

# Modèle relationnel et conception de bases de données

## I. Modèle relationnel [GARD]

### A. Organisation des données en tables

Contexte historique 1 Le modèle relationnel a été introduit par E. F. Codd en 1969, quand il travaillait au centre de recherche d'IBM San-José.

Objectifs 2 Le modèle relationnel vise à fournir un système de manipulation de données qui

- ▶ Permette un indépendance entre applications manipulant les données et représentation interne des données
- ▶ Aide à gérer les défis de cohérence et de redondance des données

Intuition et terminologie 3 Le modèle relationnel organise les données en tables, appelées **relations**. Les lignes de ces tables sont des  $n$ -uplets appelées **enregistrements**. Les colonnes de ces tables sont appelées **attributs** et associées à des ensembles de valeurs possibles appelés **domaines**.

Définition 4 Un **domaine** est un ensemble de valeurs. On peut les rapprocher de la notion de **type** en programmation.

Exemple 5 INTEGER l'ensemble des entiers et VARCHAR l'ensemble des chaînes de caractères sont des domaines.

Définition 6 Un **attribut** est un domaine, auquel on associe un nom et une sémantique dans un contexte particulier.

Exemple 7 VARCHAR avec le nom « Pays » représentant le nom d'un pays est un attribut, tout comme INTEGER avec le nom « Population » représentant le nombre d'habitants.

Définition 8 **Schéma relationnel** Un schéma relationnel est une liste d'attributs, caractérisée par un nom.

Exemple 9 Un exemple de schéma relationnel est « Démographie » mettant en relation l'attribut « Pays » de domaine VARCHAR et l'attribut « Population » de domaine INTEGER.

Définition 10 **Relation**. Étant donné un schéma relationnel avec des domaines  $D_1, \dots, D_n$ , une **relation** qui suit ce schéma est un sous-ensemble du produit cartésien  $D_1 \times \dots \times D_n$ .

Exemple 11 La relation suivante suit le schéma "Démographie" ("Pays": VARCHAR, "Population": INTEGER):

Démographie :	Pays	Population
	France	68 000 000
	Belgique	12 000 000

Définition 12 Chacun des éléments d'une relation est appelé un **enregistrement**. On parle parfois aussi de **n-uplet**.

Remarque 13 **Absence de doublons**. Puisque nous avons utilisé des ensembles pour définir les relations, cela implique qu'il n'y a pas de doublons: chaque enregistrement apparaît au plus une fois. Cependant ceci n'est pas toujours le cas en pratique.

Définition 14 La valeur spéciale NULL sert à indiquer quand une valeur dans un enregistrement est manquante ou non-renseignée.

Définition 15 **Contraintes** On peut poser des contraintes d'intégrité et de cohérence sur les données. Par exemple en exigeant qu'un attribut en particulier ne puisse pas être NULL.

Définition 16 Une **clef** d'une relation est un sous-ensemble d'attributs permettant d'identifier de façon unique chaque enregistrement. Quand on crée une relation on choisit une clef en particulier qu'on appelle **clef primaire**.

Exemple 17 Dans la relation "Démographie", l'attribut "Pays" est une clef: chaque pays a au plus une ligne dans cette relation. La paire d'attributs ("Pays", "Population") est aussi une clef. On peut choisir "Pays" en tant que clef primaire.

Remarque 18 La notion de **clef** est contextuelle. Dans une relation suivant le schéma "Hydrographie"("Fleuve": VARCHAR, "Pays": VARCHAR) qui à un fleuve associe le pays où il se trouve, l'attribut "Pays" n'est pas une clef, car un pays peut avoir plusieurs fleuves.

**Définition 19** Une **contrainte référentielle** consiste à imposer qu'un des attributs d'une table ne contienne que des valeurs présentes dans un autre attribut d'une autre table. On parle également de **clef étrangère**.

## II. Algèbre relationnelle

**Définition 20** Une **algèbre relationnelle** est un ensemble d'opérations prenant en entrée une ou deux relations pour construire une nouvelle relation.

**Remarque 21** Plusieurs **algèbres relationnelles** équivalentes existent (SPC, SPJR, etc) [ABI]

**Exemple 22** Il est possible de construire une relation « Pays ayant plus de  $n$  habitants » à partir de la relation « Démographie » au moyen d'opérations de l'algèbre relationnelle.

**Application 23** Les **systèmes de gestion de base de données** (SGDB) interprètent des requêtes en langage SQL vers une représentation proche de l'algèbre relationnelle. Il est alors plus simple d'optimiser les requêtes sous cette forme.

**Théorème 24** Codd (admis) L'algèbre relationnelle est équivalente au calcul relationnel. Toute requête du calcul relationnel peut donc se traduire en algèbre relationnelle et vice-versa.

### A. Opérations ensemblistes

**Définition 25** L'**union** est une fonction prenant deux relations de même schéma et renvoyant une relation de même schéma contenant l'union des  $n$ -uplets des deux relations.

**Notation 26** L'**union** entre deux relations  $A$  et  $B$  se note  $A \cup B = \{t \mid t \in A \vee t \in B\}$ .

**Définition 27** L'**intersection** est une fonction prenant deux relations de même schéma et renvoyant une relation contenant les  $n$ -uplets communs à ces deux relations.

**Notation 28** L'**intersection** entre deux relations  $A$  et  $B$  se note  $A \cap B = \{t \mid t \in A \wedge t \in B\}$ .

**Définition 29** Le **produit cartésien** est une fonction prenant deux relations et renvoyant une relation ayant pour schéma la concaténation de ceux des deux relations, et pour  $n$ -uplets le produit cartésien des  $n$ -uplets de ces relations.

**Notation 30** Le **produit cartésien** entre deux relations  $A$  et  $B$  se note  $A \times B = \{(r, s) \mid r \in A \wedge s \in B\}$ .

**Définition 31** La **différence** est une fonction prenant deux relations de même schéma et renvoyant une relation ayant ledit schéma en gardant uniquement les  $n$ -uplets apparaissant dans la première relation mais pas la seconde.

**Notation 32** La **différence** entre deux relations  $A$  et  $B$  se note  $A - B = \{t \mid t \in A \wedge t \notin B\}$ .

### B. Opérations spécifiques

**Définition 33** La **projection**, notée  $\pi_{a_1, \dots, a_n}$ , est une opération unaire sur les relations. Elle permet d'extraire un sous-ensemble d'attributs, renvoyant donc une relation de schéma  $(a_1, \dots, a_n)$ .

**Définition 34** Le **renommage** renomme un attribut d'une table. Il peut être utile par exemple pour appliquer une opération d'union.

**Définition 35** La **sélection**, notée  $\sigma_P A = \{r \in A \mid P(r)\}$  est une fonction unaire sur les relations renvoyant une relation ne contenant que les  $n$ -uplets satisfaisant un prédicat  $P$ .

**Définition 36** La **jointure**, notée  $\bowtie_P$  regroupe deux relations en ne gardant que les  $n$ -uplets satisfaisant le prédicat binaire  $P$ .

**Remarque 37** La **jointure naturelle** est une jointure fréquente dont le prédicat est l'égalité sur les attributs ayant le même nom.

**Définition 38** La **division**, notée  $A \div B$ , prends en entrée deux relations de schémas  $A(x_1, \dots, x_n, y_1, \dots, y_n)$  et  $B(y_1, \dots, y_n)$ , puis retourne la relation  $A \div B = \{t \mid \forall s \in B, (t, s) \in A\}$ , où  $(t, s)$  est la décomposition d'un  $n$ -uplet de  $A$  avec  $t$  de schéma  $(x_1, \dots, x_n)$  et  $s$  de schéma  $(y_1, \dots, y_n)$ .

## Pratique 39 Opérateurs SQL

Algèbre relationnelle	SQL
Projection	SELECT
Renommage	AS
Sélection	WHERE
Jointure	JOIN / NATURAL JOIN

**Extension 40** Opérateurs d'agrégats [SIL] Les opérations SQL comme GROUP BY ou AVG ne s'expriment pas naturellement en algèbre relationnelle. On peut cependant ajouter une opération d'agrégat  $\mathcal{G}_f$  qui applique une fonction d'agrégat  $f$  à une relation.

DEV

### III. Conception d'une base de données

**Motivation 41** La conception d'une base de données demande de représenter des interactions parfois nombreuses et complexes. Afin de simplifier cette tâche, plusieurs méthodes existent pour représenter et formaliser ces interactions.

#### A. Méthodes graphiques [GARD]

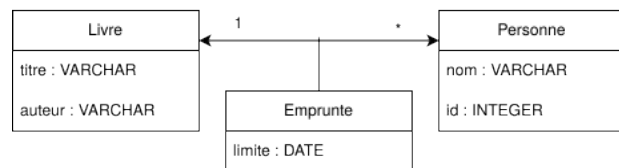
**Intuition 42** Le modèle entité-association est un modèle graphique permettant d'organiser visuellement les données et leur liens.

**Définition 43** Une entité est une modélisation d'un objet à l'aide d'un type défini par un nom ainsi qu'une liste d'attributs.

**Définition 44** Une association est un lien logique entre les entités modélisées par un verbe et une éventuelle liste d'attributs.

**Définition 45** La cardinalité d'une association indique le nombre d'associations possibles d'une entité vers une autre pour une association donnée. Par convention, on note \* pour une cardinalité de zéro ou plus, et  $n$  pour une cardinalité d'exactly  $n$ . La cardinalité doit être précisée dans les deux sens.

#### Exemple 46



**Méthode 47** Pour transformer un modèle graphique en schéma relationnel on représente les entités et associations par des tables :  
► chaque entité est représentée par une table de même nom ayant pour attributs la liste des attributs de l'entité.  
► chaque association est représentée par une table de même nom ayant pour attributs la liste des clés primaires des entités associées et les attributs propres à l'association.

**Remarque 48** Une association ayant pour cardinalité 1 – 1 peut être regroupée en une seule table, où l'association est attachée par jointure à la clé primaire de chaque entité.

**Application 49** La méthode ECS (Entités-Composants-Systèmes) adapte une méthode entité-relation dans le contexte d'applications interactives comme les jeux vidéo ou les simulations physiques.

#### B. Formes normales

**Motivation 50** Certains choix de schémas relationnels peuvent introduire de la redondance entre les données. La redondance introduit un coût en mémoire et en performances, et peut affecter la cohérence des données, par des anomalies de mise à jour.

**Définition 51** Il y a une dépendance fonctionnelle entre un ensemble d'attributs  $X$  et un ensemble d'attributs  $Y$ , notée  $X \rightarrow Y$  si à chaque valeur de  $X$  il n'existe qu'une valeur de  $Y$ .

DEV

**Remarque 52** Boyce et Codd ont proposé des transformations vers des formes normales, assurant de bonnes propriétés comme par exemple la réduction de la redondance d'informations. Nous présenterons les 3 premières formes normales, ainsi que la forme normale de Boyce-Codd.

<b>Modèle relationnel et conception de bases de données</b>	
<b>I. Modèle relationnel [GARD]</b>	
<b>A. Organisation des données en tables</b>	
1	Contexte historique Le modèle relationnel
2	Objectifs Le modèle relationnel
3	Intuition et terminologie Le modèle relationnel
4	Def Un domaine
5	Ex Domaines
6	Def Un attribut
7	Ex Attributs
8	Def Schéma relationnel
9	Ex Schéma relationnel
10	Def Relation.
11	Ex La relation suivante
12	Def Enregistrement ou n-uplet
13	Rem Absence de doublons.
14	Def La valeur spéciale NULL
15	Def Contraintes
16	Def Une clef
17	Ex Clef primaire
18	Rem La notion de clef est contextuelle.
19	Def Une contrainte référentielle
<b>II. Algèbre relationnelle</b>	
20	Def Une algèbre relationnelle
21	Rem Plusieurs algèbres relationnelles
22	Ex
23	App Les systèmes de gestion de base de données (SGDB)
24	Thm Codd (admis)
<b>A. Opérations ensemblistes</b>	
25	Def L'union
26	Not L'union
27	Def L'intersection
28	Not L'intersection
<b>B. Opérations spécifiques</b>	
29	Def Le produit cartésien
30	Not Le produit cartésien
31	Def La différence
32	Not La différence
33	Def La projection,
34	Def Le renommage
35	Def La sélection,
36	Déifiniton La jointure,
37	Rem La jointure naturelle
38	Def La division,
39	Prat Opérateurs SQL
40	Extension Opérateurs d'agrégats
<b>III. Conception d'une base de données</b>	
41	Motiv
<b>A. Méthodes graphiques [GARD]</b>	
42	Intuition Le modèle entité-association
43	Def Une entité
44	Def Une association
45	Def La cardinalité
46	Ex
47	Métho Pour transformer un modèle graphique en schéma relationnel
48	Rem
49	App La méthode ECS (Entités-Composants-Systèmes)
<b>B. Formes normales</b>	
50	Motiv
51	Def Il y a une dépendance fonctionnelle
52	Rem Formes normales

## Programme 53

- TERM 3 : Modèle relationnels, contraintes, structure vs contenu, services SGBD, SQL
- PREP 21 : clefs primaires et étrangères, agrégation
- COMP 2 : algèbre relationnelle et optimisation de requêtes
- COMP 12 C : calcul relationnel et théorème de Codd (/!\ programme complémentaire 3B)

## Bibliographie

[GARD] G. Gardarin, *Bases de données*.

[ABI] S. Abiteboul & R. Hull & V. Vianu, *Foundations of Databases*.

[SIL] A. Silberschatz, *Database System Concepts, 6th Edition*.