
Modèle relationnel

BARBIER J.M. - LGT Dumezil - NSI

20 janvier 2026

Table des matières

1	Introduction	1
1.1	Programme	1
2	Exemple : bibliothèque	2
2.1	Activité	2
2.2	Modèle	2
2.3	Contraintes	2
3	Le modèle relationnel	2
3.1	Généralité	2
3.2	Le modèle relationnel	3
3.3	Exemple	3
3.4	Schéma de relation	3
3.5	Schéma de base de données	3
4	Modélisation relationnelle des données	4
4.1	Principes généraux	4
4.2	Contraintes d'intégrité	4
4.3	Contrainte de domaine	4
4.4	Contrainte d'entité (ou contrainte de relation)	4
4.5	Contrainte de référence (intégrité référentielle)	5
4.6	Contraintes utilisateurs	5
5	Exercices	5
5.1	Exercice 1	5
5.2	Exercice 2	5
5.3	Exercice 3	5

1 Introduction

1.1 Programme

Notions : modèle relationnel : relation, attribut, domaine, clef primaire, clef étrangère, schéma relationnel.

Capacités attendues : identifier les concepts définissant le modèle relationnel.

Remarques : ces concepts permettent d'exprimer les contraintes d'intégrité (domaine, relation et référence)

2 Exemple : bibliothèque

2.1 Activité

Emprunt d'un livre dans une bibliothèque :

- Alice trouve un livre
- elle scanne sa carte : "Alice : aucun prêt en cours"
- scanne le livre : "Alice : emprunt en cours : "Programmation en pratique, xxxxx, année, ISBN""
- Bob vient rendre un livre
- scanne sa carte : "Bob, un livre en cours de prêt"
- scanne le livre : "Bob : rendu : "Le python pour les nuls, xxxx, année, ISBN""

Illustration des caractéristiques d'un **système d'information**.

2.2 Modèle

Contient :

- des descriptions d'objets (livres)
- des descriptions de personnes physiques (utilisateurs)
- des processus (emprunt / restitution)

Chaque description d'objet est un **modèle** de la réalité : le système ne retient que ce qui est nécessaire au bon fonctionnement des processus (la couleur des yeux d'Alice est inutile pour le fonctionnement d'une bibliothèque).

2.3 Contraintes

En plus du **stockage** es objets (livres, personnes, emprunts, ...) le système doit garantir l'absence d'erreurs.

- si un livre est rendu sans avoir été scanné, on ne peut pas l'emprunter
- si un livre a été rendu, on doit pouvoir le scanner
- certaines personnes (documentalistes) doivent pouvoir ajouter / retirer / modifier / réserver des livres (niveaux d'accès)
- quelle que soit la borne utilisée, la procédure doit être valable => données centralisées, les bornes sont des **clients** permettant de consulter l'information.

3 Le modèle relationnel

3.1 Généralité

De manière générale, un modèle de données est une représentation de concepts que l'on souhaite étudier.

Un modèle permet d'énoncer des *propriétés* des données en termes *logiques*, et permet de *programmer* des processus réels complexes sous forme d'opérations *élémentaires* sur les données.

3.2 Le modèle relationnel

Défini en 1970 par Edgar F. Codd, employé à IBM.

- un objet modélisé (une **entité**) est représenté par un *n-uplet* de valeurs *scalaires*¹
- une collection d'objets est représentée par des *ensembles*² de n-uplets

Modèle de données != structure de données. La structure de données est le choix d'une implémentation pour le modèle de données.

3.3 Exemple

Un livre peut être représenté par un quintuplet : ("Titre du livre", "Auteur", "Editeur", année, "ISBN")

L'ensemble des livres de la bibliothèque est alors représenté par un *ensemble* de n-uplets, appelé une **relation**.

```
livres = [  
    ("Titre 1", "Auteur 1", "Editeur 1", 1988, "000-0000000001"),  
    ("Titre 2", "Auteur 3", "Editeur 2", 1987, "000-0000000002"),  
    ("Titre 3", "Auteur 1", "Editeur 2", 2015, "000-0000000003"),  
    ("Titre 4", "Auteur 2", "Editeur 1", 2001, "000-0000000004"),  
]
```

3.4 Schéma de relation

Chaque relation se conforme à un *schéma* : description du **nom** et du **domaine** de chaque composant (**attribut du schéma**).

Exemple : les éléments de la *relation* Livre possèdent 5 *attributs*

- titre : titre du livre, une chaîne de caractères
- auteurs : le ou les auteurs, une chaîne de caractères
- éditeur : l'éditeur, une chaîne de caractères
- année : année de publication, un entier
- isbn : identifiant du livre, une chaîne de caractères de 13 chiffres

On peut aussi noter le *schéma* de la manière suivante :

```
Livre(titre: str, auteur: str, editeur: str,  
      annee: int, isbn: str)
```

3.5 Schéma de base de données

La **base de données** est un **ensemble de relations**. Par extension, on appelle *schéma d'une base de données* l'ensemble des schémas des relations constituant la base.

Exemple : la bibliothèque pourrait avoir 3 relations : Livres, Utilisateurs, Emprunts

1. **scalaire** : une valeur qui ne peut pas être décomposée en d'autres valeurs.
2. **ensemble** : collection non ordonnée d'éléments distincts.

4 Modélisation relationnelle des données

4.1 Principes généraux

La modélisation se décompose en plusieurs étapes :

- déterminer les entités que l'on souhaite manipuler
- modéliser les ensembles d'entités comme des relations en donnant leur schéma, en choisissant le bon domaine pour chaque attribut
- définir les *contraintes* de la base de données = l'ensemble des *propriétés logiques* que les données doivent vérifier à chaque moment

4.2 Contraintes d'intégrité

Une contrainte d'intégrité est une propriété logique, préservée à tout instant par la base de données, et qui garantit la *cohérence des données*. On distingue 4 grandes catégories de contraintes d'intégrité

- contraintes de domaine
- contrainte d'entité
- contrainte de référence
- contrainte utilisateur

4.3 Contrainte de domaine

Comment le choix de l'un domaine pour un attribut permet de garantir certaines propriétés ? Ou comment faire les bons choix de domaine(s) pour représenter une ou des propriétés pour garantir leurs propriétés ?

Exemples :

- pour une adresse : str ou décomposer numéro / rue / cp..
- pour un âge : str / int / Date naissance décomposée ?
- pour un choix (abonné / non abonné) : str / int / booléen ?

Le choix des domaines doit répondre à deux impératifs :

- permettre de représenter exactement et sans perte d'information toutes les valeurs possibles pour un attribut
- limiter autant que possible la saisie de valeurs illégales ou mal formées

4.4 Contrainte d'entité (ou contrainte de relation)

La contrainte d'entité (ou contrainte de relation) permet de s'assurer que chaque élément d'une relation est **unique** et identifie une entité de manière **non ambiguë**.

Exemple : relation *Usager* où on ne prendrait que "nom" et "prénom" comme attribut...

Chaque relation doit être identifiée par une **clef primaire, unique** et **non nulle**.

La clef primaire peut être un attribut ou une combinaison d'attributs. Le système garantit l'unicité de la clef primaire, donc en fonction des choix de clef primaire, certains choix seront impossibles (exemple : si clef primaire = email, deux personnes ne peuvent pas partager le même email)

4.5 Contrainte de référence (intégrité référentielle)

Les *clefs primaires* permettent de servir de **référence** dans une autre relation.

Exemple : un n-uplet emprunt dans la bibliothèque contient deux “liaisons” entre entités (une personne et un livre)

On utilise pour cela une **clef étrangère** (ex : code-barre); la clef étrangère doit être une valeur pour cet attribut existant dans l’autre relation - on utilise pour cle la clef primaire).

La contrainte de référence empêche aussi la suppression d’éléments reliés : on ne pourrait pas supprimer un livre si il est emprunté, car alors la clef étrangère du n-uplet emprunt correspondant serait cassée (violation de la contrainte de référence).

Les clefs étrangères permettent de créer des relations de type *un à plusieurs* ou *plusieurs à plusieurs*

Exemple : ville de naissance d’une personne / auteur d’un livre / auteurs d’un livre / ingrédients d’une pizza.

4.6 Contraintes utilisateurs

Les contraintes utilisateur (ou contrainte métier) sont toutes les contraintes qui ne peuvent pas être exprimées par les précédentes.

Exemples :

- l’âge d’une personne ne doit pas être supérieur à 150
- une adresse email doit contenir le caractère @
- un code postal contient exactement 5 caractères, tous des chiffres

5 Exercices

5.1 Exercice 1

Proposer une modélisation pour une bibliothèque de recettes : chaque recette comporte des ingrédients avec une certaine quantité, un texte, une catégorie (desserts, entrée, etc..) et zéro ou plusieurs “badges” (économique, végétarien, hiver, été, etc..)

5.2 Exercice 2

Proposer une modélisation des départements et régions de France. Pour chaque département et région, on veut pouvoir stocker son nom, son code (27, 91, mais aussi 2A/2B et 971 à 976 pour les DOM), son chef-lieu, et la liste de tous ses voisins de même type, ainsi que la région d’appartenance pour les départements.

5.3 Exercice 3

Proposer une modélisation pour un catalogue en ligne basique, qui présente des vêtements par catégorie et sous-catégorie (hiérarchie à 2 niveaux). Chaque vêtement peut avoir plusieurs tailles et couleurs, et le prix dépend du vêtement, de la taille et de la couleur.