

SMI6U05L : Bases de données

Le modèle relationnel

Luigi Santocanale
LIS, Aix-Marseille Université

Contenu basé sur le cours de Rémi Eyraud

Plan

Introduction

Domaines, relations, attributs

Algèbre Relationnelle

Plan

Introduction

Domaines, relations, attributs

Algèbre Relationnelle

Historique

- Travaux de Edgar F. Codd 1970
- Codd (IBM) :
langage SEQUEL (Structured English Query Language)
- Oracle :
langage SQL (Structured Query Language)

Premiers objectifs

- Idée : utiliser un **modèle ensembliste** pour décrire et manipuler un ensemble d'enregistrements.
- Objectifs :
 - ▶ **Indépendance** entre programmes d'application et représentation interne
 - ▶ **Base théorique** solide pour la cohérence et la non-redondance des données.

- **SGBD Relationnel** : respecte les 12 règles de Codd (voir l'appendice).
- Principales raisons du succès :
 - ▶ simplicité d'utilisation
 - ▶ bases théoriques solides
 - ▶ **normalisations** (standardisation) de SQL (1986, 1992, 1999, 2003, 2008, 2011, 2016).

Avant de commencer...

- Dans ce chapitre : **modèle relationnel**.
Il s'agit de l'ensemble des principes théoriques supportant le développement des SGBDR.
- La maîtrise de ce domaine, **avant même SQL**, facilite l'adaptation à TOUS les SGBDR et la réalisation de bonne conception de BD
C'est un minimum pour un concepteur/administrateur.
- Attention : **le modèle relationnel n'est pas SQL**.

Plan

Introduction

Domaines, relations, attributs

Algèbre Relationnelle

Structure des données

- Vocabulaire simple
 - Plusieurs termes pour le même concept car :
 - ▶ théorie/implémentation
 - ▶ informatique/mathématique
- Ex. : tuple = ligne = enregistrement
- Théorie relationnelle fondée sur la **théorie des ensembles**.
 - 3 briques principales : **domaine**, **attribut**, **relation**.

Domaine

- Def. Un **domaine** est
 - ▶ un **ensemble de valeurs**,
 - ▶ caractérisé par un **nom**.
- Ensemble nommé
- Idée : les données peuvent prendre leurs valeurs dans cet ensemble.
- Domaine \neq type de données (mais parfois confondu) : pas (forcément) d'opérations applicables aux valeurs.

Domaine (suite)

- Domaine défini **en extension** :
via l'ensemble de **toutes** les valeurs qui le composent.
Exemple : Devise = { *dollar, euro, livre, yen* }
- Domaine défini **en intention** :
via les **propriétés** respectées par ses valeurs (éléments).
Exemple : N_plus = les entiers *positifs*
- Extension vs. intension : voir
 - ▶ axiome d'extensionnalité dans la théorie des ensemble
 - ▶ égalité de Leibniz, axiomes de compréhension

Relation

- **Produit cartésien** d'une liste de domaines D_1, D_2, \dots, D_n

$$D_1 \times D_2 \times \dots \times D_n := \{ (v_1, v_2, \dots, v_n) \mid v_i \in D_i, \text{ for } i = 1, \dots, n \}.$$

- Exemple :

$$\text{Pays} = \{ \text{France}, \text{Italie}, \text{Japon} \},$$

$$\text{Devise} = \{ \text{euro}, \text{yen} \},$$

$$\text{Pays} \times \text{Devise} = \{ (\text{France}, \text{euro}), (\text{France}, \text{yen}), (\text{Italie}, \text{euro}), (\text{Italie}, \text{yen}), (\text{Japon}, \text{euro}), (\text{Japon}, \text{yen}) \}.$$

Relation

- Def. Une **relation** est
 - ▶ un **sous-ensemble** du produit cartésien (nommé)
d'une liste **finie** de domaines,
 - ▶ caractérisée par un **nom**.
- Une relation est donc un ensemble de tuplets avec un nom.
- Exemple :

$$\textit{DeviseCourante} = \{ (\textit{France}, \textit{euro}), (\textit{Italie}, \textit{euro}), (\textit{Japon}, \textit{yen}) \}.$$

- La définition d'une relation vise à **définir l'espace de définition** des tuplets (pas forcément en dimension 2).

Relation

- Famille d'ensembles indexée par A :

$$\{(E_a \mid a \in A)\}$$

- Pour chaque $a \in A$, le couple (a, E_a) est un domaine.
- **Produit cartésien nommé** d'un famille de domaines

$$\prod_{a \in A} E_a = \{f : A \rightarrow \bigcup_{a \in A} E_a \mid f(a) \in E_a, \text{ pour tout } a \in A\}.$$

Exemple :

Devise	Pays
euro	France

$$f(x) = \begin{cases} \textit{euro}, & x = \textit{Devise}, \\ \textit{France}, & x = \textit{Pays}, \end{cases}$$

Relation (fin)

- Pour représenter visuellement le contenu d'une relation on utilise les **tables**.
- Exemple :

DeviseCourante	Devise	Pays
	euro	France
	euro	Italie
	yen	Japon

- Chaque ligne correspond à un tuple (donc à une donnée),
- Chaque colonne donne un sous-ensemble du domaine.
- En SQL, le mot 'table' remplace 'relation'.

Attribut

- Def. Un **attribut** est
 - ▶ une **colonne d'une relation**
- Un attribut est caractérisé par un **nom unique** dans cette relation (= le nom du domaine)
- Attention : la colonne n'est pas le domaine (c'est plutôt un sous-ensemble).
- Un domaine est potentiellement infini, une colonne est toujours finie.
- Par les noms, un même ensemble peut être impliqué plusieurs fois dans la même relation.
- On peut accéder à une colonne via les noms. On a pas besoin d'utiliser les indices.

Tuple(t)s et représentation des relations

- Une ligne d'une relation est un n -uplet de valeurs. Chaque valeur est accédée via le nom de l'attribut qu'elle value.
On les appelle des **tuples** (**lignes** en SQL).
- Une relation de n domaines correspond à un **sous espace** d'un produit cartésien n dimensionnel.
- Elle peut donc être représentée **par un diagramme à n dimensions** dont chaque axe correspond à un domaine. Un tuple est un point dans cet espace.

Schéma de relation

- Une relation peut évoluer avec le temps.
- Sa définition/structure est figée en terme de domaines et d'attributs.
 - structure fixe vs contenu évolutif.
- Intention d'une relation (description par les attributs) vs extension (ensemble des tuples).
- **Schéma de relation** : description de l'intention (structure) d'une relation, i.e. de l'espace dans lequel évoluera l'extension, ainsi que les éventuelles contraintes.
- Exemple :

```
DeviseCourante(Devise,Pays)
```

Schéma, une définition formelle

- **Def.** Soit (n, R) une relation telle que $R \subseteq \prod_{a \in A} E_a$. Le schéma de (n, R) et le couple

$$(n, \{E_a \mid a \in A\}).$$

- Exemple :

$$\begin{aligned}A &= \{ \textit{Devise}, \textit{Pays} \} \\ E_{\textit{Devise}} &= \{ \textit{euro}, \textit{yen}, \textit{livre} \} \\ E_{\textit{Pays}} &= \{ \textit{France}, \textit{Italie}, \textit{Japon} \} \\ n &= \textit{DeviseCourante}\end{aligned}$$

- Dans le jargon informatique, on écrit cela par :

DeviseCourante(Devise,Pays)

Plan

Introduction

Domaines, relations, attributs

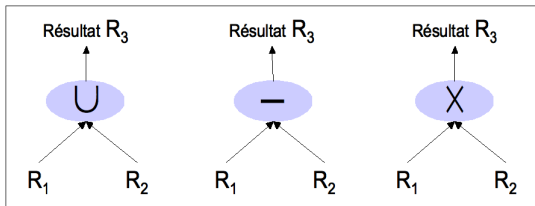
Algèbre Relationnelle

Opérations de base en Algèbre Relationnelle

- L'algèbre relationnelle est un ensemble d'opérateurs qui agissent sur des relations pour créer d'autres relations.
- Tout est relation : même si la relation résultante d'une opération n'a qu'une colonne et qu'une ligne, c'est encore un relation.
- Maîtrise de l'algèbre relationnel : essentiel pour la compréhension du langage SQL et des SGBDRs.
- 6 opérations de bases associées à une représentation

Opérations ensemblistes

- 3 opérations directement adaptées de la théorie des ensembles : **union**, **différence**, **produit cartésien**.
- Ne prennent pas en compte les contraintes de clés.
- Le résultat est une relation
(pas de tuplets/lignes doublées, l'ordre ne compte pas, ...)



Union

- **Def.** L'union des relations R_1 et R_2 de mêmes schémas, est la relation R_3 , toujours de même schéma, telle que l'ensemble des tuples de R_3 est l'union (sans doublons) de l'ensemble des tuples de R_1 et de l'ensemble des tuples de R_2 .
- Opération commutative !!!
- Notations :
 - ▶ $R_1 \cup R_2$,
 - ▶ `UNION(R_1 , R_2)`
- Note : les relations opérandes d'opérations d'algèbre relationnel ne sont pas forcément stockés en tant que telle dans la base.

Union : exemple

Acteur1	nom	prénom	datenaissance
	Braschi	Nicoletta	10/08/1960
	Depardieu	Gérard	27/12/1948
	Benigni	Roberto	27/10/1952
	Casta	Laetitia	11/05/1978

Acteur2	nom	prénom	datenaissance
	Blanc	Michel	16/04/1952
	Waits	Tom	NULL
	Clavier	Roberto	06/05/1952
	Casta	Laetitia	11/05/1978
	Depardieu	Gérard	27/12/1948

Acteur1 ∪ Acteur2	nom	prénom	datenaissance
	Braschi	Nicoletta	10/08/1960
	Depardieu	Gérard	27/12/1948
	Benigni	Roberto	27/10/1952
	Casta	Laetitia	11/05/1978
	Blanc	Michel	16/04/1952
	Waits	Tom	NULL
	Clavier	Roberto	06/05/1952

Différence

- **Def.** La **différence** entre les relations R_1 et R_2 de mêmes schémas, est la relation R_3 , de même schéma, telle que l'ensemble des tuples de R_3 est l'ensemble des tuples de R_1 auquel on a enlevé l'ensemble des tuples de R_2 .
- La différence n'est pas commutative !!!
- Notations :
 - ▶ $R_1 \setminus R_2$
 - ▶ $R_1 - R_2$
 - ▶ `EXCEPT(R_1 , R_2)`

Différence : exemple

Acteur1	nom	prénom	datenaissance
	Braschi	Nicoletta	10/08/1960
	Depardieu	Gérard	27/12/1948
	Benigni	Roberto	27/10/1952
	Casta	Laetitia	11/05/1978

Acteur2	nom	prénom	datenaissance
	Blanc	Michel	16/04/1952
	Waits	Tom	NULL
	Clavier	Roberto	06/05/1952
	Casta	Laetitia	11/05/1978
	Depardieu	Gérard	27/12/1948

Acteur1 - Acteur2	nom	prénom	datenaissance
	Braschi	Nicoletta	10/08/1960
	Benigni	Roberto	27/10/1952

Produit cartésien de relations

- **Def.** Le *produit cartésien* entre les relations R_1 et R_2 de schémas quelconques, est la relation R_3 qui a pour schéma la concaténation de ceux de R_1 et R_2 et, pour extension, l'ensemble de toutes les combinaisons possibles entre les tuples de R_1 et ceux de R_2 .
- Le produit cartésien de deux relations est une opération commutative (si on ne considère pas d'ordre sur les colonnes).
- Opération très courante : à la base des requêtes de confrontations.
- Notations :
 - ▶ $R_1 \times R_2$
 - ▶ `TIMES(R_1 , R_2)`
 - ▶ `PRODUCT(R_1 , R_2)`

Produit cartésien : exemple

Pays	nom	capitale	monnaie
	Italie	Roma	3
	France	Paris	3
	Gabon	Libreville	6
	Bénin	Porto-Novo	6

X

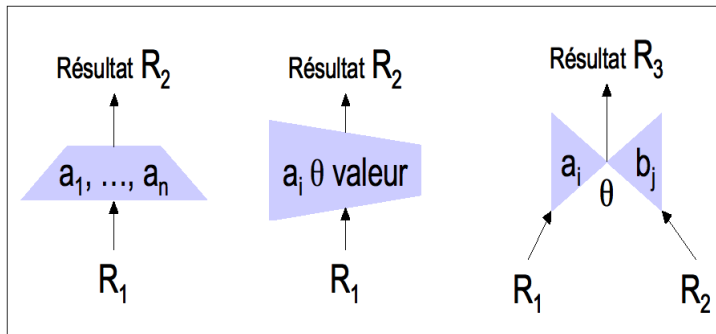
Monnaie	num	nom
	1	Dollar US
	3	Euro
	6	Franc CFA

Pays P	P.nom	capitale	monnaie	num	M.nom
X Monnaie M	Italie	Roma	3	1	Dollar US
	France	Paris	3	1	Dollar US
	Gabon	Libreville	6	1	Dollar US
	Bénin	Porto-Novo	6	1	Dollar US
	Italie	Roma	3	3	Euro
	France	Paris	3	3	Euro
	Gabon	Libreville	6	3	Euro
	Bénin	Porto-Novo	6	3	Euro
	Italie	Roma	3	6	Franc CFA
	France	Paris	3	6	Franc CFA
	Gabon	Libreville	6	6	Franc CFA
	Bénin	Porto-Novo	6	6	Franc CFA

Attention : Préfixage pour 2 colonnes partageant le même nom.

Opérations propre de l'algèbre relationnelle

- 3 opérations spécifiques, essentielles à la mise en œuvre du modèle dans le cadre opératoire.
- Projection, Restriction, Jointure



Projection

- Permet de ne retenir que quelques attributs d'une relation.
- Def. Soit
 - ▶ R une relation de schéma $R(a_1, \dots, a_n)$,
 - ▶ $B = \{a_{i_1}, a_{i_2}, \dots, a_{i_m}\} \subseteq \{a_1, \dots, a_n\}$.

La **projection de R sur les attributs B** , est la relation R' dont le schéma est $R'(a_{i_1}, a_{i_2}, \dots, a_{i_m})$ (avec même contraintes de domaine que R) et qui a la même extension que celle de R , mais dont les tuples ont moins attributs.

- Notations :
 - ▶ $\Pi_{(a_{i_1}, \dots, a_{i_m})}(R)$
 - ▶ $R[a_{i_1}, \dots, a_{i_m}]$,
 - ▶ $\text{PROJECT}(R, a_{i_1}, \dots, a_{i_m})$
- Pour $R \subseteq \prod_{a \in A} E_a$ et $B \subseteq A$, on a

$$\Pi_B(R) := \{f \upharpoonright_B \mid f \in R\}.$$

Projection : exemple

Acteur	nom	prénom	datenaissance
	Braschi	Nicoletta	10/08/1960
	Depardieu	Gérard	27/12/1948
	Benigni	Roberto	27/10/1952
	Casta	Laetitia	11/05/1978
	Blanc	Michel	16/04/1952
	Waits	Tom	NULL
	Clavier	Roberto	06/05/1952

(Acteur)[nom, prénom]	nom	prénom
	Braschi	Nicoletta
	Depardieu	Gérard
	Benigni	Roberto
	Casta	Laetitia
	Blanc	Michel
	Waits	Tom
	Clavier	Roberto

Restriction

- Réduit le nombre de tuples mais même nombre d'attributs.
- **Def.** Soit P un prédicat (critère logique) portant sur les attributs des tuples d'une relation R .
La **restriction de la relation R à P** est la relation R' dont l'extension sont les tuples de R satisfaisant le prédicat P .
- Critère de restriction :
 - ▶ `<attribut> <comparaison> <valeur>`
où `<comparaison>` dans $\{=, <, >, \leq, \geq, \neq\}$,
+ combinaison logique
 - ▶ Dialecte SQL :
 - ▶ (LIKE, IN, NOT NULL, etc.)
 - ▶ application de fonctions sur opérandes (TO_DATE(), ...)
- Notations :
 - ▶ $\sigma_{\text{critère}}(R)$
 - ▶ $R[\text{critère}]$
 - ▶ RESTRICT(R , critère)

Restriction : exemple

Acteur	nom	prénom	datenaissance
	Braschi	Nicoletta	10/08/1960
	Depardieu	Gérard	27/12/1948
	Benigni	Roberto	27/10/1952
	Casta	Laetitia	11/05/1978
	Blanc	Michel	16/04/1952
	Waits	Tom	NULL
	Clavier	Roberto	06/05/1952

(Acteur) [datenaissance < 1970]	nom	prénom	datenaissance
	Braschi	Nicoletta	10/08/1960
	Depardieu	Gérard	27/12/1948
	Benigni	Roberto	27/10/1952
	Blanc	Michel	16/04/1952
	Clavier	Roberto	06/05/1952

Jointure


- **Def.** La **jointure** des deux relations R_1 et R_2 (de schémas quelconques) **selon un critère logique** donné portant sur au moins un attribut de chaque relation, est la relation R_3 contenant l'ensemble de
 - ▶ tous les tuples obtenus en concaténant chaque tuple de R_1 et chaque tuple de R_2
 - ▶ qui vérifient, ensemble, le critère logique.
- Cas simple :
 - ▶ a_1 et a_2 des attributs respectivement de R_1 et R_2 ,
 - ▶ critère de jointure de la forme : $a_1 \theta a_2$, où θ est un opérateur de comparaison (cf. restriction)
 - ▶ on parle de **θ -jointure**.
- Pas vraiment une opération de base : peut être définie à partir du produit cartésien et d'une restriction.

Jointure : exemples

Pays	nom	capitale	monnaie
	Italie	Roma	3
	France	Paris	3
	Gabon	Libreville	6
	Bénin	Porto-Novo	6


Monnaie	num	nom
	1	Dollar US
	3	Euro
	6	Franc CFA

(a)

Pays P  Monnaie M
P.monnaie = M.num

P.nom	capitale	monnaie	num	M.nom
Italie	Roma	3	3	Euro
France	Paris	3	3	Euro
Gabon	Libreville	6	6	Franc CFA
Bénin	Porto-Novo	6	6	Franc CFA

(b)

Pays P  Monnaie M	nom	capitale	monnaie	num

Jointure (fin)

- Notations :
 - ▶ $R \bowtie_{\theta} R'$
 - ▶ $\text{JOIN}(R, R', \theta)$
- Si R a n attributs et t tuplets, R' a n' attributs et t' tuplets, alors $\text{JOIN}(R, R', \theta)$ a $n+n'$ attributs et au max $t+t'$ tuplets.
- Relation fondamentale :

$$\text{JOIN}(R, R', \theta) = \text{RESTRICT}(\text{TIMES}(R, R'), \theta)$$

- Jointure naturelle :
 - ▶ jointure entre 2 relations avec critère d'égalité (equi-jointure) entre 2 attributs de même noms
 - ▶ fusion des colonnes de même nom(s).