

Le modèle relationnel: notions de base

Mohamed EL ADNANI

Le modèle relationnel: Introduction

1. Les concepts de bases
2. Les domaines
3. Produit cartésien
4. Les tables (relations)
5. Les attributs
6. Contraintes d'intégrités
 1. Les clés
 2. Les clés étrangères
7. Schéma

Les concepts de base

- Le modèle relationnel a été introduit par E. F. Codd vers les années 70 aux laboratoires de recherche d'IBM san-josé.
- C'est un *modèle logique* associé aux SGBD relationnels (ex. Oracle, DB2, SQLServer, Access, Paradox, dBase...).
- Le modèle relationnel est simple (3 concepts), facile à appréhender, même pour un non spécialiste
- Le modèle relationnel représente l'information dans une collection de relations ou tables

Objectifs

- C'est un modèle basé sur la **théorie des ensembles** : notion mathématique fondée et approuvée.
- Parmi les objectifs de ce modèle :
 - permettre un haut degré d'indépendance entre les programmes d'application et les représentations internes des données (stockage, type d'indexage, ...)
 - fournir une base solide pour traiter les problèmes de cohérence et de redondance des données.

Autres objectifs

- Ces deux objectifs ont été **pleinement satisfaits** par le modèle relationnel grâce à la notion de vues externes et aux règles d'intégrités.
- D'autres objectifs ont été atteints par la suite :
 - permettre le développement de langages de manipulation de données non procéduraux basés sur des théories solides
 - être un modèle extensible permettant de modéliser et de manipuler simplement des données tabulaires, mais pouvant être étendu pour modéliser et manipuler d'autres données plus complexes
 - devenir un standard pour la description et la manipulation des bases de données (le modèle et son langage SQL sont normalisés en 1986).

Structures de base

- La structure de base est très simple
- Trois structures:
 - Le domaine
 - L'Attribut
 - La Relation
- Une analogie très forte avec la théorie des ensembles

Domaine

- Un domaine est un ensemble de valeurs atomiques.
- Le terme atomique signifie que ces valeurs sont considérées comme insécables au niveau du modèle
- Un domaine est caractérisé par un nom.
- Peut être défini
 - soit en intension : en définissant une propriété caractéristique des valeurs du domaine,
 - soit en extension : en donnant la liste de toutes les valeurs le composant
- Exemples :
 - entier
 - Réel
 - Chaînes de caractères
 - Salaire = {4 000..100 000}
 - Couleur= {bleu, blanc, rouge}

Les attributs

- La notion d'attribut en relationnel correspond à la notion de propriété pour une entité
- Un attribut est défini par un nom et un domaine
- Le domaine d'un attribut décrit les valeurs autorisées pour cet attribut
- Les valeurs d'un attribut sont atomiques: non décomposables
- Contraintes supplémentaires: unicité, non nullité, clé, clé étrangère, valeur par défaut

Produit cartésien de domaines

- Le produit cartésien $d_1 \times d_2 \times \dots \times d_n$ est l'ensemble des tuples (n-uplets) :
 $\langle V_1, V_2, \dots, V_n \rangle$ tel que $v_i \in d_i$

- Exemple

- $D_1 = \{\text{bleu}, \text{blanc}, \text{rouge}\}$
- $D_2 = \{\text{vrai}, \text{faux}\}$

Bleu	Vrai
Bleu	Faux
Blanc	Vrai
Blanc	Faux
Rouge	Vrai
Rouge	Faux

Les relations (tables)

- Sous-ensemble du produit cartésien d'une liste de domaines
- Une relation est caractérisée par un nom
- La forme tabulaire est la forme la plus utilisée pour représenter une relation.
- Dans cette table:
 - chaque ligne représente un vecteur ou enregistrement de l'ensemble du produit cartésien des domaines
 - chaque colonne représente un domaine.
- Exemple
 - D1 = couleur
 - D2 = booléen

RelCoul	Coul	Choix
	Bleu	Faux
	Blanc	Vrai
	Rouge	Vrai

Caractéristiques des relations

- une relation est un **ensemble de nuplets**, il n'y a donc pas de notion d'ordre sur les nuplets → l'ordre des lignes et des colonnes n'est pas significatif
- par contre un nuplet est une séquence ordonnée d'attributs,
- une valeur d'attribut est atomique mais peut être éventuellement nulle (valeur particulière qui indique que la valeur est manquante)

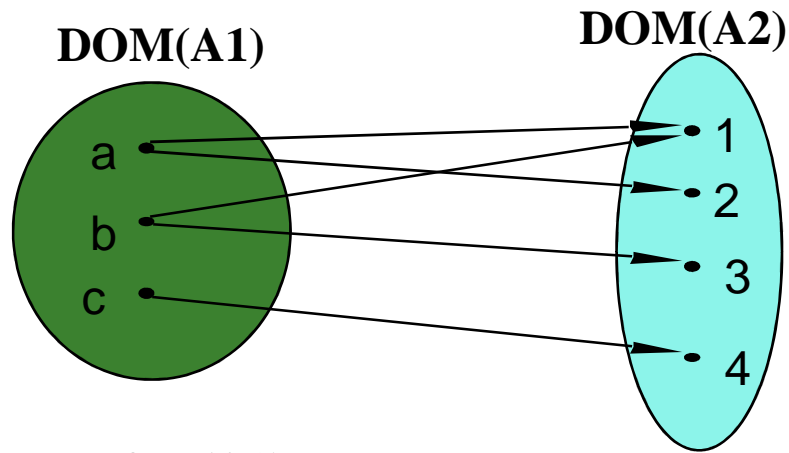
Les tables (relations) exemple

- Représentation graphique d'une table:
- EMPLOYES

Matricule	Nom	DateNaiss	Salaire	#NomService

- ❑ Nom de la table: employés
- ❑ Schéma: 1 ère ligne
- ❑ Tuple: chacune des autres lignes
- ❑ Attribut: une colonne
- ❑ Clé de la relation: Matricule
- ❑ Clé étrangère: #NomService

Graphe d'une relation



- Relation binaire $r(A1, A2)$
- Une relation n-aire est une généralisation a n dimensions

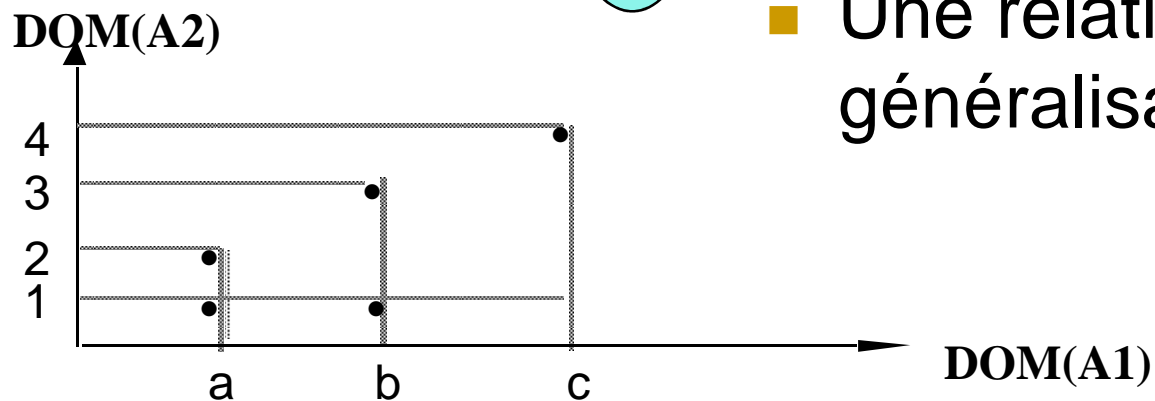


Schéma de relation

- un schéma de relation R , dénoté par $R(A_1, A_2, \dots, A_n)$, est un ensemble d'attributs $R = \{A_1, A_2, \dots, A_n\}$.
- A chaque attribut A_i est associé un domaine D_i , A_i indiquant le rôle joué par le domaine D_i dans la relation R .
- Un schéma de relation décrit une relation et représente **l'intension** de celle-ci.
- Le **degré** de la relation est le nombre d'attributs de celle-ci.
- Exemple :
- ETUDIANT(Nom, No-ss, adresse, age, diplôme)

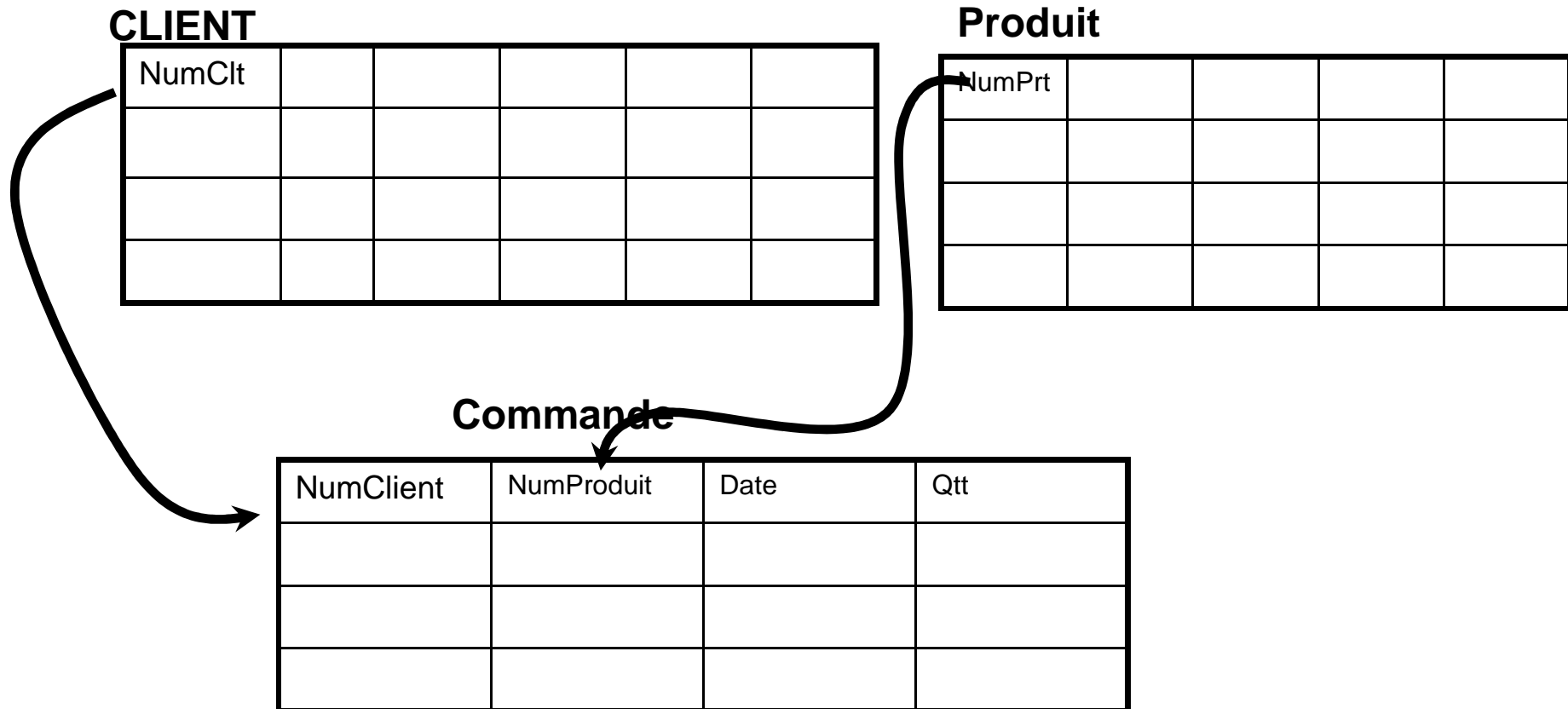
Schéma de relation en extension

- une relation r sur le schéma de relation $R(A_1, A_2, \dots, A_n)$ est un ensemble de nuplets $r = \{t_1, t_2, \dots, t_n\}$.
- r est souvent appelée extension du schéma R .
- On peut également définir une relation à partir du produit cartésien des domaines de son schéma R :
 - $r(R) \text{ inclus-ou-egal } \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$
- Un schéma de relation est quasi invariant dans le temps, alors que l'extension représente les données présentes à un instant donné dans la base

Schéma de BD

- Un schéma de base de données est un ensemble de schémas de relation $S = \{R1, R2, \dots, Rn\}$ et un ensemble de **contraintes d'intégrité CI**
- Exemple
 - ❑ CLIENT (NumCli, Nom, Prénom, DateNaiss, Rue, Ville)
 - ❑ PRODUIT (NumProd, Dési, PrixUni, *NumFour*)
 - ❑ FOURNISSEUR (NumFour, RaisonSoc)
 - ❑ COMMANDE (*NumCli*, *NumProd*, Date, Quantité)

Diagramme des Liens



Les contraintes d'intégrité

- Les contraintes (règles) d'intégrité sont les assertions qui doivent être vérifiées par les données d'une base.
- C'est une propriété du schéma, invariante dans le temps
- On peut distinguer plusieurs catégories de contraintes.
- Les contraintes structurelles: définissent plus précisément la structure des associations entre les données (le modèle de données les supporte en partie)
- les contraintes sur les valeurs : donnent des relations entre les données (un chef gagne plus que ses subordonnés).
- Le modèle relationnel impose une contrainte minimale qui est **l'unicité des clés**
- Les autres contraintes structurelles sont :
 - Les contraintes de référence
 - Les contraintes de domaine

Les clés

- La **clé** d'une table est un attribut ou groupe d'attributs dont les valeurs identifient de façon **unique** une ligne de la table
- Il est possible que plusieurs combinaisons d'attributs puissent être exploitées comme clé:
 - On choisit la combinaison la plus courte comme clé primaire
 - Les autres combinaisons sont des clés candidates
- Il est parfois utile de créer des clés artificielles lorsque les clés candidates sont compliquées

Les clés étrangères

- Employés

Matricule	Nom	DateNaiss	Salaire	NomService*

- SERVICES

NomService	Budget	Etage
Comptabilité	1M	1
Finance	1.5 M	2
Informatique	3M	2

- Dans la table Services NomService est une clé qui permet de retrouver la ligne décrivant un service donné
- Dans la table EMPLOYES NomService* est une clé étrangère qui permet d'aller chercher dans la table Service la description du service auquel appartient un employé donné

Les clés étrangères

- Le terme clé étrangère signifie clé d'une autre table
- La notion de clé étrangère est utilisée pour établir le lien sémantique entre deux tables
- Soit A une table possédant un attribut b^* défini comme clé étrangère provenant de la table B:
 - Chaque ligne de la table A possède une valeur pour la clé étrangère b^* , permettant de retrouver une **ligne** de la table B
 - Chaque ligne de la table B possède une valeur pour la clé b qui peut apparaître dans 0 ou n lignes de la table A comme valeur de b^*
 - b^* établit un lien $n-1$ entre A et B
- Les clés étrangères sont les seules redondances autorisées: plusieurs lignes dans la table A peuvent avoir la même valeur b^*

Les clés étrangères

Règles d'intégrité référentielle

- Toute valeur saisie pour une clé étrangère doit être incluse dans l'ensemble des valeurs de la clé référencée
 - Toute valeur saisie dans une colonne b^* doit exister dans la colonne b
 - Le non respect de cette règle entraîne que la table A référence des lignes n'existant pas dans la table B
- Lorsque l'on supprime une ligne dans une table référencée, la valeur de la clé supprimée doit avoir été préalablement supprimée des colonnes clé étrangères

Contrainte de domaine

- **Contraintes de domaine** : les attributs doivent respecter une condition logique

ex. $\text{PrixUni} > 0$ ET $\text{PrixUni} \leq 10000$

Valeurs nulles

- Une valeur nulle est un abus de langage pour désigner une absence de valeur d'un attribut
 - On dit aussi un nul

<u>S#</u>	SNAME	STATUS	CITY
S1		20	London
S2	Jones	10	
S3	Blake	30	Paris
S4	Clark		London
S5	Adams	30	Athens

Types de nuls

- **Valeur inconnue**
 - **Ville de fournisseur inconnue**
- **Valeur inapplicable**
 - **Fournisseur connu pour être sans statut**
- **Cette distinction est rarement appliquée en pratique**

Contrainte de référence

- C'est une contrainte exprimée entre deux relations.
- Cela consiste à vérifier que l'information utilisée dans un nuplet pour désigner un autre nuplet est valide, notamment si le nuplet désigné **existe bien**.
- Par exemple, soient les deux relations:
 - EMPLOYE(**no-ss**, nom, adresse, rôle, no-dept)
 - DEPARTEMENT(**no-dept**, nom, no-ss-chef),
- un nuplet de EMPLOYE référence un nuplet de DEPARTEMENT via l'attribut no-dept (numéro de départ)
- un nuplet de DEPARTEMENT référence un nuplet de EMPLOYE via l'attribut no-ss-chef (numéro sécurité sociale du chef de département).
- Il est important de s'assurer que les nuplets référencés existent bien.

Contrainte de référence

- On peut noter qu'un nuplet peut référencer un autre nuplet de la même relation. Dans ce cas, no-dept et no-ss-chef sont appelés des clés étrangères, puisque se sont des attributs clés d'une autre relation.
- Cette contrainte implique un ordre dans la création ou la destruction des entités, puisqu'on ne peut créer un département si le nuplet correspondant au chef n'a pas été créé et on ne peut détruire un chef de département si le département existe toujours (ou bien si on n'a pas mis un autre chef).

Conclusion

- Un ensemble de concepts bien compris et bien formalisé
- Un modèle unique, de plus en plus riche et normalisé
 - intégration des BD actives
 - intégration des BD objets
 - évolution vers un L4G standardisé
- Un formalisme qui s'étend plutôt bien
 - algèbre d'objets
- Un langage associé défini à plusieurs niveaux :
 - SQL1, 2, 3