

Chapitre 4

Modèle Relationnel

I. Introduction

Le modèle relationnel est un modèle d'informations structurées un tableau de valeurs. Les relations sont des tables avec des restrictions :

- l'ordre des lignes (tuples) n'est pas significatif ;
- l'ordre de colonnes (attributs) n'est pas significatif ;
- pas de duplication de lignes.

II. Concepts de base :

1. Domaine

Un domaine est un ensemble de valeurs que peut prendre un attribut :

- *Dnom* : chaîne de caractères de longueur maximale 30 ;
- *Djour* : {lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche} ;
- *Dmois* : {janvier, février, mars, avril, ..., octobre, novembre, décembre} ;
- *Dcouleur* : {rouge, vert, bleu, jaune, ..., violet} ;
- *Dâge* : [0,125] ;
- *Dsalaire* : \mathbb{R}^+ .

2. Relation

Une relation est définie par son schéma de relation et ses valeurs.

Schéma d'une relation : $R(A_1:D_1, \dots, A_n:D_n)$

- *R* : nom de la relation
- A_1, \dots, A_n : attributs, tous distincts
- D_1, \dots, D_n : domaines des attributs, non nécessairement tous distincts
- $(A_1:D_1, \dots, A_n:D_n)$: structure (type de tuple)

Tuple : $\{A_1:d_1, \dots, A_n:d_n\}$ un ensemble de couples <attribut,valeur>

Valeur d'une relation : sous-ensemble de tuples de l'ensemble :

$$\{ \{ A_i:d_i, \dots, A_n:d_n \} \mid d_i \in D_i \}$$

Exemple

- Schéma de la relation Etudiant :
Etudiant (N'étud:Dnum, Nom:Dnom, Prénom:Dnom, Age:Dâge)
- Une valeur de la relation Etudiant :

Age	Prénom	Nom	N°étud
21	Marc	Rochat	147
20	André	Duval	101
20	Annie	Aubry	253
19	Jean	Rochat	136

III. Base de données relationnelle

1. Introduction

Un schéma d'une BD relationnelle est un ensemble de schémas de relations auquel on ajout les contraintes d'intégrité.. Une base de données relationnelle est un ensemble de relations. Une valeur d'une BD est un ensemble de valeurs de relations.

2. Clés d'une relation

a. Définitions

Les **clés d'une relation** sont des ensembles d'attributs qui déterminent tous les autres attributs de la relation. Ainsi, les clés d'une relation ont la propriété **d'identification unique de tuples** lorsque deux tuples n'ont jamais la même valeur pour cet ensemble d'attributs.

Formellement, quelles que soient tuples de la relation R de valeurs (a,b) et (a',b') pour (A,B) :

$$a \text{ est un clé de la relation R si : } a = a' \Rightarrow b = b'$$

Une **super-clé** est un ensemble d'attributs qui identifie de manière unique un tuple de la relation. Une **clé** est une super-clé minimale, c'est-à-dire une superclé dont on ne peut enlever aucun attribut sans lui faire perdre son statut de superclé; en général, une relation a plusieurs clés (on parle de **clés candidates**).

La définition des clés est une **information intentionnelle** (appartient au schéma), donc valide pour toutes les extensions possibles de la relation

b. Clé primaire

Une **clé primaire** est une **clé privilégiée** choisie parmi les clés candidates. Ses valeurs doivent toujours être connues (sans valeur NULL). Son choix est arbitraire, peut se faire par des heuristiques (e.g. plus simple), mais elle peut être inventée pendant la phase de conception de la BD. Dans le schéma de relation, la clé primaire sera soulignée.

Exemple

Etudiant (Code_per, Nom, Prenom, Adresse, Date_nais, Lieu_nais, NASS)
 Quelles sont les clés candidates ? *Code_per, NASS, (Nom,Prenom,Adresse)*
 Quelle clé primaire choisir ? *Code_per*

c. Clé étrangère

Soit R_2 une relation, un sous-ensemble d'attributs FK dans R_2 est dit **clé étrangère** lorsque :

- il existe une relation R_1 de clé primaire PK (R_1 peut être R_2)
- à tout instant, chaque valeur de FK dans R_2 est identique à une valeur de PK dans R_1

On fera précéder les clés étrangères du caractère dièse (#).

Exemple

Etudiant(N°étud, nom, prénom, âge)
 Cours(NomC, horaire, prof)
 Suit(#N°étud, #NomC)

Propriétés

- Il n'est pas nécessaire d'avoir pour toute valeur de PK dans R_1 une valeur identique de FK dans R_2 ;
- Un attribut d'une clé étrangère peut faire partie d'une clé candidate de sa relation ;
- Chaque attribut d'une clé étrangère doit être défini sur le même domaine que son équivalent dans la clé primaire correspondante.

3. Contraintes d'intégrité

a. Définition :

Contrainte : c'est une assertion vérifiée par les données de la base à tout moment sans être définie au niveau de la structure. Les contraintes relationnelles comprennent :

- Les contraintes structurelles : liées au modèle relationnel, à l'unicité des valeurs de clés,...
- Les contraintes comportementales : liées aux applications
- Les contraintes intra-relationnelles : mettant en jeu une seule relation sur la non nullité d'un attribut
- Les contraintes inter-relationnelles : mettant en jeu plusieurs relations sur l'intégrité référentielle (existence des clés étrangères)

Remarque :

Une contrainte est tout ce que l'on souhaiterait exprimer au niveau de la structure de la BD (i.e., le schéma) et que l'on ne peut pas exprimer avec le mécanisme de description disponible dans le modèle de données.

b. Intégrité référentielle

Typiquement relationnelle, l'**intégrité référentielle** résulte de l'expression de liens avec égalité de valeurs, exprime une information de comparabilité (comme les domaines) pour deux attributs différents. Une fois déclarée la clé étrangère FK de R_2 sur R_1 :

- le SGBD vérifie toutes les insertions dans R_2 (il vérifie que la valeur de FK existe dans R_1) ; il vérifie de la même façon toutes les modifications dans FK ;
- Il vérifie toutes les suppressions de tuples dans R_1 (il vérifie qu'il n'existe pas de tuple dans R_2 référençant ce tuple à supprimer).

c. Nom spéciaux pour les contraintes

- **Intégrité de clés** : chaque relation a une clé primaire + d'éventuelles clés candidates ;
- **Intégrité de domaines** : les attributs doivent respecter la définition de leur domaine ;
- **Intégrité d'entités** : les valeurs d'une clé primaire ne peuvent être NULL ;
- **Intégrité de clés candidates** : les valeurs d'une clé doivent être uniques ;
- **Intégrité de colonnes** : contraint d'avantage l'intégrité de domaine (e.g., la date de prise de fonction du gérant est une date après 1970) ;
- **Intégrité de lignes** : concerne un tuple singulier (e.g., si *Date_Naiss* < 1950, alors salaire \geq 4000).

IV. Relations de base et relations dérivées

1. Définitions

Les **relations de base** sont celles dans le schéma commun, elles sont stockées sur disque.

Les **relations dérivées** sont produites en combinant relations de base et/ou relations dérivées par des opérations du modèle relationnel.

2. Relations dérivées : Vues

Les **vues** sont des relations virtuelles, toujours logiquement consistantes avec les relations de base sous-jacentes. Les vues simplifient les programmes d'application. Elles peuvent être utilisées pour contrôler les accès. Les vues peuvent être ou ne pas être stockées sur disque. Les stocker :

- introduit de la redondance et complique le renforcement d'intégrité ;
- accélère les requêtes et ralenti les mises à jour.

V. Opérations sur le modèle relationnel

Langage de Définition de Données (LDD) : simple

- déclarer une relation, une contrainte, des structures physiques

Langage de Manipulation de Données (LMD) :

- l'algèbre, les calculs de domaines et de tuples sont généralement considérés comme faisant partie du modèle ;
- SQL a redéfini les opérations correspondantes et défini quelques extensions, notamment pour les fonctions de groupe.

Opérations de mise à jour : simples

- création, suppression, modification d'une relation dans le schéma ;
- insertion, suppression, modification d'un tuple.

VI. Passage du schéma EA au relationnel

Le modèle EA possède deux principales constructions : entités et associations. Le modèle relationnel ne possède que des relations :

- entités ↔ relations ;
- associations ↔ relations ;
- attributs multi-valués ↔ relations.

Sans donner un algorithme, on présente ici les règles qui permettent de traduire les composants de base des schémas entité-association en composant de schémas relationnels.

1. Règle concernant les entités

Règle n°1 : Chaque entité E est exprimée sous la forme d'une relation R (d'une table).

Règle n°2 : Chaque attribut de l'entité E est traduit en attribut de la relation R (en une colonne de la table).

Règle n°3 : Chaque clé (identifiant) d'une entité E devient clé primaire de la relation R correspondante.

2. Règle concernant les associations binaires

A leur tour, les associations seront transformées en relation (tables) suivant leur cardinalité.

Règle n°4 : Une association binaire ayant N en cardinalités max des deux côtés donnera naissance à une relation (table) dont la clé primaire est la concaténation des clés primaires des tables correspondant aux entités associées. Chaque clé d'une entité associée deviendra en plus clé étrangère de la nouvelle relation.

Exemple :

Schéma relationnel correspondant :
 CLIENT (N°Client, Nom, Prénom)
 PRODUIT (Code)
 COMMANDE (#N°Client, #Code)

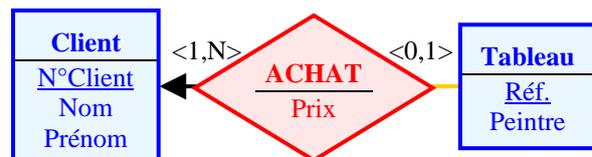


Règle n°5 : Une association binaire ayant 0,1 en cardinalités sur un des deux côtés donnera naissance ou non à une relation (à une table) suivant que le SGBD tolère ou non les valeurs NULL. Ne pas oublier de créer les clés étrangères adéquates. Si le SGBD tolère les valeurs NULL, les attributs de l'association deviennent des attributs de la relation correspondant à l'entité ayant 1 en cardinalité max.

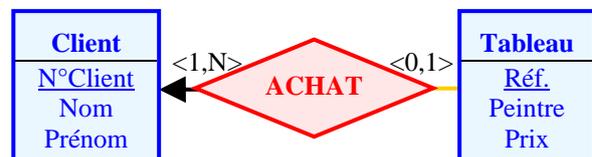
Exemple avec des cardinalités max à 1 d'un côté, et N de l'autre :

Si le SGBD ne supporte pas les valeurs NULL (noter qu'ici, *prix* dépend à la fois du client et du tableau), le schéma relationnel correspondant est :

CLIENT (N°Client, Nom, Prénom)
 TABLEAU (Réf., Peintre)
 ACHETE (#N°Client, #Réf., Prix)



Si le SGBD autorise les valeurs NULL (noter qu'ici, *prix* ne dépend que du tableau), le schéma relationnel correspondant est :



CLIENT (N°Client, Nom, Prénom)
 TABLEAU (Réf., Peintre, #N°Client, Prix)

Exemple avec des cardinalités max à 1 des deux côtés :

Si le SGBD ne supporte pas les valeurs NULL, le schéma relationnel correspondant est :

FEMME (N°SSépouse, Nom, Prénom)
 HOMME (N°SSépoux, Nom, Prénom)
 MARIER(#N°SSépouse, #N°SSépoux)



Si le SGBD autorise les valeurs NULL, le schéma relationnel correspondant est :

FEMME (N°SSépouse, Nom, Prénom, #N°SSépoux)
 HOMME (N°SSépoux, Nom, Prénom, #N°SSépouse)

Règle n°6 : Une association binaire ayant 1,1 en cardinalités sur un des deux côtés ne donnera naissance à aucune relation, mais dans l'entité qui a les cardinalités 1,1 est insérée la clé de l'autre entité. Cette clé devient clé étrangère de la relation correspondant à l'entité qui a les cardinalités 1,1.

Exemple :

Schéma relationnel correspondant :
 CLIENT (N°Client, Nom, Prénom)
 TABLEAU (Réf., Peintre, #N°Client)

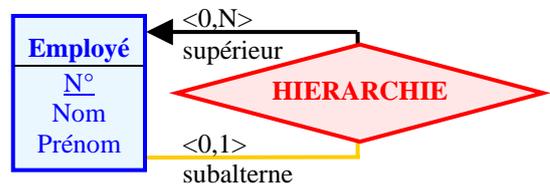


Règle n°7 : Une association réflexive est transformée en une relation quelle que soit la valeur de ses cardinalités.

Exemple :

Ce schéma EA a pour traduction en relationnel :
 EMPLOYE (N°, Nom, Prénom)
 HIERARCHIE (#N°Supérieur, #N°Employé)

Cette règle peut être nuancée en fonction des cas particuliers correspondants aux cardinalités impliquées dans l'association réflexive. Mais en faisant ainsi, on est sûr de ne pas se tromper.



3. Règle concernant les associations n-aires

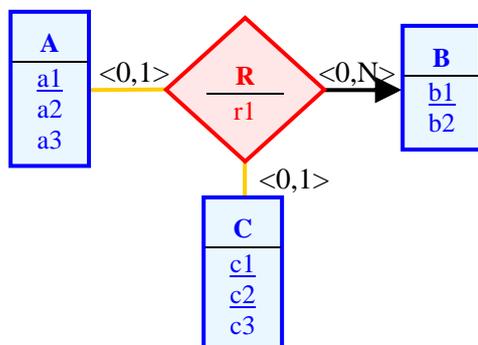
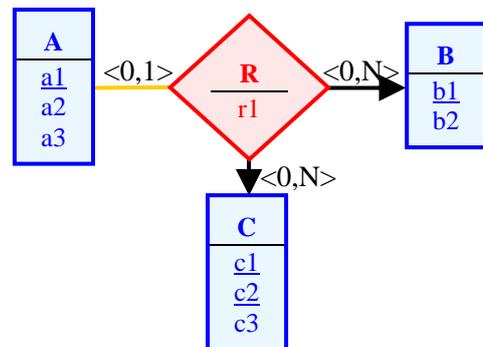
Règle n°8 : Une association n-aire (avec $n > 2$) est transformée en une relation (en une table) quelle que soit la valeur de ses cardinalités.

Si toutes les cardinalités max sont N, alors la clé de cette relation est constituée comme la concaténation des clés des relations obtenues à partir des entités associées. Sinon, chaque clé d'une entité qui a 1 en cardinalité max dans l'association est clé primaire de la nouvelle relation.

Exemple :

Le schéma EA de droite est traduit en relationnel par le schéma suivant :

A (a1, a2, a3)
 B (b1, b2)
 C (c1, c2, c3)
 R (#a1, #b1, #c1, #c2, r1)



Exemple :

Le schéma EA de gauche est traduit en relationnel par le schéma suivant :

A (a1, a2, a3)

B (b1, b2)

C (c1, c2, c3)

R (#a1, #b1, #c1, #c2, r1)

ou R (#a1, #b1, #c1, #c2, r1)

→ 2 clés primaires possibles, en choisir une.

4. Cas de l'héritage et de l'identification

a. Héritage

Exemple :

Cette deuxième notation n'est jamais employée, c'est juste pour comprendre le passage en relationnel. Ainsi on obtient le schéma relationnel suivant :

EMPLOYE (N°Employé)

PILOTE (#N°Employé, Nom, Prénom)

b. Identification

Exemple :

Cette deuxième notation n'est jamais employée, c'est juste pour comprendre le passage en relationnel. Ainsi on obtient le schéma relationnel suivant :

SERIE (Nom, Producteur)

EPISODE (Numéro, #Nom Série)

VII. Modèle Entité-Association vs Modèle Relationnel

Le modèle EA est plus compact que le modèle relationnel (meilleure intuition), mais on a une perte d'information :

- perte de précision : un seul concept relationnel (relation) représente trois concepts EA (entité, association, attribut multi-valué) ;
- la perte d'information se situe dans la correspondance avec le monde réel.