

Bases de données

Cours 3 : Bases de données relationnelles

Odile PAPINI

POLYTECH
Université d'Aix-Marseille
odile.papini@univ-amu.fr

<http://odile.papini.perso.esil.univmed.fr/sources/BD.html>

Plan du cours

- 1 Introduction
- 2 Structure
- 3 Contraintes d'Intégrité
- 4 Du conceptuel au relationnel
- 5 Manipulation
 - Algèbre relationnelle
- 6 Normalisation

Bibliographie

Livres :

- G. Gardarin : Bases de données objet et relationnel. Eyrolles ed. 1999.
- C. J. Date : Introduction aux bases de données. (8ième édition). Vuibert ed. 2004.
- H. Garcia-Molina, J. D. Ullman, J. Widow : Database systems, the complete book. Prentice Hall ed. 2002.

Supports de cours :

- Support de cours : J. Le Maitre :
<http://lemaitre.univ-tln.fr/cours.htm>
- Support de cours : C. Sabatier, Université de la Méditerranée.

Introduction

le **modèle relationnel** présente 3 aspects :

- **structuel** : toutes les données perçues comme des tables
- **intégrité** : les tables vérifient des contraintes d'intégrité
- **manipulation** : opérations pour accéder aux données, manipuler les tables ...

Le modèle relationnel

origine : E. F. Codd, chercheur à IBM, 1967 – 1970, prix Turing 1986

ce modèle repose sur :

- **la théorie des ensembles**
- **la logique des prédicats**

depuis les années 1980 de **nombreux produits commerciaux** :

DB2 d'IBM, Oracle, Sybase, Access, SQL serveur, ...

succès du à :

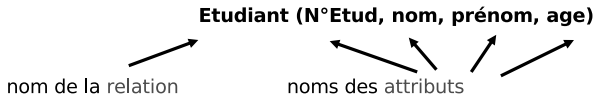
- **simplicité** : BD ensemble de tables
- **fondements théoriques** : outils théoriques solides

Le modèle relationnel : exemple

exemple de base de données relationnelle

<i>cave</i>				
<i>NUM</i>	<i>VIN</i>	<i>PRODUCTEUR</i>	<i>ANNÉE</i>	<i>NB</i>
2	<i>CabernetSauvignon</i>	<i>ChateauduMayne</i>	1983	5
3	<i>Chardonnay</i>	<i>Jaboulet – Vercherre</i>	1984	7
6	<i>PinotNoir</i>	<i>Jaboulet – Vercherre</i>	1984	10
12	<i>Brunello</i>	<i>EmilioCostanti</i>	1977	6
15	<i>Silvaner</i>	<i>Guntrum</i>	1985	8
23	<i>Riesling</i>	<i>BalthasarRess</i>	1976	2

Le modèle relationnel : exemple



schéma

Etudiant			
N°Etud	nom	prénom	age

population

136	Dupont	Jean	19
253	Aubry	Annie	20
101	Duval	André	21
147	Dupont	Marc	21

tuple ou occurrence

Base de données relationnelle

une Base de données relationnelle est constituée :

- un ensemble de domaines
- un ensemble de relations
- un ensemble de contraintes d'intégrité

Base de données relationnelle : Domaines

un **domaine** est un ensemble de valeurs atomiques

domaines prédéfinis :

- l'ensemble des chaînes de caractères
- l'ensemble des nombres entiers
- l'ensemble des booléens
- ...

domaines définis :

- **en extension** : énumération des valeurs
- **en intension** : définition d'une propriété caractéristique des valeurs

Base de données relationnelle : Domaines : exemples

domaine défini en **extension** :

$$\text{couleur} = \{ \text{" rouge", " rosé", " blanc" } \}$$

domaine défini en **intension** :

$$\text{mois} = \{ m , t. q. m \text{ est un entier et } 1 \leq m \leq 12 \}$$

domaines **compatibles** : valeurs sémantiquement comparables

$$\begin{aligned} \text{année} &= \{ a , t. q. a \text{ est un entier et } a \geq 1980 \} \\ \text{nombre} &= \{ n , t. q. n \text{ est un entier et } n \geq 1980 \} \end{aligned}$$

les années et les nombres de bouteilles ne sont pas compatibles

Base de données relationnelle : Domaines : Relations

sous-ensemble du produit cartésien de n domaines D_1, \dots, D_n

$$r \subseteq D_1 \times \dots \times D_n$$

une relation est définie par :

- **son nom**
- **son type**
- **son extension**

Base de données relationnelle : schéma d'une relation

définit les domaines sur lesquels elle est construite et donne un nom à ces domaines

$(A_1 : D_1, \dots, A_n : D_n)$ ou $r(A_1 : D_1, \dots, A_n : D_n)$

D_i : nom de domaine

A_i : nom d'**attribut** : rôle du domaine D_i dans la relation

schéma de la relation cave

(NUM : entier, VIN : chaîne de caractères, PRODUCTEUR :
chaîne de caractères, ANNÉE : entier , NB :entier)

ou

cave (NUM : entier, VIN : chaîne de caractères, PRODUCTEUR :
chaîne de caractères, ANNÉE : entier , NB :entier)

Base de données relationnelle : Extension d'une relation

extension d'une relation de schéma $(A_1 : D_1, \dots, A_n : D_n)$:

ensemble de n-uplets (v_1, \dots, v_n) tels que $(v_1 \in D_1, \dots, v_n \in D_n)$

exemple :

extension possible de la relation cave de schéma

(NUM : entier, VIN : chaîne de caractères, PRODUCTEUR :
chaîne de caractères, ANNÉE : entier , NB :entier)

{ (6 , Pinot Noir, Jaboulet-Vercherre, 1984, 10), (15, Silvaner,
Guntrum, 1985, 8) }

Base de données relationnelle : Visions d'une relation

Soit une relation r de schéma $(A_1 : D_1, \dots, A_n : D_n)$

vision tabulaire

- table de nom r
- n colonnes nommées A_1, \dots, A_n
- une ligne représente un n -uplet de l'extension de la relation

vision prédicative

- prédicat r
- $r(v_1, \dots, v_n)$ est
VRAI si (v_1, \dots, v_n) appartient à l'extension de r
FAUX sinon

Base de données relationnelle : Exemple

Soit une relation cave de schéma :
(NUM : entier, VIN : chaîne de caractères, PRODUCTEUR :
chaîne de caractères, ANNÉE : entier , NB :entier)

vision tabulaire

cave				
NUM	VIN	PRODUCTEUR	ANNÉE	NB
6	Pinot Noir	Jaboulet-Vercherre	1984	10
15	Silvaner	Guntrum	1985	8

vision prédicative

cave(6 , Pinot Noir, Jaboulet-Vercherre, 1984, 10) est VRAI

cave(15, Silvaner, Guntrum, 1985, 8) est VRAI

Base de données relationnelle : Clés

constituant d'une relation :

sous-ensemble éventuellement vide des attributs de la relation

un constituant X est clé candidate d'une relation r si :

- pour chaque n-uplet de r la valeur de X identifie de façon unique ce n-uplet
- aucun attribut de X ne peut être supprimé sans détruire la propriété précédente

clé primaire : clé choisie parmi les clés candidates

clé étrangère : Y clé étrangère de r_i

- il existe une relation r_j possédant une clé primaire X
- Y a pour domaine l'ensemble des valeurs de X

Y réfère la relation r_j

Base de données relationnelle : Exemple

cave				
NUM	VIN	PRODUCTEUR	ANNÉE	NB
6	Pinot Noir	Jaboulet-Vercherre	1984	10

buveur			
NUB	NOM	PRÉNOM	ADRESSE
367	Poivrot	Hercule	rue de la Loi Bruxelles

abus			
NUM	NUB	DATE	QUANTITÉ
6	367	23/11/2006	2

NUM : **clé primaire** de la relation cave
NUB : **clé primaire** de la relation buveur
{NUM, NUB, DATE} : **clé primaire** de la relation abus
NUB : **clé étrangère** de la relation abus dans la relation buveur
NUM : **clé étrangère** de la relation abus dans la relation cave

Base de données relationnelle : Contraintes d'intégrité

s'expriment par :

- appartenance des valeurs d'attributs à des domaines
- définition de clés
- normalisation des relations
- ensemble d'assertations
- conditions associées aux opérations de mise à jour

valeur nulle :

absence ou non-pertinence d'information : nul (ou NUL)

Base de données relationnelle : Contraintes d'intégrité

Exemple

cave					
NUM	VIN	PRODUCTEUR	ANNÉE	NB	DEGRÉ
6	Pinot Noir	Jaboulet-Vercherre	1984	10	NUL

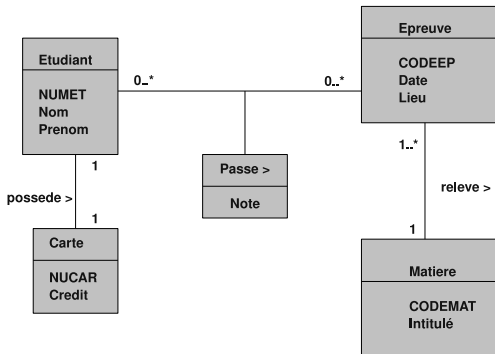
contrainte d'intégrité d'entité : toute relation possède une clé primaire et les valeurs d'attribut de cette clé doivent être non nulles

contrainte d'intégrité référentielle : chaque valeur d'une clé étrangère Y :

- soit existe comme valeur de la clé primaire d'un n-uplet de la relation que Y réfère
- soit nulle

Base de données relationnelle : Du conceptuel au relationnel

Comment traduire en relationnel ce schéma conceptuel ?



Base de données relationnelle : Du conceptuel au relationnel

Règles de traduction

types d'entité

chaque type d'entité est traduit en une relation

- attribut → attribut de la relation
- identifiant → clé primaire de la relation

Base de données relationnelle : Du conceptuel au relationnel

Règles de traduction

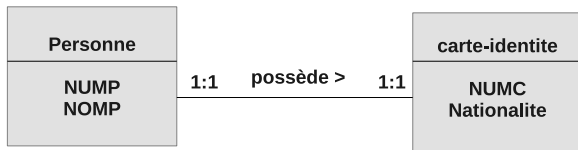
types d'association

type d'association cardinalité (1 : 1) - (1 : 1)

- pas de création de relation
- la clé primaire d'une relation est rajoutée comme clé étrangère dans l'autre relation

Base de données relationnelle : Du conceptuel au relationnel

Exemple type d'association cardinalité (1 : 1) - (1 : 1)



traduction relationnelle

personne(NUMP, NOMP)

carte-identite(NUMC, Nationalite, #NUMP)

Base de données relationnelle : Du conceptuel au relationnel

Règles de traduction

types d'association

type d'association cardinalité (0 : 1) - (1 : 1)

- pas de création de relation
- la clé primaire de la relation de cardinalité (1 :1) est rajoutée comme clé étrangère dans l'autre relation

Base de données relationnelle : Du conceptuel au relationnel

Exemple type d'association cardinalité (0 : 1) - (1 : 1)



traduction relationnelle

employe(NUME, NOME)

service(NUMS, NOMS, #NUME)

Base de données relationnelle : Du conceptuel au relationnel

Règles de traduction

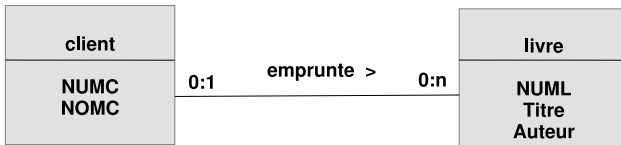
types d'association

type d'association cardinalité (0 : 1) - (* : n)

- pas de création de relation
- la clé primaire de la relation dont la cardinalité MAX est 1 est rajoutée comme clé étrangère dans l'autre relation

Base de données relationnelle : Du conceptuel au relationnel

Exemple type d'association cardinalité (0 : 1) - (* : n)



traduction relationnelle

client(NUMC, NOMC)

livre(NUML, Titre, Auteur, #NUMC)

Base de données relationnelle : Du conceptuel au relationnel

Règles de traduction

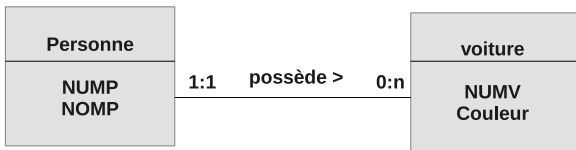
types d'association

type d'association cardinalité (1 : 1) - (* : n)

- pas de création de relation
- la clé primaire de la relation dont la cardinalité MAX est 1 est rajoutée comme clé étrangère dans l'autre relation

Base de données relationnelle : Du conceptuel au relationnel

Exemple type d'association cardinalité (1 : 1) - (* : n)



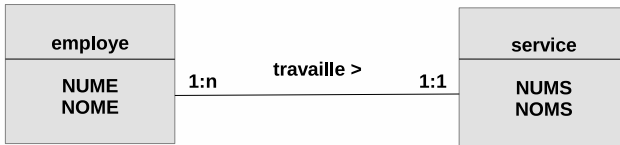
traduction relationnelle

personne(NUMP, NOMP)

voituree(NUMV, Couleur, #NUMP)

Base de données relationnelle : Du conceptuel au relationnel

Autre exemple type d'association cardinalité (1 : 1) - (* : n)



traduction relationnelle

employe(NUME, NOME, #NUMS)

service(NUMS, NOMS,)

Base de données relationnelle : Du conceptuel au relationnel

Règles de traduction

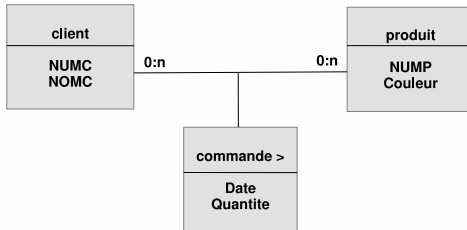
types d'association

type d'association cardinalité (* : n) - (* : n)

- création d'une nouvelle relation
- la clé primaire de cette relation est la concaténation des clés primaires des relations participantes
- les attributs du type d'association sont insérés dans cette nouvelle relation si nécessaire

Base de données relationnelle : Du conceptuel au relationnel

Exemple type d'association cardinalité (* : n) - (* : n)



traduction relationnelle

client(NUMC, NOMC)

produit(NUMP, Couleur)

Base de données relationnelle : Du conceptuel au relationnel

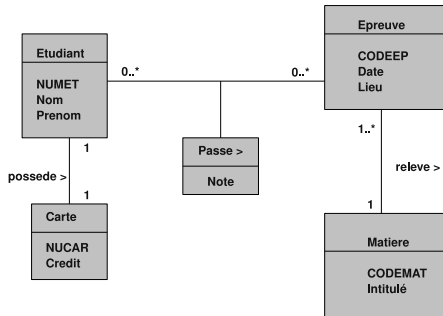
Contraintes d'intégrité

pour chaque relation

- contraintes de domaine (vérification d'appartenance des données au domaine)
- contraintes de clés primaires (clé primaire unique, non nulle (renseignée))
- contraintes de clés étrangères (vérification de références correctes)

Base de données relationnelle : Du conceptuel au relationnel

Exercice : traduire ce schéma conceptuel



Base de données relationnelle : Manipulation

Langages relationnels

- **langages algébriques :**
basés sur l'algèbre relationnelle, vision ensembliste
- **langages prédicatifs :**
basés sur le calcul relationnel, vision assertionnelle

Base de données relationnelle : Manipulation

Algèbre relationnelle

ensemble d'opérateurs qui manipulent des extensions de relations (ensemble de n -uplets)

Tout opérateur s'applique à une ou plusieurs relations et produit une relation

opérateurs

- **ensemblistes** :
union, intersection, différence, produit cartésien
- **spécifiques aux bases de données** :
sélection, jointure, projection, division, ...

Base de données relationnelle : Manipulation

Expression de l'algèbre relationnelle

construite à partir :

- **noms des relations de la BD** (relations de base)
- **noms des variables** aux quelles sont affectées les relations
- **comparateurs** : $<$, \leq , $=$, \geq , $>$
- **connecteurs logiques** : \wedge , \vee , \neg
- **opérateurs de l'algèbre relationnelle**
- **opérateur d'affectation** : $=$

Base de données relationnelle : Manipulation

Notations

- **expressions de l'algèbre relationnelle** : lettres minuscules p, q, r, \dots
- **noms d'attributs** : lettres majuscules A, B, C, \dots
- **valeurs d'attribut dans un n -uplet** : $r.A$
- **schéma de la relation valeur de l'expression r** : $schema(r)$
- **extension de la relation valeur de l'expression r** : $ext(r)$
- **concaténation de deux n -uplets** : $conc$

Base de données relationnelle : Manipulation

Opérateurs ensemblistes

p et q deux expressions algébriques telles que :
 $schema(p) = schema(q) = (A_1 : D_1, \dots, A_n : D_n)$

- **union :**

- $schema(p \cup q) = (A_1 : D_1, \dots, A_n : D_n)$
- $ext(p \cup q) = \{t \mid t \in ext(p) \vee t \in ext(q)\}$

- **intersection :**

- $schema(p \cap q) = (A_1 : D_1, \dots, A_n : D_n)$
- $ext(p \cap q) = \{t \mid t \in ext(p) \wedge t \in ext(q)\}$

- **différence :** –

- $schema(p - q) = (A_1 : D_1, \dots, A_n : D_n)$
- $ext(p - q) = \{t \mid t \in ext(p) \wedge t \notin ext(q)\}$

Base de données relationnelle : Manipulation

union : exemple

p		
A	B	C
a	b	c
d	a	f
c	b	d

q		
A	B	C
b	g	a
d	a	f

$p \cup q$		
A	B	C
a	b	c
d	a	f
c	b	d
b	g	a

Base de données relationnelle : Manipulation

intersection : exemple

p		
A	B	C
a	b	c
d	a	f
c	b	d

q		
A	B	C
b	g	a
d	a	f

$p \cap q$		
A	B	C
d	a	f

Base de données relationnelle : Manipulation

différence : exemple

p		
A	B	C
a	b	c
d	a	f
c	b	d

q		
A	B	C
b	g	a
d	a	f

p - q		
A	B	C
a	b	c
c	b	d

Base de données relationnelle : Manipulation

Opérateurs ensemblistes

r expression algébrique : $schema(r) = (A_1 : D_1, \dots, A_n : D_n)$

- **complément** : –

- $schema(-r) = (A_1 : D_1, \dots, A_n : D_n)$
- $ext(-r) = \{t \mid t \notin ext(r)\}$

exemple :

$D_A = \{a, b, c\}$ et $D_B = \{e, d\}$

r	
A	B
a	e
c	d
c	e

$-r$	
A	B
a	d
b	e
b	d

Base de données relationnelle : Manipulation

Opérateurs ensemblistes : produit cartésien

p et q deux expressions algébriques telles que :

- $schema(p) = (A_1 : D_1, \dots, A_m : D_m)$
- $schema(q) = (B_1 : D'_1, \dots, B_n : D'_n)$
- $\{A_1, \dots, A_m\} \cap \{B_1, \dots, B_n\} = \emptyset$
- **produit cartésien :**
 - $schema(p \times q) = (A_1 : D_1, \dots, A_m : D_m, B_1 : D'_1, \dots, B_n : D'_n)$
 - $ext(p \times q) = \{u \text{ conc } v \mid u \in ext(p) \wedge v \in ext(q)\}$

Base de données relationnelle : Manipulation

Produit cartésien : exemple

p		
A	B	C
a	b	c
d	a	f
c	b	d

q		
D	E	F
b	g	a
d	a	f

p × q					
A	B	C	D	E	F
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f

Base de données relationnelle : Manipulation

Opérateurs spécifiques : sélection

σ extrait l'ensemble des n -uplets qui vérifient une condition *cond*

- *cond* : expression booléenne
 - opérandes : valeurs du domaine, noms d'attributs
 - opérateurs : comparateurs, connecteurs logiques

r une expression algébrique telle que :

$$\text{schema}(r) = (A_1 : D_1, \dots, A_n : D_n)$$

- **sélection** :

$$\text{schema}(\sigma_{\text{cond}}(r)) = (A_1 : D_1, \dots, A_n : D_n)$$

$$\text{ext}(\sigma_{\text{cond}}(r)) = \{t \mid t \in \text{ext}(r) \wedge \text{val}(\text{cond})\}$$

Base de données relationnelle : Manipulation

sélection : exemple , $cond : r.B = b$

r		
A	B	C
a	b	c
d	a	f
c	b	d

$\sigma_{cond}(r)$		
A	B	C
a	b	c
c	b	d

Base de données relationnelle : Manipulation

Opérateurs spécifiques : projection

π extrait l'ensemble des valeurs d'un constituant de relation

r une expression algébrique telle que :

$$\text{schema}(r) = (\dots, A_1 : D_1, \dots, A_n : D_n, \dots)$$

- **projection :**

$$\text{schema}(\pi_{A_1, \dots, A_n}(r)) = (A_1 : D_1, \dots, A_n : D_n)$$

$$\text{ext}(\pi_{A_1, \dots, A_n}(r)) = \{(t.A_1, \dots, t.A_n) \mid t \in \text{ext}(r)\}$$

Base de données relationnelle : Manipulation

projection : exemple

r		
A	B	C
a	b	c
d	a	f
c	b	d

$\pi_{A,C}(r)$	
A	C
a	c
d	f
c	d

Base de données relationnelle : Manipulation

Opérateurs spécifiques : jointure (\bowtie ou join)

join fusionne deux relations en produisant toutes les concaténations de leurs n -uplets respectifs qui vérifient une condition *cond*

jointure : équivalente au produit cartésien suivi d'une sélection

$$p \bowtie_{cond} q \equiv \sigma_{cond}(p \times q)$$

Base de données relationnelle : Manipulation

jointure : exemple , $cond : p.B < q.D$

p		
A	B	C
1	2	3
4	5	6
7	8	9

q	
D	E
3	1
6	2

$p \bowtie_{cond} q$				
A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

Base de données relationnelle : Manipulation

Opérateurs spécifiques : jointure externe

conservation dans le résultat d'une jointure des n -uplets de chacune des deux relations qui ne peuvent pas être joints

$$schema(p) = (A_1 : D_1, \dots, A_m : D_m)$$

$$schema(q) = (B_1 : D'_1, \dots, B_n : D'_n), \{A_1, \dots, A_m\} \cap \{B_1, \dots, B_n\} = \emptyset$$

- **jointure externe :**

$$schema(p \text{ ejoin}_{cond} q) = (A_1 : D_1, \dots, A_m : D_m, B_1 : D'_1, \dots, B_n : D'_n)$$

$$ext(p \text{ ejoin}_{cond} q) = union(p_{ext}, j, q_{ext})$$

$$j = p \bowtie_{cond} q$$

$$p_{ext} = (p - \pi_{A_1, \dots, A_m}(j)) \times \{(nul_1, \dots, nul_n)\}$$

$$q_{ext} = \{(nul_1, \dots, nul_m)\} \times (q - \pi_{B_1, \dots, B_n}(j))$$

Base de données relationnelle : Manipulation

jointure externe : exemple , $cond : p.B < q.D$)

p		
A	B	C
1	2	3
4	5	6
7	8	9

q	
D	E
3	1
6	2
1	8

p $ejoin_{cond}$ q				
A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2
7	8	9	nul	nul
nul	nul	nul	1	8

Base de données relationnelle : Manipulation

Opérateurs spécifiques : jointure naturelle (\bowtie ou natural join)

join fusionne deux relations en produisant toutes les concaténations de leurs n -uplets respectifs qui ont les mêmes valeurs d'attribut sur les attributs communs.

$$\text{schema}(p) = (A_1 : D_1, \dots, A_m : D_m) \quad \text{schema}(q) = (B_1 : D'_1, \dots, B_n : D'_n)$$

$$\{A_1, \dots, A_m\} \cap \{B_1, \dots, B_n\} = \{C_1, \dots, C_k\}$$

- **jointure naturelle :**

$$\text{schema}(p \bowtie q) = \text{schema}(p) \cup \text{schema}(q)$$

$$\text{ext}(p \bowtie q) = \sigma_{\text{cond}}(p \times q) \text{ avec } \text{cond} : p.C_i = q.C_i \text{ pour } 1 \leq i \leq k$$

Base de données relationnelle : Manipulation

jointure naturelle : exemple , $cond : p.A = q.A$

p		
A	B	C
1	2	3
4	5	6
7	8	9

q	
A	E
1	4
2	5
7	6

$p \bowtie q$			
A	B	C	E
1	2	3	4
7	8	9	6

Base de données relationnelle : Manipulation

Opérateurs spécifiques : semi-jointure

extrait l'ensemble des n -uplets d'une relation dont l'union avec l'un au moins des n -uplets d'une seconde relation vérifie un condition

exprime la quantification existentielle

- **semi-jointure** : *sjoin*

$$p \text{ sjoin}_{cond} q \equiv \pi_{attributs(p)}(p \bowtie_{cond} q)$$

$$ext(\text{sjoin}_{cond}(p, q, cond)) = \{u \mid u \in ext(p) \wedge \exists v \in ext(q)(val(cond))\}$$

Base de données relationnelle : Manipulation

semi-jointure : exemple , $cond : p.B < q.D$

p		
A	B	C
1	2	3
4	5	6
7	8	9

q	
D	E
3	1
6	2

$p \text{ sjoin}_{cond} q$		
A	B	C
1	2	3
4	5	6

Opérateurs spécifiques : division

p et q deux expressions algébriques telles que :

$$\text{schema}(p) = (A_1 : D_1, \dots, A_m : D_m, B_1 : D'_1, \dots, B_n : D'_n)$$

$$\text{schema}(q) = (B_1 : D'_1, \dots, B_n : D'_n)$$

$$(A_1 : D_1, \dots, A_m : D_m) \cap (B_1 : D'_1, \dots, B_n : D'_n) = \emptyset$$

exprime la quantification universelle

- **division** : %

$$\text{schema}(p \% q) = (A_1 : D_1, \dots, A_m : D_m)$$

$$\text{ext}(p \% q) = \{t \mid t \in \text{ext}(\pi_{A_1, \dots, A_m}(p))$$

$$\wedge (\forall v \in \text{ext}(q)(\exists u \in \text{ext}(p), u = t \text{ conc } v))\}$$

Base de données relationnelle : Manipulation

division : exemple

p		
A	B	C
1	2	3
1	6	2
4	5	6
5	2	3
5	6	2
7	8	9

q	
B	C
2	3
6	2

p % q
A
1
5

Base de données relationnelle : Manipulation

Opérateurs spécifiques : renommage

renommer certains attributs
 r une relation telle que :

$$\text{schema}(r) = (A_1 : D_1, \dots, A_n : D_n)$$

- **renommage** : ρ

$$\text{schema}(\rho_{B_1, \dots, B_n}(r)) = (B_1 : D_1, \dots, B_n : D_n)$$

$$\text{ext}(\rho_{B_1, \dots, B_n}(r)) = \text{ext}(r)$$

Base de données relationnelle : Manipulation

renommage : exemple

r		
A	B	C
1	2	3
4	5	6
7	8	9

$\rho_{X,Y,Z}(r)$		
X	Y	Z
1	2	3
4	5	6
7	8	9

Base de données relationnelle : Manipulation

Propriétés des opérateurs ensemblistes

Soit p , q et r trois expressions algébriques

- **idempotence :**

- $p \cup p = p$

- $p \cap p = p$

- **commutativité :**

- $p \cup q = q \cup p$

- $p \cap q = q \cap p$

- **associativité :**

- $(p \cup q) \cup r = p \cup (q \cup r)$

- $(p \cap q) \cap r = p \cap (q \cap r)$

- **distributivité :**

- $p \cup (q \cap r) = (p \cup q) \cap (p \cup r)$

- $p \cap (q \cup r) = (p \cap q) \cup (p \cap r)$

Base de données relationnelle : Manipulation

Propriétés des autres opérateurs

Soit p , q et r trois expressions algébriques

- **commutativité :**

- $p \bowtie_{cond} q = q \bowtie_{cond} p$
- $pxq = qxq$

- **associativité :**

si p , q et r n'ont aucun attribut en commun

- $(p \bowtie_{cond_1} q) \bowtie_{cond_2} r = p \bowtie_{cond_1} (q \bowtie_{cond_2} r)$
- $(pxq)xr = qx(pxr)$

Base de données relationnelle : Manipulation

Propriétés des autres opérateurs

Soit p , q et r trois expressions algébriques

- **cascade de sélections :**

- $\sigma_{cond_1}(\sigma_{cond_2}(r)) = \sigma_{cond_2}(\sigma_{cond_1}(r)) = \sigma_{cond_1 \wedge cond_2}(r)$

- **cascade de projections :**

soit 2 ensembles d'attributs tels que

$$\{B_1, \dots, B_m\} \subseteq \{A_1, \dots, A_n\}$$

- $(\pi_{B_1, \dots, B_m}(\pi_{A_1, \dots, A_n}(r))) = (\pi_{B_1, \dots, B_m}(r))$

Base de données relationnelle : Manipulation

présence de valeurs nulles dans les n -uplets : introduction
d'une valeur de vérité INCONNUE

résultat de la comparaison entre la valeur nul et tout autre valeur a
pour valeur INCONNU

modification des tables de vérité des connecteurs logiques

table de vérité pour le connecteur \wedge

\wedge	VRAI	FAUX	INCONNUE
VRAI	VRAI	FAUX	INCONNUE
FAUX	FAUX	FAUX	FAUX
INCONNUE	INCONNUE	FAUX	INCONNUE

Base de données relationnelle : Manipulation

table de vérité pour le connecteur \vee

\vee	VRAI	FAUX	INCONNUE
VRAI	VRAI	VRAI	VRAI
FAUX	VRAI	FAUX	INCONNUE
INCONNUE	VRAI	INCONNUE	INCONNUE

table de vérité pour le connecteur \neg

	\neg
VRAI	FAUX
FAUX	VRAI
INCONNUE	INCONNUE

Base de données relationnelle : Manipulation

Algèbre relationnelle

- **complète** : permet d'exprimer toutes les requêtes formulables dans un langage du premier ordre
- les **requêtes simples** : opérateurs de l'algèbre relationnelle
- les **requêtes complexes** : composition d'opérateurs de l'algèbre relationnelle
- optimisation des requêtes

Base de données relationnelle : Manipulation

Quelques remarques sur l'utilisation des opérateurs

- les **sélections diminuent** le nombre de n -uplets et donc la taille des tables
- les **projections diminuent un peu** la taille des tables
- les **produits** et les **jointures augmentent considérablement** la taille des tables

Base de données relationnelle : Manipulation : exemple (1)

exemple : BD vols-réservations (cours C. Sabatier (Univ. Méditerranée))

Voici 3 schémas de relations :

- avions(No_AV, NOM_AV, CAP, LOC)
- pilotes(No_PIL, NOM_PIL, VILLE)
- vols(No_VOL, No_AV, No_PIL, V_d, V_a, H_d, H_a)

Base de données relationnelle : Manipulation : exemple (2)

et voici les 3 tables correspondantes :

avions				pilotes		
No_AV	NOM_AV	CAP	LOC	No_PIL	NOM_PIL	VILLE
100	airbus	300	nice	1	laurent	nice
101	airbus	300	paris	2	sophie	paris
102	carav	200	toulouse	3	claud	grenoble

vols						
No_VOL	No_AV	No_PIL	V_d	V_a	H_d	H_a
it100	100	1	nice	paris	7	8
it101	100	2	paris	toulouse	11	12
it102	101	1	paris	nice	12	13
it103	102	3	grenoble	toulouse	9	11
it104	101	3	toulouse	grenoble	17	18

Base de données relationnelle : Manipulation : exemple

exemple : BD vols-réservations

comment formuler la question suivante en utilisant les opérateurs de l'algèbre relationnelle ?

Quels sont les noms des pilotes qui conduisent tous les types d'avions ?

Normalisation

Définition d'un schéma relationnel afin d'éviter

- la redondance des données
- les incohérences lors des mises à jour
- les anomalies lors des suppressions et/ou des insertions

La **normalisation** repose sur l'analyse des dépendance entre attributs

- décomposition des relations sujettes à redondance

Normalisation

exemple (cours F. Trichet (Univ. Nantes))

- Une entreprise de vente de bateaux qui souhaite constituer un système d'information relatif à son activité

achats ID_client(NOM_CLIENT, PRENOM_CLIENT,
ADRESSE_CLIENT, IDENTIFICATION_BATEAU,
MODELE_BATEAU, LONGUEUR_BATEAU DATE_ACHAT,
MONTANT_ACHAT)

- problèmes potentiels :
 - redondance des données
 - incohérence suite à une mise à jour
 - anomalies lors d'insertion / suppression

Normalisation

exemple

- Une entreprise de vente de bateaux qui souhaite constituer un système d'information relatif à son activité

achats ID_client(NOM_CLIENT, PRÉNOM_CLIENT, ADRESSE_CLIENT, IDENTIFICATION_BATEAU, MODELE_BATEAU, LONGUEUR_BATEAU DATE_ACHAT, MONTANT_ACHAT)

- problèmes potentiels :
 - redondance des données : achat de plusieurs bateaux
 - incohérence suite à une mise à jour : changement d'adresse
 - anomalies lors d'insertion / suppression : client potentiel ne peut pas être enregistré dans la BD s'il n'a pas encore acheté un bateau, lorsqu'un client vend son bateau il est supprimé du système d'information

Normalisation

autre exemple : BD alpinisme (cours J. Le Maitre (USTV))

r			
<u>NOM_SOMMET</u>	<u>FACE</u>	<u>ALTITUDE</u>	<u>ANNÉE</u>
Everest	S	8848	1953
Manaslu	S	8125	1972
Hidden-Peak	NO	8068	1975
Everest	SO	8848	1975

problèmes potentiels :

- insertion : insertion d'un sommet seulement si une première a eu lieu sur celui-ci
- modification : si l'altitude de l'Everest est modifiée il faut la modifier dans toutes les premières de ce sommet
- suppression : si l'unique première sur un sommet est supprimée, l'information sur son altitude est perdue

Normalisation : dépendance fonctionnelle

dépendance fonctionnelle

r : relation, X et Y : deux sous-ensembles d'attributs
 Y dépend fonctionnellement de X ou X détermine
fonctionnellement Y : $X \rightarrow Y$

si pour toute extension de r , $ext(r)$, et tout tuples t_1 et t_2 de
 $ext(r)$ on a

$$\text{si } \pi_X(t_1) = \pi_X(t_2) \text{ alors } \pi_Y(t_1) = \pi_Y(t_2)$$

Une dépendance fonctionnelle $X \rightarrow Y$ est dite **élémentaire** ou
triviale si $Y = \emptyset$ ou $Y \subseteq X$

Normalisation : dépendance fonctionnelle

Exemple de dépendances fonctionnelles

voiture				
NUMERO	MARQUE	TYPE	PUISSANCE	COULEUR
872RH75	Peugeot	P206A	7	BLEUE
975AB80	Peugeot	P206A	7	ROUGE

Dépendances fonctionnelles

NUMERO → **COULEUR**

TYPE → **MARQUE**

TYPE → **PUISSANCE**

(TYPE, MARQUE) → **PUISSANCE**

Normalisation : dépendance fonctionnelle

Exemple de dépendances fonctionnelles

R			
<u>NOM_SOMMET</u>	<u>FACE</u>	ALTITUDE	ANNÉE
Everest	S	8848	1953
Manaslu	S	8125	1972
Hidden-Peak	NO	8068	1975
Everest	SO	8848	1975

Dépendances fonctionnelles :

NOM_SOMMET → **ALTITUDE**
(NOM_SOMMET, FACE) → **ANNÉE**

Normalisation : dépendance fonctionnelle : propriétés

propriétés de base (Armstrong)

- **réflexivité** : si $Y \subseteq X$ alors $X \rightarrow Y$
- **augmentation** : si $X \rightarrow Y$ alors $X \cup Z \rightarrow Y \cup Z$
- **transitivité** : si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$

propriétés déduites

- **union** : si $X \rightarrow Y$ et $X \rightarrow Z$ alors $X \rightarrow Y \cup Z$
- **pseudo-transitivité** : si $X \rightarrow Y$ et $W \cup Y \rightarrow Z$ alors $W \cup X \rightarrow Z$
- **décomposition** : si $X \rightarrow Y \cup Z$ alors $X \rightarrow Y$ et $X \rightarrow Z$

Normalisation : dépendance fonctionnelle : propriétés

- **fermeture** d'un ensemble de dépendances fonctionnelles F :
 F^+ ensemble de toutes les dépendances fonctionnelles déduites obtenues par l'application répétées des propriétés de base.
- F et G deux ensembles de dépendances fonctionnelles
 F et G sont **équivalents** ssi $F^+ = G^+$
- **fermeture transitive** d'un ensemble de dépendances fonctionnelles F :
ensemble de toutes les dépendances fonctionnelles déduites obtenues par transitivité

Normalisation : dépendance fonctionnelle : propriétés

- **un ensemble de dépendances fonctionnelles F est irréductible si :**
 - le membre droit de chaque dépendance fonctionnelle contient un seul attribut
 - chaque dépendance fonctionnelle de F est élémentaire
 - aucune dépendance fonctionnelle de F ne peut être supprimée sans changer la fermeture de F
- tout ensemble de dépendances fonctionnelles irréductible et équivalent à F est appelé **couverture minimale** de F

Normalisation

- analyse des dépendances à l'origine des redondances
- méthode de décomposition des relations permettant de les reconstruire par jointure
- propriétés de conservation des dépendances fonctionnelles et de l'information (jointure sans perte)

Normalisation : 1ère forme normale

Décomposition en 1ère forme normale :

tous les attributs sont atomiques

r	
<u>NOM_SOMMET</u>	PAYS
Everest	Chine, Népal
K2	Chine, Pakistan



r	
<u>NOM_SOMMET</u>	PAYS
Everest	Chine
Everest	Népal
K2	Chine
K2	Pakistan

Normalisation : 2ième forme normale

Décomposition en 2ième forme normale :

- décomposition en 1ière forme normale
- tout attribut non clé dépend totalement de la clé primaire
- sans perte d'information (reconstitution par jointure)
- préservation des dépendances fonctionnelles

Normalisation : 2ième forme normale

r			
<u>NOM_SOMMET</u>	<u>FACE</u>	ALTITUDE	ANNÉE
Everest	S	8848	1953
Manaslu	S	8125	1972
Hidden-Peak	NO	8068	1975
Everest	SO	8848	1975

NOM_SOMMET → ALTITUDE

(NOM_SOMMET, FACE) → ANNÉE

r_1			r_2	
<u>NOM_SOMMET</u>	<u>FACE</u>	ANNÉE	<u>NOM_SOMMET</u>	ALTITUDE
Everest	S	1953	Everest	8848
Manaslu	S	1972	Manaslu	8125
Hidden-Peak	NO	1975	Hidden-Peak	8068
Everest	SO	1975	Everest	8848

Normalisation : 3ième forme normale

Décomposition en 3ième forme normale :

- décomposition en 2ième forme normale
- tout attribut non clé ne dépend pas d'un autre attribut non clé
- sans perte d'information (reconstitution par jointure)
- préservation des dépendances fonctionnelles

Normalisation : 3ième forme normale

r		
<u>GRIMPEUR</u>	VILLE	PAYS
Campanoni	Cortina	Italie
Lacedelli	Cortina	Italie
Buhl	Innsbruck	Autriche

GRIMPEUR → VILLE

VILLE → PAYS

r_1	
<u>GRIMPEUR</u>	VILLE
Campanoni	Cortina
Lacedelli	Cortina
Buhl	Innsbruck

r_2	
<u>VILLE</u>	PAYS
Cortina	Italie
Innsbruck	autriche

Normalisation : 4ième forme normale

Dépendance multi-valuée :

Dépendance multi-valuée entre les constituants X et Y d'une relation $r(X,Y,Z)$ si pour toute extension de r ($X \twoheadrightarrow Y$)

- à chaque valeur de X il correspond toujours le même ensemble de valeurs de Y
- cet ensemble de valeurs ne dépend pas de Z

r	
<u>NOM_SOMMET</u>	<u>PAYS</u>
Everest	Népal
Everest	Chine
Manaslu	Népal
Hidden-Peak	Chine
Hidden-Peak	Pakistan

$NOM_SOMMET \twoheadrightarrow PAYS$

Normalisation : 4ième forme normale

Décomposition en 4ième forme normale :

- décomposition en 3ième forme normale
- pour toute dépendance multivaluée $X \twoheadrightarrow Y$ t. q. $X \cup Y$ n'inclut pas tous les attributs de r , X est une super-clé de r

r		
<u>NOM_SOMMET</u>	<u>ALTITUDE</u>	<u>PAYS</u>
Everest	8848	Népal
Everest	8848	Chine

$NOM_SOMMET \twoheadrightarrow PAYS$

r n'est pas en 4ième forme normale

Normalisation : 4ième forme normale

<i>r</i>		
<u>NOM_SOMMET</u>	<u>ALTITUDE</u>	<u>PAYS</u>
Everest	8848	Népal
Everest	8848	Chine

<i>r₁</i>	
<u>NOM_SOMMET</u>	<u>PAYS</u>
Everest	Népal
Everest	Chine

<i>r₂</i>	
<u>NOM_SOMMET</u>	<u>ALTITUDE</u>
Everest	8848

r_1 et r_2 sont en 4ième forme normale