2, ... Aip, est une relation R' de schéma R' (Ai1, Ai2, ... Aip) obtenue par élimination des valeurs des attributs de R n'appartenant pas à R' et suppression des tuples en double.

On notera une telle projection :  $\pi_{A_{i1}, A_{i2}, \dots, A_{ip}}(R)$ .

- La projection permet de supprimer des attributs d'une relation et d'éliminer les tuples en double qui apparaissent dans la nouvelle relation.

#### 4 - La conception de schéma relationnel

##### L'approche par décomposition

- L'opération inverse de la projection est la jointure naturelle.

##### Jointure naturelle :

La jointure de deux relations R (A1, A2, ... An) et S (B1, B2, ... Bm) est une relation T ayant pour attributs l'union des attributs de R et S et pour tuples, tous ceux obtenus par concaténation des tuples de R et S ayant même valeurs pour les attributs de même nom.

$$R * S = T (\{A_1, A_2, \dots, A_n\} \cup \{B_1, B_2, \dots, B_m\})$$

On aura donc :

$$\pi_{A_1, A_2, \dots, A_n}(T) = R$$

$$\pi_{B_1, B_2, \dots, B_m}(T) = S$$

- La jointure est une opération commutative et associative

#### 4 - La conception de schéma relationnel - L'approche par décomposition

- Il est maintenant possible d'introduire plus précisément la notion de décomposition :

##### Décomposition :

Substitution d'une relation R (A1, A2, ... An) par une collection de relations R1, R2, ... Rn, obtenues par des projections de R et telles que la relation résultat des jointures R1 \* R2 \* ... \* Rn ait le même schéma que R

VOITURE	NV	Marque	Type	Puissance	Couleur
	1453 WA 69	Renault	Laguna TX	7	Bleue
	5236 XA 69	Renault	Laguna TX	7	Blanche

Décomposition 1			
R1	NV	Type	Couleur
	1453 WA 69	Laguna TX	Bleue
	5236 XA 69	Laguna TX	Blanche

R2	Type	Marque	Puissance
	Laguna TX	Renault	7

Décomposition 2		
V1	NV	Type
	1453 WA 69	Laguna TX
	5236 XA 69	Laguna TX

V2	Type	Puissance	Couleur
	Laguna TX	7	Bleue
	Laguna TX	7	Blanche

V2	Type	Marque
	Laguna TX	Renault

#### 4 - La conception de schéma relationnel - L'approche par décomposition

##### Décomposition sans perte :

Décomposition d'une relation R en R1, R2, ... Rn telle que pour toute extension de R, on ait :

$$R = R_1 * R_2 * \dots * R_n$$

- Parmi les deux décomposition ci-dessus, si l'on admet qu'à un type de véhicule sont associées une seule marque et une seule puissance, la décomposition 2 ne permet pas de retrouver toutes les informations par jointure

$$\text{VOITURE} \Leftrightarrow V1 * V2 * V3$$

#### 4 - La conception de schéma relationnel - L'approche par décomposition

V1	NV	Type
	1453 WA 69	Laguna TX
	5236 XA 69	Laguna TX

V2	Type	Puissance	Couleur
	Laguna TX	7	Bleue
	Laguna TX	7	Blanche

V1 * V2	NV	Type	Puissance	Couleur
	1453 WA 69	Laguna TX	7	Bleue
	5236 XA 69	Laguna TX	7	Bleue
	1453 WA 69	Laguna TX	7	Blanche
	5236 XA 69	Laguna TX	7	Blanche

V3	Type	Marque
	Laguna TX	Renault

V1 * V2 * V3	NV	Type	Puissance	Couleur	Marque
	1453 WA 69	Laguna TX	7	Bleue	Renault
	5236 XA 69	Laguna TX	7	Bleue	Renault
	1453 WA 69	Laguna TX	7	Blanche	Renault
	5236 XA 69	Laguna TX	7	Blanche	Renault

## 5 – Contraintes d'intégrité

- Un des avantages des bases de données réside dans la possibilité d'intégrer des contraintes que doivent vérifier les données à tout instant.

### Contrainte d'intégrité :

Expressions booléennes qui **doivent être satisfaites à tout instant** par la base de données.

- Trois types de contraintes d'intégrité :
  - **Contraintes sur les attributs,**
  - **Contraintes sur les n-uplets,**
  - **Contraintes sur les relations.**

## 5 – Contraintes d'intégrité – contraintes sur les attributs

- **Domaines : l'appartenance des valeurs à un domaine :**
  - ➔ Des domaines souvent très généraux : chaînes de caractères, nombres, dates
  - ➔ **Sexe char check (sexe in ('M', 'F'))**
  - ➔ **EtatCivil char check (EtatCivil in ('C', 'M', 'D', 'V'))**
- **Valeurs nulles : la possibilité d'avoir des valeurs d'attributs nulles**
  - ➔ Dans la réalité il est rare que toute l'information soit disponible
  - ➔ **Null** : attribut peut accepter des valeurs nulles.
  - ➔ **Not Null** : on oblige l'initialisation de l'attribut
  - ➔ **Sexe char check (Sexe in ('M', 'F')) not null**

## 5 – Contraintes d'intégrité – contraintes sur les n-uplets

- **La valeur d'un attribut peut dépendre d'une ou plusieurs autres valeurs du même n-uplet**
- **Exemple :**
  - ➔ Le champ nom de jeune fille doit être fourni si, et seulement si, (1) le sexe de la personne est féminin et (2) elle est soit mariée, soit veuve :  
$$\text{NomJeuneFille} \neq \text{null} \wedge \text{Sexe} = 'F' \wedge \text{EtatCivil} \in \{'M', 'V'\}$$
  
$$\text{NomJeuneFille} = \text{null} \wedge (\text{Sexe} \neq 'F' \vee \text{EtatCivil} \in \{'C', 'D'\})$$
- **Ceci se traduit par :**
  - **Check** ( (NomJeuneFille is not null and  
Sexe = 'F' and  
EtatCivil in ('M', 'V')) Or  
(NomJeuneFille is null and  
(Sexe <> 'F' Or EtatCivil in ('C', 'D')) ) )

## 5 – Contraintes d'intégrité – contraintes sur les relations

- **Contrainte de clé : une relation doit posséder une clé primaire :**
  - ➔ **Primary Key** : pour désigner une clé primaire
  - ➔ **NumEtudiant number(10) primary key,**
- **Contrainte de référence (clés étrangères) : contrainte exprimée entre deux tables. Tout n-uplet d'une relation faisant référence à une autre relation doit se référer à un n-uplet qui existe.**
  - ➔ **References** « nom\_table » (« nom\_atribut »)
  - ➔ Exemple : **COURS (DISCIPLINE, NBP, DATE-DEB, DATE-FIN, NADH)**  
**ADHERENT (NOM, PRENOM, DATE-NAISS, NADH)**
  - ➔ Dans la table COURS nous avons une contrainte de référence :  
➔ **NADH Number(5) REFERENCES ADHERENT(NADH)**