



Le modèle relationnel

1. Définition

Le modèle relationnel a été formalisé par E.F. CODD (IBM San José) en 1970. Quelques exemples de réalisation en sont :

DB2 (IBM), INFORMIX, INGRES, ORACLE.

Dans ce modèle, les données sont stockées dans des tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Un ensemble de données sera donc modélisé par un ensemble de tables.

Le succès du modèle relationnel auprès des chercheurs, concepteurs et utilisateurs est dû à la puissance et à la simplicité de ses concepts. En outre, contrairement à certains autres modèles, il repose sur des bases théoriques solides, notamment la théorie des ensembles et la logique mathématique (théorie des prédicats d'ordre 1).

2. Objectifs du modèle relationnel :

- proposer des schémas de données faciles à utiliser,
- améliorer l'indépendance logique et physique,
- mettre à la disposition des utilisateurs des langages de haut niveau pouvant éventuellement être utilisés par des non informaticiens,
- optimiser les accès à la base de données,
- améliorer l'intégrité et la confidentialité,
- fournir une approche méthodologique dans la construction des schémas.

De façon informelle, on peut définir le modèle relationnel de la manière suivante :

- Les données sont organisées sous forme de tables à deux dimensions, encore appelées relations et chaque ligne est un n-uplet ou tuple,
- les données sont manipulées par des opérateurs de l'algèbre relationnelle,
- l'état cohérent de la base est défini par un ensemble de contraintes d'intégrité.

Au modèle relationnel est associée la théorie de la normalisation des relations qui permet de se débarrasser des incohérences au moment de la conception d'une base de données.

3. Concepts de base

3.1. Relation

Une « table » structurée en colonnes fixes, et en lignes pouvant varier dans le temps en nombre et en contenu, est appelée **relation**.

Le contenu, à un instant donné, de cette table, est une table instance de cette relation.

Exemple : Relation Voiture

IdV	NomM	CatM	NbCh
19202A15	Renault	Clio	7
15422A17	Renault	Megane	8
14878A15	Mercedes	C220	10

Une relation peut être vue comme un tableau à deux dimensions dont chaque colonne correspond à un domaine et porte un nom et chaque ligne à un tuple.

Une relation n'est pas définie par des concepts positionnels, les lignes (resp. les colonnes) peuvent être permutées.

Une table instance d'une relation est un ensemble non-ordonné de tuples (lignes). Chaque tuple est composé de valeurs correspondant aux attributs (noms des colonnes) de la relation.

3.2. Domaine

Représente l'ensemble de valeurs acceptables pour un composant d'une relation. Un domaine représente le type de l'attribut et éventuellement une restriction sur le type. Ceci permettra de faire un contrôle au moment de la saisie de la donnée.

3.3. Attribut

Colonne d'une relation caractérisée par un nom et définie par un domaine. L'ordre des colonnes dans une relation est sans importance. La valeur d'un attribut est atomique et brut (non calculée) . Un attribut peut être null (ne possédant aucune valeur) sauf pour la clé.

3.4. Tuple

Soit R une relation, ayant comme ensemble d'attributs $A = \{A_1, A_2, \dots, A_n\}$

On appelle tuple défini sur R, tout ensemble t de valeurs v_1, v_2, \dots, v_n associées respectivement aux attributs A_1, A_2, \dots, A_n , avec comme seule contrainte, $v_i \in \text{dom}(A_i)$.

Autrement dit, un tuple est une ligne d'une relation, l'ajout d'un tuple ne change pas la nature de la relation. La cardinalité d'une relation R est le nombre de tuples contenus dans la relation.

Exemple :

{19202A15,"Renault","Clio",7} est un tuple de la relation voiture.

4. Schéma de relation

Le schéma d'une relation définit les propriétés de chaque attribut (nom, type, contraintes,...). Il s'agit d'un ensemble fini d'attributs et de domaines :

$$R = \{ (A_1, \text{dom}_1), \dots, (A_n, \text{dom}_n) \}$$

Avec $A = \text{attr}(R) = \{A_1, \dots, A_n\}$ désigne l'ensemble des attributs de R,
Et $\text{dom}_i = \text{dom}(A_i)$ désigne le domaine non vide de chacun des attributs A_i .

Chacun de ces domaines définit le type du contenu des colonnes qui formeront une table. Ils définissent donc des contraintes sur le contenu de chacun des tuples qui seront présents dans une instance de la relation.

Ces domaines sont, dans SQL ANSI2, toujours de type scalaire (entiers, chaînes, ...) et fini. On ne dispose d'aucun opérateur permettant de leur associer des types structurés.

Du point de vue logique, chaque relation est représentée par son nom, suivi de la liste des attributs (avec éventuellement leurs domaines).

Exemple : Soit la relation étudiant caractérisé par le cne de l'étudiant, son nom, son prénom, sa date de naissance et son numéro de groupe.

Le schéma de la relation est représenté comme suit :

Etudiant (cne , nom , prenom , dnais , nogpe)

nom de la relation : Etudiant
liste de ses attributs : cne , nom , prenom , dnais et nogpe

Au niveau physique, la relation étudiant est une table représentée comme suit :

Etudiant	Cne	nom	prenom	Dnais	Nodept
		Alaoui	Salma	12/01/96	1
		Bennani	Amine	10/02/96	2
		Lahlou	Ismail	05/08/95	1

5. Clé d'une relation

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation, et X un sous-ensemble de (A_1, A_2, \dots, A_n) , X (groupe d'attributs) est une clé si, et seulement si, :

- X est unique, elle détermine tous les attributs (A_1, A_2, \dots, A_n)
- X est minimal

La clé dans un schéma de relation est soulignée

Exemple1 : Etudiant (cne , nom , prenom , dnais , nogpe)

cne est une clé de la relation Etudiant signifie que le cne est unique cad deux étudiants différents ne peuvent jamais avoir la même valeur de cne.

Exemple2 : LigneCommande(nocom, Pref, quantité)

nocom et Pref est une clé composée, elle détermine la quantité achetée.

Une même commande (nocom) peut avoir plusieurs produits. Un produit (Pref) peut appartenir à plusieurs commandes. Par contre un produit dans une commande ne se répète pas.

6. Schéma d'une base de données relationnelle

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

Exemple : Etudiant (cne , nom , prenom , dnais , nodept)
Departement (nodept , nom , matricule)
Enseignant (matricule , nom , prenom , grade)
Matiere (nomat , titre ,coef)
Assure (matricule , nomat , periode)

Le même nom d'attribut utilisé dans deux relations distinctes véhicule le même type d'informations. (SQL ne tient pas compte de cette présupposition).

Les liens sémantiques entre tuples sont implicites à travers les valeurs de certains attributs, et non par adresse ou pointeur comme dans des modèles navigationnels.

7. Contrainte d'intégrité

Une propriété du schéma, invariante dans le temps. Il existe différents types de contraintes d'intégrité:

- liées au modèle (pas de doublons dans une relation.);
- de domaine *Exemple* : une note doit être comprise entre 0 et 20
- clé primaire dont la valeur est unique dans sa relation.
- référentielles dites de clé étrangère qui impose que la valeur d'attribut de la relation r_1 apparaît comme valeur de clé dans une autre relation r_2 .

Exemple : Etudiant (**cne** , nom , prenom , dnais , nodept)
 Département (**nodept** , nom , matricule)

nodept est une clé primaire dans la relation Département, et une clé étrangère dans la relation Etudiant. La valeur de cet attribut n'existe dans la table étudiant que s'il est déjà défini dans la table Département.

Exercice : Vérification des contraintes d'intégrités

Un centre de formation offre des cours du soir aux employés de différentes sociétés intéressées.

Un cours appartient à un thème particulier, est assuré par un professeur pendant un nombre d'heures fixé.

Chaque heure a un tarif, le prix du cours est le tarif de l'heure * nombre d'heures.

Les employés peuvent s'inscrire à plusieurs cours.

Le schéma normalisé de la base de données est le suivant :

Cours (codeCours, niveau, dateCours, tarifHeure, codeTheme, codeProf, nbHeures)

Thème (codeTheme , titre)

Employé (codeEmploye , nom , prénom , ville , codeSociété)

Société (codeSociété , nom , ville)

Professeur (codeProf , nom , prénom)

Inscrit (codeEmploye , codeCours)

On suppose un extrait des tables Société et Employé :

Société	CodeSociété	Nom	Ville
	10	Atlas Mega	Casa
	12	Metal XX	Fès
	50	FSJES	Fès

Employé	CodeEmploye	Nom	Prénom	ville	CodeSociété
	123	Chami	Salim	Fès	12
	136	Abbad	Amal	Fès	50
	133	Slaoui	Saad	Casa	15
	123	Slimani	Mohamed	Rabat	50

1. Donner la clé primaire et les clés étrangères de chaque relation.
2. Donner et expliquer la liste des règles d'intégrité non respectées dans l'exemple ci-dessus.
3. Donner la liste des contraintes d'intégrités vérifiées par le système lors de l'insertion d'un tuple dans la table Employé.
4. Donner un ordre possible pour procéder à l'insertion des tuples dans les différentes tables.

Solution

1. Donner la clé primaire et les clés étrangères de chaque relation.

Relation	Clé primaire	Clé étrangère1	Clé étrangère2
Cours	CodeCours	CodeThème	CodeProf
Thème	CodeThème		
Employé	CodeEmployé	CodeSociété	
Société	CodeSociété		
Professeur	CodeProf		
Inscrit	CodeEmployé, CodeCours	CodeEmployé	CodeCours

2. Donner et expliquer la liste des règles d'intégrité non respectées dans l'exemple ci-dessus.

Solution

Dans la table Employé :

- La contrainte de clé primaire non respectée; l'employé de code 123 est dupliqué.
- La contrainte d'intégrité référentielle non respectée; le codeSociété 15 n'existe pas dans la table Société.

3. Donner la liste des contraintes d'intégrités vérifiées par le système lors de l'insertion d'un tuple dans la table Employé.

Solution :

Pour insérer un tuple dans la table employé suivante:

Employe (codeEmploye , nom , prénom , ville , codeSociété)

Le système vérifie la contrainte de clé primaire : non duplication du codeEmployé

Et la contrainte d'intégrité référentielle: la présence de la valeur codeSociété dans la table Société.

4. Donner un ordre possible pour procéder à l'insertion des tuples dans les différentes tables.

Solution

Il s'agit de donner une valeur à la clé primaire avant de l'insérer comme clé étrangère.

Un ordre possible :

1. Thème, Société, professeur
2. Employé, Cours
3. Inscrit