Introduction aux Bases de Données

Joël Quinqueton

Licence Mass, Universités de Montpellier II et III

Algèbre relationnelle

- Concepts et outils : introduction à SQL
- Compléments sur SQL : schéma , tri, calcul
- Les méthodes: Modèle E/A (Entité Association), Schéma conceptuel, Schéma SQL
- Algèbre relationnelle
- Etude de cas, index, clés, et implémentation
- Dépendances fonctionnelles et formes normales

Sources du cours

- Cours de Witold Litwin (Université Paris 9 Dauphine)
- Cours de Maria berger (AES Paris 6)
- Cours de Fabrice Jouanot (EPFL et UJF Grenoble)

Algèbre relationnelle: définitions

- Ensemble d'opérateurs qui s'appliquent aux relations
- Résultat : nouvelle relation qui peut à son tour être manipulée
- L'algèbre relationnelle permet de faire des recherches dans les relations

L'algèbre relationnelle en 5 points

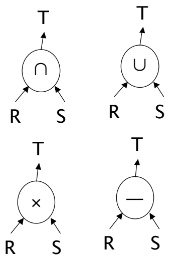
- **Opérandes**: relations du modèle relationnel
- **Fermeture**: le résultat de toute opération est une nouvelle relation
- **Complétude**: permet toute opération sauf les fermetures transitives
- **Opérations unaires** (une seule opérande):
  sélection (noté σ), projection (π), renommage ($\mathbf{a}$)
- **Opérations binaires**:
  produit cartésien (**x**), jointures (|X|), union ($\cup$), intersection ($\cap$), différence ($-$), division (/)

Spécificité des opérateurs

- Opérateurs ensemblistes:
  UNION, INTERSECT, DIFFERENCE, TIMES
- Ces opérateurs sont reformulés spécifiquement pour le modèle relationnel
- Opérateurs relationnels spécifiques
  RESTRICT, PROJECT, JOIN, DIVIDE

Opérateurs ensemblistes

- Soient R et S deux relations de même schéma
  - Union: T = R ∪ S
  - Intersection: T = R∩S
  - Différence: T=R-S
- Produit cartésien T=RxS



Union ∪

- opération binaire
- syntaxe : R ∪ S
- sémantique : réunit dans une même relation les tuples de R et ceux de S
- exemple :

R1

A	B
a	b
b	b
y	z

R2

A	B
u	v
y	z

R1 ∪ R2

A	B
a	b
b	b
y	z
u	v

Union: exemple

N°Ens	Nom	Prénom	Matière
12	Duciel	Betty	Economie
18	Nenfant	Ludivine	Droit
16	Duciel	Sandra	Droit
17	Nenfant	Hélène	Politique

N°Ens	Nom	Prénom	Matière
13	Duciel	Babar	Economie
14	Duciel	Noël	Droit
15	Duciel	Candide	Droit
19	Moncostume	Jimmy	Politique

Filles ∪ Garçons

N°Ens	Nom	Prénom	Matière
12	Duciel	Betty	Economie
13	Duciel	Babar	Economie
14	Duciel	Noël	Droit
15	Duciel	Candide	Droit
16	Duciel	Sandra	Droit
17	Nenfant	Hélène	Politique
18	Nenfant	Ludivine	Droit
19	Moncostume	Jimmy	Politique

Intersection ∩

- syntaxe : R ∩ S
- sémantique : sélectionne les tuples qui sont à la fois dans R et S
- exemple :

R1

A	B
a	b
y	z
b	b

R2

A	B
u	v
y	z

R1 ∩ R2

A	B
y	z

Différence -

- syntaxe : R – S
- sémantique : sélectionne les tuples de R qui ne sont pas dans S
- exemple :

R1

A	B
a	b
y	z
b	b

R2

A	B
u	v
y	z

R1 – R2

A	B
a	b
b	b

Produit cartésien ×

- but: construire toutes les combinaisons de tuples de deux relations
- syntaxe : R × S
- Sémantique : tuple de R est combiné avec chaque tuple de S
- Exemple :

R

A	B
a	b
b	c
c	b

n tuples

S

C	D	E
c	d	e
b	a	b
a	a	c

m tuples

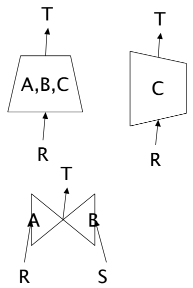
R × S

A	B	C	D	E
a	b	c	d	e
a	b	b	a	b
a	b	a	a	c
b	c	c	d	e
b	c	b	a	b
b	c	a	a	c
c	b	c	d	e
c	b	b	a	b
c	b	a	a	c

n x m tuples

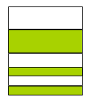
Opérateurs spécifiques

- Projection: ∏[A,B,C](R)
- Restriction: σ[C](R)
- Jointure naturelle: R]A,B[S = σ[A=B](RxS)
- Division: R/S

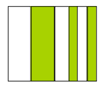


Opérateurs relationnels

- Opérateurs unaires
  - Restriction



  - Projection



Opérateurs relationnels

- Opérateurs binaires
  - Jointure (naturelle)

a1	b1
a2	b1
a3	b2

c1	b1
c2	b1
c3	b2

a1	b1	c1
a1	b1	c2
a2	b1	c1
a2	b1	c2
a3	b2	c3

  - Division

a	x
a	y
a	z
b	x
c	y

x
y

a

Restriction

- Syntaxe: σ [test](relation)
- Sémantique: relation composée de n-uplets vérifiant une condition

Quels sont les coureurs suisses ?

Numéro coureur	Nom Coureur	Code équipe	Code pays
8	ULLRICH Jan	TEL	ALL
31	JALABERT Laurent	ONC	FRA
61	ROMINGER Tony	COF	SUI
91	BOARDMAN Chris	GAN	G-B
114	CIPOLLINI Mario	SAE	ITA

Relation résultat

R = σ [CodePays = "SUI"](COUREUR)

Ou SELECTION [CodePays = "SUI"](COUREUR)

Restriction: syntaxe

- < prédicat-élémentaire opérateur-logique prédicat-élémentaire >
  - opérateur-logique Î { et, ou }
  - prédicat-élémentaire :

  <[non] attribut opérateur-de-comparaison constante|attribut>
    - attribut est un attribut de la relation R
    - opérateur-de-comparaison Î {=, ≠,<,>, ≤, ≥}
- schéma (résultat) = schéma (opérande)
- population (résultat) ⊆ population (opérande)
- Calculée par des opérations ensemblistes

Restriction: exemple

On ne veut que les pays dont la valeur de surface est inférieure à 100:
Petit-pays = s [surface < 100] (Pays)

Pays
Petit-pays

nom	*capitale*	*population*	*surface*
Autriche	Vienne	8	83
~~UK~~	~~Londres~~	~~56~~	~~244~~
Suisse	Berne	7	41

Projection

- syntaxe: **π [attributs] (R)**
  - attributs: liste l'ensemble d'attributs de R à conserver dans le résultat
- sémantique : crée une nouvelle relation de population l'ensembles des tuples de R réduits aux seuls attributs de la liste spécifiée

Nom et nationalité des coureurs ?

Relation résultat

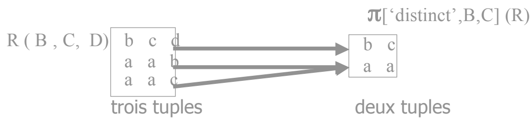
Nom Coureur	Code pays
ULLRICH Jan	ALL
JALABERT Laurent	FRA
ROMINGER Tony	SUI
BOARDMAN Chris	G-B
CIPOLLINI Mario	ITA

R = π [NomCoureur, Nationalité] (COUREUR)

PROJECTION[NomCoureur, Nationalité](COUREUR

Projection: propriétés

- schéma (résultat) $\subseteq$ schéma (opérande)
- nb tuples (résultat) = nb tuples (opérande)
- Elimination des doubles: c'est une option (« distinct » en SQL)



Projection: exemple

On ne veut que les attributs nom et capitale:

Capitales = π [nom, capitale] Pays

Pays

Capitales

nom	*capitale*	*population*	*surface*
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Restriction-projection

- On veut les capitales des petits pays:

  Petit-pays = s [surface < 100] (Pays)

  Capitale-petit-pays = π[nom, capitale](Petit-pays)

  = π[nom, capitale](s [surface < 100] (Pays))

Pays

Petit-pays

Capitale-petit-pays

nom	*capitale*	*population*	*surface*
Autriche	Vienne	8	83
~~UK~~	~~Londres~~	~~56~~	~~244~~
Suisse	Berne	7	41

Jointure

- **Jointure** dite « naturelle »:

Numéro coureur	Nom Coureur	Code équipe	Code pays
8	ULLRICH Jan	TEL	ALL
31	JALABERT Laurent	ONC	FRA
61	ROMINGER Tony	COF	SUI
91	BOARDMAN Chris	GAN	G-B

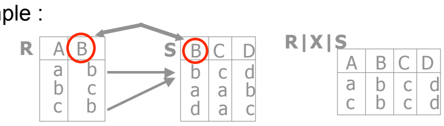
Code pays	Nom Pays
ALL	Allemagne
FRA	France
SUI	Suisse
G-B	Grande - Bretagne

Relation résultat

Numéro coureur	Nom Coureur	Code équipe	Code pays	Nom Pays
8	ULLRICH Jan	TEL	ALL	Allemagne
31	JALABERT Laurent	ONC	FRA	France
61	ROMINGER Tony	COF	SUI	Suisse
91	BOARDMAN Chris	GAN	G-B	Grande - Bretagne

Jointure naturelle

- but: créer toutes les combinaisons significatives entre tuples de deux relations
  - significatives = portent la même valeur pour les attributs de même domaine !
- précondition: les deux relations ont au moins un attribut de même domaine
- exemple :



Jointure naturelle

- **La jointure A |X| B (ou A][B) des deux relations**
  A (W, Y) et B (Z, Y)
  est la relation C avec les attributs :
  C (W, Y, Z)
  et les tuples (W:w, Y:y, Z:z) tels que (w, y) est dans A et (y, z) est dans B
- **W, Y, Z peuvent être composés**
- **La jointure naturelle est associative et commutative**

Jointures

- Syntaxe: C = A]Y=Z[B,
  - les attributs sont omis si …?
- θ-jointure : jointure sous condition θ autre que l’égalité

Theta-jointure

- Opération binaire
- syntaxe : R]test[S
  - test: prédicat/condition de jointure
    < prédicat-élémentaire et/ou prédicat-élémentaire >
- sémantique : combine les tuples qui satisfont le prédicat
- schéma (R]test[S) = schéma (R) ∪ schéma (S)
- Associative? Commutative?

Theta-jointure

- but: créer toutes les combinaisons significatives entre tuples de deux relations
  - significatives = critère de combinaison explicitement défini en paramètre de l’opération
- exemple :

R

A	B
a	b
b	c
c	b

S

C	D	E
b	c	d
b	a	b
c	a	c

R]B≠C[S

A	B	C	D	E
a	b	c	a	c
b	c	b	c	d
b	c	b	a	b
c	b	c	a	c

Jointures externes

- but: créer toutes les combinaisons significatives entre tuples de deux relations
  - Mettre en évidence les tuples qui n’apparaissent que dans une table (gauche ou droite)
  - Valeur NULL pour les attributs de l’autre table
- précondition: au moins un attribut de même domaine
- exemple :

R

A	B
a	b
b	c
c	b

S

B	C	D
b	c	d
a	a	b
d	a	c

R|gX|S

A	B	C	D
a	b	c	d
b	c	Null	
c	b	c	d

Division

- relation composée des n-uplets tels que le produit cartésien avec le diviseur soit un sous-ensemble de la relation dividende

Participer

Athlète	Epreuve
Dupont	200 m
Durand	400 m
Dupont	400 m
Martin	110 m H
Dupont	110 m H
Martin	200 m

Epreuve

Epreuve
200 m
400 m
110 m H

Athlète
Dupont

Relation résultat de Participer ÷ Epreuve

Quels sont les athlètes qui ont participé à toutes les épreuves ?

Division: définition formelle

- but: traiter les requêtes de style «les … tels que TOUS les …»
- soient $R(A_1, \ldots, A_n)$ et $V(A_1, \ldots, A_m)$
  avec n>m et A1, …, Am des attributs de même nom dans R et V
- $R/V = \{ <a_{m+1}, a_{m+2}, \ldots, a_n> /$
  $\forall <a_1, a_2, \ldots, a_m> \in V,$
  $\exists <a_1, a_2, \ldots, a_m, a_{m+1}, a_{m+2}, \ldots, a_n> \in R \}$
- $R/S = R_1$-P[A](R_1xS - R), avec R(A,B), S(B) et R_1 = P[A](R)

La division: exemples

R x S / S = R, schéma(R/S)=schéma(R)-schéma(S)
Associativité?

R

A	B	C
1	1	1
1	2	0
1	2	1
1	3	0
2	1	1
2	3	3
3	1	1
3	2	0
3	2	1

V

B	C
1	1
2	0

R/V

A
1
3

V'

B	C
1	1

R/V'

A
1
2
3

V''

B	C
3	5

R/V''

A
/

Exemple de division

R

STUDENT	COURSE	PASSED
Francois	RDB	yes
Francois	Prog	yes
Jacques	RDB	yes
Jacques	Math	yes
Pierre	Prog	yes
Pierre	RDB	no

V

COURSE	PASSED
Prog	yes
RDB	yes

R/V

STUDENT
Francois

Renommage

- but: résoudre des problèmes de compatibilité entre noms d'attributs de deux relations opérandes d'une opération binaire
- opération unaire
- syntaxe : α [nom_attribut : nouveau_nom] R
- sémantique : les tuples de R avec un nouveau nom de l'attribut
- schéma : schéma (α [n, m] R) de même schéma que (R) avec n renommé en m

Renommage

- précondition : le nouveau nom n'existe pas déjà dans R
- exemple : R2 = α [B: C] R1

R1

A	B
a	b
y	z
b	b

R2

A	C
a	b
y	z
b	b

Equivalences

- R ∩ S = S ∩ R, R ∪ S = S ∪ R , etc.
- σ [p1] (σ [p2] (R))= σ [p2] (σ [p1] (R)) = σ [p2 et p1] (R) = σ [p1] (R) ∩ σ [p2] (R)
- π [a] (σ [p] (R)) = σ [p] (π [a] (R))
  si attributs(p) ⊆ a
- R/S = R1-P[A](R1xS - R), avec R(A,B), S(B) et R1 = P[A](R)

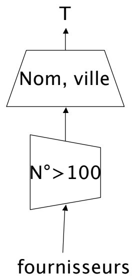
Ensemble minimal

- 8 opérateurs relationnels
- 5 opérateurs de base: union, différence, restriction, produit cartésien
- 3 opérateurs dérivés: intersection, jointure, division
- Exercice: le montrer

SQL: Exemple 1

```
Select nom, ville
  from
  fournisseurs
  where n° > 100 ;
```

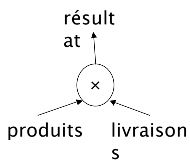
∏[nom,ville](
σ[n°>100](fournisseurs))



SQL: Exemple 2

```
Select * from produits, livraisons ;
```

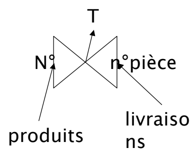
produits x livraisons



SQL: Exemple 3

```
Select * from produits, livraisons
  where  n° = n° pièce;
```

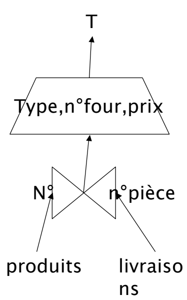
Produits]N°,n°pièce[livraisons



SQL: Exemple 4

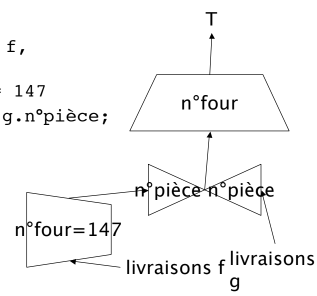
```
Select n°, type, n°
  four, prix from
  produits, livraisons
  where  n° = n° pièce;
```

∏[type,n°four,prix](
produits]N°,n°pièce[livraisons
)



SQL: Exemple avec variables

```
Select g.n°four
from livraisons f,
  livraisons g
where f.n°four = 147
And f.n°pièce = g.n°pièce;
```



Propriétés

- Associativité de la jointure
  - R]A,B[S = σ[A=B](RxS)
  - Associativité de x
- Commutativité avec la projection et la restriction
- A et B: clés primaires ou étrangères
  - Doivent être du même type

Ordre des opérations

- La jointure est en général coûteuse
- Faire autant que possible les restrictions et les projections avant
- Pourquoi une requête est-elle **meilleure** qu'une autre ?
  - Une requête n'est pas l'unique solution d'un problème.
  - **efficacités** différentes

Exemple

Fournisseur (N°fno, Nom, Adresse, Ville)
Produit (N°prod, Designation, Prix, Poids, Couleur)
Commande (N°comm, N°fno, N°prod, , Quantité)
Produit = 8 lignes * 5 colonnes
Commande = 10 lignes * 4 colonnes

Référence (N° prod), prix et quantité des produits commandés en plus de 10 exemplaires par commande ?

Solution 1

- R1= Commande]N°prod, N°prod[Produit
- R2 = s [Quantité > 10] (R1)
- R3 = P [N°prod, Prix, Quantité] (R2)
- R1 = jointure sur la table Commande et la table Produit = 10*8 = 80 tuples au pire

Solution 2

- R1 = P [N°prod, Quantité] (s[Quantité>10] (Commande))
- R2 = R1]N°prod, N°prod[P [N°prod, Prix] (Produit)
- R3 = P [N°prod, Prix, Quantité] (R2)
- R2 = jointure sur le couple (N°prod, Prix) de la table Produit: 8 tuples (sur 2 attributs) et sur le couple (N°prod, Quantité) de la table Commande : 2 tuples (sur 2 attributs)
- Total = 2x8 = 16 éléments (sur 4 attributs)

Restructuration d'un arbre

- Heuristiques:
  - remonter projections et restrictions
  - Descendre les jointures
- Propriétés utilisées
  - Associativité de |X|
  - Commutativité |X| et σ
  - Commutativité |X| et π
  - Commutativité π et σ