

Chap. 7 – Modèle relationnel

Exemple de la bibliothèque/médiathèque

Si on imagine le fonctionnement d'une bibliothèque, on réalise que la gestion des personnes, des objets, mais aussi des liens qui les relie, est complexe.

- il faut lister les personnes, les objets et les relations (usager, livre, emprunt, employé, etc.)
- pour chaque élément de la liste, il faut choisir les données pertinentes et donc procéder à une approximation des données : c'est la **modélisation**

Lors du fonctionnement du système, il faut fournir un certain nombre de services :

- garantir l'absence d'erreur
- permettre l'ajout/suppression de données
- gérer les droits (car un usager ne doit pas avoir les mêmes possibilités qu'un employé)
- etc.

C'est dans ce but qu'ont été mis en place les **systèmes de gestion de bases de données**.

7.1 – Le modèle relationnel

Le modèle relationnel est une façon de représenter et de manipuler les bases de données.

Ce modèle se fonde sur quelques principes essentiels

- une assise mathématique forte sur la théorie des ensembles
- une implémentation informatique très proche des mathématiques

- une formulation des énoncés sous forme de termes **logiques**
- une ensemble d'opérations élémentaires que l'on peut enchaîner pour créer des requêtes complexes

Modèle de données, structure de données

Attention, le chapitre *modèle de données* n'est pas lié au chapitre *structure de données* :

- structure de données : s'intéresse à la façon d'organiser les données en machine (tableaux, piles, files, objets, etc.)
- modèle de données : s'intéresse à information contenue dans ces données. L'implémentation machine est complètement occultée par le SGBD.

Ce sont donc deux niveaux d'**abstractions** différents. Le modèle de donnée est d'un niveau plus élevé.

7.2 – Modélisation relationnelle des données

Le modèle relationnel a été inventé par **Edgar Cobb**, ingénieur travaillant pour IBM, dans les années 1970.

Son travail théorique (mathématique) a prouvé la forte cohérence de son modèle ainsi que sa puissance.

L'implémentation de ce modèle quelques années plus tard a donné lieu à un outil robuste et efficace qui a dominé le monde professionnel à partir des années 1980.

Les bases de données sont partout : la gestion d'une entreprise (salariés, payes, gestion, comptabilité, etc.), des produits (stocks, entrées, sorties, valeurs, etc.), des clients (paniers, cartes fidélités, etc.) ou encore actuellement des catalogues (photos, e-mails, musiques, etc.) ou des réseaux sociaux

Principes généraux

Pour énoncer les principes du modèle relationnel, nous allons en détailler et décrire ses **constituants** qui sont :

- nom
- domaine
- attribut
- entités
- relation
- schéma
- base de donnée

- objet (**entité**) : n-uplet
 - (`"Introduction à l'algorithmique" : String, "Thomas Cormen" : String, "Dunod" : String, 01/12/1994 : Time, "2-10-001933-3" : String`)

- collection (**relation**) : ensemble de nuplets
 - { (`"Introduction à l'algorithmique", "Thomas Cormen", "Dunod", 01/12/1994, "2-10-001933-3"`), (`"Python pour le data scientist", "Emmanuel Jakobowicz", "Dunod", 03/03/2021, "978-2-10-081224-0 "`), (`"Astérix Tome 39", "Didier Conrad", "Albert René (Editions)", 21/10/2021, "9782864973492"`), ... }

- **schéma** : pour chaque **attribut** il faut un nom et un **domaine**
(*titre* : String, *auteur* : String, *editeur* : String, *date* : Time, *isbn* : String)

- **BDD** = ensemble des relations

Modéliser

.

1. déterminer entités à modéliser (objets, personnes, actions, etc.)
2. modéliser les entités par une relation => schéma
3. définir les contraintes

D'autres modèles

.

- paradigmes réseaux, objets, etc.

Contraintes d'intégrité

.

- invariants : prop logique préservée
- garantit la cohérence des données

Contrainte de domaine .

- choix du domaine importante
 - il ne faut pas de perte d'information
 - il faut limiter les valeurs illégales

Contrainte d'entité .

- unicité de chaque élément
- non ambiguë
- clé primaire

Contrainte de référence .

- clés étrangères

Contrainte utilisateur

.

- droits d'accès

7.3 Activités

Personne : (*id* : *Int*, *nom* : *Str*, *prnom* : *Str*)

Annuaire : (*id_personne* : *Int*, *tel* : *Int*)

Numéro : (*tel* : *Int*)

On souhaite modéliser un annuaire téléphonique simple dans lequel chaque personne (identifiée par son nom et son prénom) est associée à son numéro de téléphone.

Proposer une modélisation relationnelle de cet annuaire.

On considère la solution donnée pour l'activité précédente. Dire si chacun des ensembles est une relation valide pour le schéma Annuaire.

1. {}
2. {('Titi', 'Toto', '0123456789')}
3. {('Titi', 'Toto', '0123456789'), ('Doe', 'John', '0123456789')}
4. {('Titi', 'Toto', '0123456789'), ('Titi', 'Toto', '987654343210')}
5. {('Titi', 'Toto', '0123456789'), ('Doe', 'John')}
6. {('Titi', 'Toto', 42)}

Donner la modélisation relationnelle d'un bulletin scolaire.

Cette dernière doit permettre de mentionner

- des élèves, possédants un numéro d'étudiant alphanumérique unique
- un ensemble de matières fixées, mais qui ne sont pas données
- au plus une note sur 20, par matière et par élève.

On prendra soin de préciser toutes les contraintes utilisateurs qui ne peuvent être inscrites dans les schémas des relations.

On considère la solution donnée pour l'activité précédente. Dire si chacun des ensembles est une relation valide pour le schéma de la base de données du bulletin de notes.

1.
 - Eleve = {}
 - Matiere = {}
 - Note = {}
2.
 - Eleve = {('Titi', 'Toto', 'AB56789'), }
 - Matiere = {('NSI', 0), ('Sport', 1)}
 - Note = {('AB56789', 1, 17)}
3.
 - Eleve = {('Titi', 'Toto', 'AB56789'), }
 - Matiere = {('NSI', 0)}
 - Note = {('AB56789', 1, 17)}
4.
 - Eleve = {('Titi', 'Toto', 'AB56789'), }
 - Matiere = {('NSI', 0), }
 - Note = {('AB56789', 0, 17), ('AB56789', 0, 18)}
5.
 - Eleve = {('Titi', 'Toto', 'AB56789'), }
 - Matiere = {('NSI', 0), ('Sport', 1)}
 - Note = {('AB56789', 0, 17), ('AB56789', 1, 17)}

Modéliser des informations sur les départements français.

Pour chaque département on veut pouvoir stocker son nom, son code, son chef-lieu et la liste de tous les départements voisins. Attention, les codes de département sont tous des nombres, sauf la Corse du Sud et la Haute Corse qui ont les codes 2A et 2B respectivement. Les départements d'Outre-Mer ont un code sur trois chiffres (de 971 à 976).

Proposer une contrainte utilisateur permettant d'éviter la redondance d'information dans la liste des voisins.

Proposer une modélisation pour un réseau de bus. Cette dernière doit être suffisamment riche pour permettre de générer, pour chaque arrêt de bus du réseau, une fiche horaire avec tous les horaires de passage de toutes les lignes de bus qui desservent l'arrêt.

Indication : ici, plus qu'une simple traduction du français vers le modèle relationnel, on essayera de déterminer dans un premier temps quelles informations sont pertinentes et comment les représenter. On pourra ensuite procéder à la modélisation sous forme de relations.

On considère deux relations $R(\underline{a \text{ Int}}, b \text{ Int}, c \text{ Int})$ et $S(\underline{a \text{ Int}}, e \text{ Int})$ où l'attribut a de S est une clé étrangère faisant référence à a de R .

Dire si les affirmations suivantes sont vraies ou fausses, en justifiant.

1. Les a de R sont tous deux à deux distincts.
2. Les b de R sont tous deux à deux distincts.
3. Les a de S sont tous deux à deux distincts.
4. Les e de S sont tous deux à deux distincts.
5. S peut être vide alors que R est non vide.
6. R peut être vide alors que S est non vide