

L'algèbre relationnel

HAI502I

Pascal Poncelet

Pascal.Poncelet@umontpellier.fr

<http://www.lirmm.fr/~poncelet>



Introduction

- Introduit par CODD en 1970
- Traitement de requêtes de lecture et écriture
- Deux principaux types d'opérateurs
 - Opérateurs ensemblistes : UNION, INTERSECTION, DIFFERENCE, PRODUIT CARTESIEN
 - Opérateurs relationnels : SELECTION, PROJECTION, JOINTURE et DIVISION
 - Opérateurs dérivés : renommage, jointure externe, semi-jointure gauche, droite
- Notation : t un tuple d'une relation et $t(A)$ dans R , le sous tuple de R relatif à A



Introduction

- Ensemble d'opérateurs définis sur l'ensemble des relations :
 - le résultat de toute opération algébrique est une relation(*propriété de fermeture*)
$$R \text{ op } Q \rightarrow S$$
 - une base formelle pour les requêtes
 - utiles pour l'implémentation et l'optimisation des requêtes



Opérateurs ensemblistes

- Pour l'union, l'intersection, la différence les relations doivent être compatibles :
 - les relations doivent avoir même degré
 - les attributs associés deux à deux doivent être du même type

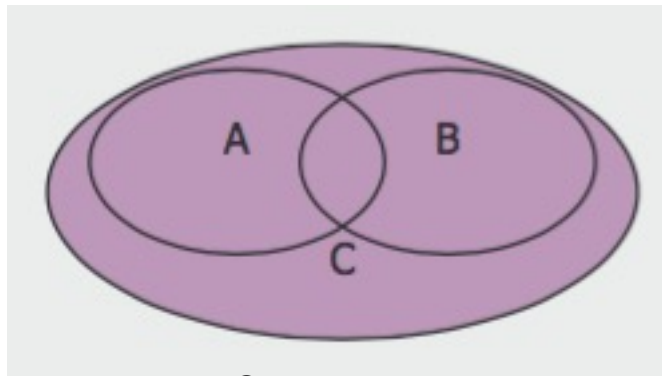
UNION

- UNION

$R \cup S$: ensemble des tuples qui appartiennent soit à R , soit à S , soit à R et S :

$$R \cup S = \{t/t \in R \text{ OU } t \in S\}$$

- Opérateur commutatif ($R1 \cup R2 = R2 \cup R1$)
- Opérateur associatif : $[(R1 \cup R2) \cup R3] = [R2 \cup (R1 \cup R3)]$



$$C = A \cup B$$

Exemple

Pilote1 : ensemble des pilotes habitant PARIS

PILOTE1	PLNUM	ADR
	100	PARIS
	101	PARIS
	120	PARIS
	110	PARIS

Pilote2 : ensemble des pilotes assurant un vol au départ de PARIS de TOULOUSE

PILOTE2	PLNUM	VD
	130	TOUL
	140	TOUL
	150	TOUL
	100	TOUL
	120	TOUL
	130	PARIS
	101	PARIS
	140	PARIS
	110	PARIS

Pilote1 UNION pilote 2

**PILOTE1 UNION
PILOTE2**

Ensemble des pilotes
habitant PARIS ou
assurant
un vol au départ de
PARIS ou TOULOUSE

PLNUM	VILLE
100	PARIS
101	PARIS
120	PARIS
110	PARIS
130	TOUL
140	TOUL
150	TOUL
100	TOUL
120	TOUL
130	TOUL
130	PARIS
140	PARIS

Pilote1

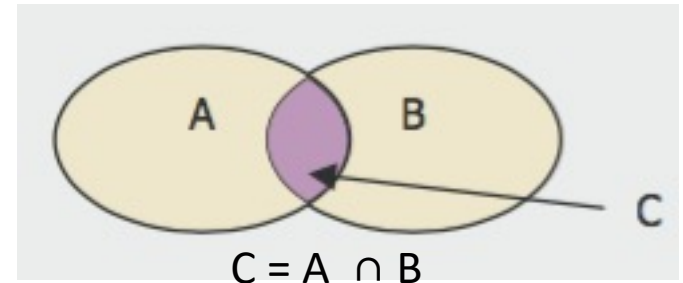
Pilote2

Les duplicats sont éliminés

L'Union permet de faire de l'ajout de tuples

INTERSECTION

- $R \cap S$: ensemble des tuples qui appartiennent à R et à S :
$$R \cap S = \{t / t \in R \text{ et } t \in S\}$$
- Opérateur commutatif : $R1 \cap R2 = R2 \cap R1$
- Opérateur associatif : $[(R1 \cap R2) \cap R3] = [R2 \cap (R1 \cap R3)]$



Avec Pilote1 INTERSECTION Pilote2

**PILOTE1
INTERSECTION
PILOTE2**

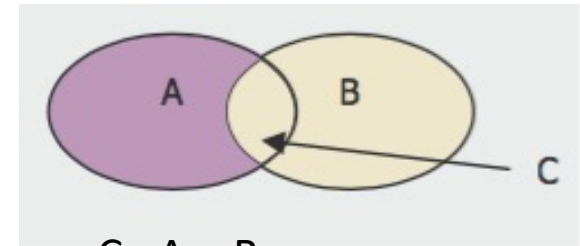
PLNUM	VILLE
101	PARIS
110	PARIS

L'intersection permet de traduire le ET logique

DIFFERENCE

- $R - S$: ensemble des tuples qui appartiennent à R sans appartenir à S . Complémentaire de l'intersection :

$$R - S = \{t / t \in R \text{ ET } t \notin S\}$$



- Opérateur non commutatif : $R1 - R2 \neq R2 - R1$
- Opérateur non associatif $[(R1 - R2) - R3] \neq [R2 - (R1 - R3)]$

PILOTE1 – PILOTE2

PLNUM	ADR
100	PARIS
120	PARIS

PILOTE 1 – PILOTE 2 : ensemble des pilotes habitant PARIS et n'assurant pas de vol au départ de PARIS ou TOULOUSE

PRODUIT CARTESIEN

- $R \otimes S$: ensemble de tous les tuples obtenus par concaténation des tuples de R et de S. Ensemble de paires ordonnées :

$$R \otimes S = \{(t(r), t(s)) \text{ avec } t(r) \in R \text{ et } t(s) \in S\}$$

- Opérateur commutatif : $R1 \otimes R2 = R2 \otimes R1$
- Opérateur associatif : $[(R1 \otimes R2) \otimes R3] = [R2 \otimes (R1 \otimes R3)]$

PILOTE3

PLNUM	ADR
103	NICE
106	NICE

AVION

AVNOM	CAP
AIRBUS	350
CARAV	250

Produit
Cartésien

PLNUM	ADR	AVNOM	CAP
103	NICE	AIRBUS	350
103	NICE	CARAV	250
106	NICE	AIRBUS	350
106	NICE	CARAV	250



Renommage

- Opérateur renommer :

Changer le nom d'un (ou plusieurs) attribut d'une relation

- Utile avant les jointures (homonymie, synonymie), ou avant les opérations ensemblistes (même nom requis)
- Notations :

Renommer[PLUM: NEWPLNUM] PILOTE

α [nom_attr1: nouveau_nom_pour_attr1, ...] R



Opérateurs relationnels

- Deux catégories
 - unaires de restriction
 - binaires d'extension
- Les opérateurs de restrictions permettent :
 - soit un découpage horizontal d'une relation (SELECTION)
 - soit un découpage vertical d'une relation (PROJECTION)



SELECTION

- Soit θ un comparateur binaire $\theta = \{<, <=, >, >=, <>\}$ applicable à l'attribut (ou à l'ensemble d'attributs) A et au tuple c (de la relation R)
- La sélection $R(A \theta c)$ est l'ensemble des tuples de R pour lesquels θ est vérifié entre la (les) composante(s) A et le tuple c :

$$R(A \theta c) = \{ t / t \in R \text{ ET } tA \theta c \}$$

SELECTION

PILOTE4

PLNUM	PLNOM	ADR
100	JEAN	PARIS
101	PIERRE	PARIS
120	PAUL	PARIS
130	SERGE	TOUL
140	MICHEL	TOUL

R

PLNUM	PLNOM	ADR
130	SERGE	TOUL
140	MICHEL	TOUL

R = PILOTE4 (ADR = « TOUL »)

SELECTION - Notations

$R = \text{PILOTE4 (ADR = « TOUL »)}$

$R = \text{SELECTION(PILOTE4/ADR = « TOUL »)}$

Notation : $\sigma_Q(R)$ où Q est le critère de la forme : Ai
θ Valeur

$R = \sigma_{\text{ADR}=\text{« TOUL »}}(\text{PILOTE4})$

- Toutes les notations sont équivalentes mais il ne faut pas les mélanger

SELECTION - remarques

- Il est possible d'utiliser des opérateurs logiques :
ET, OU, NON

$R = \text{PILOTE4}(\text{ADR}=\text{« TOUL » ET NOM}=\text{« MICHEL »})$

$R = \sigma_{(\text{ADR}=\text{« TOUL » ET NOM}=\text{« MICHEL »})}(\text{PILOTE4})$

- Pas indispensable pour le moment

PROJECTION

- Soit $R(A)$ une relation et un ensemble d'attributs A_1, \dots, A_n de R tels que $(A_1, A_2, \dots, A_n) \subset A$
- La projection $R' (A_1, A_2, \dots, A_n)$ est la relation obtenue à partir de $R (A)$ en éliminant de $R(A)$ les attributs autres que ceux spécifiés par A_1, A_2, \dots, A_n

$$R (A_1, \dots, A_n) = \{t(a_1, \dots, a_n)\}$$

- Suppression des tuples dupliqués



PROJECTION

Avion1

AVNUM	AVNOM	CAP	LOC
100	AIRBUS	350	TOUL
101	AIRBUS	350	TOUL
104	AIRBUS	150	PARIS
105	CARAV	250	PARIS

R

AVNOM	CAP
AIRBUS	350
AIRBUS	350
AIRBUS	150
CARAV	250

$R = \text{AVION1}(\text{AVNOM}, \text{CAP})$

PROJECTION - Notations

$R = \text{AVION1 (AVNOM, CAP)}$

$R = \text{PROJECTION(AVION1/AVNOM, CAP)}$

Notation : $\pi_{A_1, A_2, \dots, A_p}(R)$

$R = : \pi_{\text{AVNOM, CAP}}(\text{AVION1})$

- Toutes les notations sont équivalentes mais il ne faut pas les mélanger



Opérateurs binaires d'extension

- L'opérateur JOIN et DIVISION
- JOIN : permettre de pouvoir relier des relations entre elles – Attention à la sémantique des requêtes
- DIVISION : opérateur qui permet de sélectionner les tuples d'une relation (dividende) qui satisfont un critère de couverture énoncé via le contenu d'une autre relation (diviseur). Le résultat est une troisième relation, appelée le quotient. Exprime le « tous les »



JOIN

- Soient les relations $R(A, B_1)$ et $S(B_2, C)$ avec B_1 et B_2 attributs définis sur le même domaine, soit $\theta = \{=, >, >=, <, <=, <>\}$ applicables aux valeurs des attributs B_1 et B_2
- Le JOIN de R sur B_1 avec S sur B_2 est la relation dont les tuples sont ceux obtenus par concaténation des tuples de R avec ceux de S pour lesquels la relation θ entre les composantes B_1 et B_2 est vérifiée :

$$R (B_1 \theta B_2) S = \{t / t \in R \otimes S \text{ ET } t(B_1) \theta(B_2)\}$$

- L'opérateur JOIN est équivalent à un produit cartésien suivi d'une sélection



JOIN - Vocabulaire

$$R (B_1 \theta B_2) S = \{t / t \in R \otimes S \text{ ET } t(B_1) \theta(B_2)\}$$

- Lorsque $\theta = \{=\}$ on parle d'**équijointure** autrement de thétajointure

$$R (B_1 = B_2) S = \{t / t \in R \otimes S \text{ ET } t(B_1) \theta(B_2)\}$$

- Il est possible d'avoir des autojointures

$$R (B_1 \theta B_2) R = \{t / t \in R \otimes R \text{ ET } t(B_1) \theta(B_2)\}$$



JOIN - Exemple

PILOTE1

PLNUM	PLNOM	ADR
100	JEAN	PARIS
101	PIERRE	PARIS
120	PAUL	PARIS

VOL1

VOLNUM	AVNUM	PLNUM
IT500	110	100
IT501	130	100
IT503	110	100
IT504	110	120
IT506	120	120
IT507	130	110

JOIN

JOIN (PILOTE1, VOL1/PLNUM=PLNUM)

- Ensemble des pilotes habitant PARIS en service avec les numéros des vols et des avions correspondants

PLNUM	PLNOM	ADR	VOLNUM	AVNUM	PLNUM
100	JEAN	PARIS	IT500	110	100
100	JEAN	PARIS	IT501	130	100
100	JEAN	PARIS	IT503	110	100
120	PAUL	PARIS	IT504	110	120
120	PAUL	PARIS	IT506	120	120



JOIN

JOIN (PILOTE1, VOL1/ PLNUM>PLNUM)

Quels sont les pilotes dont le numéro est supérieur à au moins un numéro de pilote dans vol (qui effectue un vol) ?

PLNUM	PLNOM	ADR	VOLNUM	AVNUM	PLNUM
101	PIERRE	PARIS	IT500	110	100
101	PIERRE	PARIS	IT501	130	100
101	PIERRE	PARIS	IT503	110	100
120	PAUL	PARIS	IT500	110	100
120	PAUL	PARIS	IT501	130	100
120	PAUL	PARIS	IT503	110	100
120	PAUL	PARIS	IT507	130	110


} 101>100

} 120>100

} 120 > 110

JOIN

- Quels sont les avions dont la capacité est supérieure à celle de l'avion 100 ?



AVNUM	CAP
100	200
101	350
104	150
105	250

AVNUM	CAP
100	200
101	350
104	150
105	250

curseur



R1 = AVION(AVNUM=100)

R3 = JOIN(AVION, R2/CAP >= CAP)

Jointure naturelle

- La jointure naturelle de 2 tables R et S est une table T dont les attributs sont l'union des attributs de R et de S et dont les tuples sont obtenus en concaténant un tuple de R et un tuple de S ayant mêmes valeurs pour les attributs de même nom
- Elimination des attributs doublons
- $T = \text{JOIN}(R, S)$



Jointure naturelle

JOIN(PILOTE1,VOL1)

- Le seul attribut commun est PLNUM

PILOTE1

PLNUM	PLNOM
100	JEAN
101	PIERRE
120	PAUL

VOL1

VOLNUM	AVNUM	PLNUM
IT500	110	100
IT501	130	100
IT503	110	100
IT504	110	120
IT506	120	120
IT507	130	110

PLNUM	PLNOM	ADR	VOLNUM	AVNUM
100	JEAN	PARIS	IT500	110
100	JEAN	PARIS	IT501	130
100	JEAN	PARIS	IT503	110
120	PAUL	PARIS	IT504	110
120	PAUL	PARIS	IT506	120



Jointure externe

- La jointure externe de 2 tables R et S est une table T obtenue par jointure de R et S et ajout des tuples de R et de S ne participant pas à la jointure avec des valeurs nulles pour les attributs de l'autre table s'il n'y a pas de correspondances
- Intérêt : composer des vues sans perte d'information
- $T = \text{EXT-JOIN}(R,S)$

Jointure externe

EXT-JOIN(PILOTE1,PILOTE2)

PILOTE1

PLNUM	PLNOM
100	JEAN
101	PIERRE
120	PAUL

PILOTE2

PLNUM	ADR
100	PARIS
101	NICE
122	LYON

PLNUM	PLNOM	ADR
100	JEAN	PARIS
101	PIERRE	NICE
120	PAUL	NULL
122	NULL	LYON

Semi jointure externe

- On distingue :
 - la jointure externe droite (REXT-JOIN) : elle garde seulement les tuples sans correspondant de la table de droite
 - la jointure externe gauche (LEXT-JOIN) : elle garde seulement les tuples sans correspondant dans la table de gauche



Jointure externe

REXT-JOIN(PILOTE1,PILOTE2)

PILOTE1

PLNUM	PLNOM
100	JEAN
101	PIERRE
120	PAUL

PILOTE2

PLNUM	ADR
100	PARIS
101	NICE
122	LYON

PLNUM	PLNOM	ADR
100	JEAN	PARIS
101	PIERRE	NICE
122	NULL	LYON

JOIN - Notations

RES = PILOTE1 (PLNUM=PLNUM) VOL1

RES = JOIN (PILOTE1, VOL1 / PLNUM = PLNUM)

Notation : RES = R \bowtie_{θ} S

RES = PILOTE1 \bowtie VOL1
PLNUM=PLNUM



JOIN - Notations

Jointure naturelle

RES = JOIN (PILOTE1,VOL1)

Notation : RES = 

RES = PILOTE1 VOL1

Jointure externe

RES = EXT-JOIN (PILOTE1, VOL1)

Notation sigma : RES = R  S

Left - Join



Right - Join



DIVISION

- Utiliser souvent pour exprimer le « tous les »
- Division d'une relation binaire par une relation unaire

$$R(A_1 / A_2) S = \{t / t \in R[B] \text{ ET } (\{t\} \otimes S) \subseteq R\}$$

avec $R(B, A_1)$ et $S(A_2)$

- La division de R par S est le sous-ensemble des éléments de $R(B)$ dont le produit cartésien avec S est inclus dans R



DIVISION

- Avions conduits par tous les pilotes : VOL1/PIL ?

Représente VOL1 (PLNUM /PLNUM) PIL

VOL1

Dividende

AVNUM	PLNUM
30	100
30	101
30	102
30	103
31	100
31	102
32	102
32	103
33	102

Diviseur PIL1

PLNUM
100



AVNUM
30
31

Quotient

Diviseur PIL2

PLNUM
102
103



AVNUM
30
32

Quotient



DIVISION

- Chaque fois que l'on aura « pour tous les x », il suffira de mettre l'attribut x dans le diviseur. Le dividende binaire doit contenir alors le même attribut (sur lequel porte la division) et l'attribut du résultat recherché

DIVISION

- Quels sont les noms des pilotes qui conduisent tous les avions de la compagnie ?

$VOL1 = VOL (AVNUM, PLNUM) <projection>$

$AV1 = AVION (AVNUM) <projection>$

$PILOTE1 = PILOTE (PLNUM, PLNOM) <projection>$

$PILOTE2 = VOL1 (AVNUM / AVNUM) AV1 <division \text{ pour avoir les numéros des pilotes qui conduisent tous les avions}>$

$PILOTE3 = PILOTE1 (PLNUM=PLNUM)PILOTE2 <join \text{ pour avoir les noms}>$

$RES = PILOTE3(PLNOM) <projection>$



CE QU'IL FAUT RETENIR

- L'ALGEBRE RELATIONNELLE EST COMPLETE
 - Les cinq (sept) opérations de base permettent de formaliser sous forme d'expressions toutes les questions que l'on peut poser avec la logique du premier ordre (sans fonction)
 - Le résultat de l'application d'un opérateur donne une relation
 - Possibilité d'impliquer les opérateurs directement
- NOM ET PRENOM DES BUVEURS DE VOLNAY 1988

PROJECTION (NOM,PRENOM, SELECTION(CRU="VOLNAY" ET
MILL =1988, JOIN(VINS,ABUS,BUVEURS)))



Equivalences

- Intersection :

$$R \cap S = R - (R - S) = S - (S - R) \text{ ou } R \cap S = (R \cup S) - ((R - S) \cup (S - R))$$

- Jointure naturelle :

$$\text{Soient } R(X,Y) \text{ et } S(Y,Z) \quad R \bowtie S = \pi [X,Y,Z] \sigma [Y = Y'] (R \times \alpha[Y : Y']S)$$

- Thêta jointure :

$$\text{Soient } R(X,Y) \text{ et } S(U,V) \quad R \bowtie_p S = \sigma_p (R \times S)$$

- Division :

$$\text{Soient } R(X,Y) \text{ et } S(Y) \quad R/S = \pi [X] R - \pi [X] (((\pi[X]R) \times S) - R)$$

ARBRE DE REQUETES

- Pour chaque requête un arbre de requête est créé. Il permet notamment de faire de l'optimisation de requêtes

Liste des noms des pilotes Parisiens :

$\pi_{\text{PInom}}(\sigma_{(\text{Ville}=\text{« PARIS »})}(\text{Pilote}))$

π_{PInom}

Arbre de requête
correspondant

$\sigma_{\text{Ville}}(\text{« PARIS »})$

PILOTE



ARBRE DE REQUETES

- Intuitivement une requête est coûteuse : remonter les opérations de sélection et de sélection
- Une jointure est un produit cartésien suivi d'une sélection !



Complexité

- Sélection : σ [condition] R
 - Au plus: balayer la relation et tester la condition sur chaque tuple
 - Complexité = $\text{card}(R)$
 - Taille du résultat : $[0 : \text{card}(R)]$
- Projection : $\pi [A_i, A_j \dots] R$
 - Balayer la relation + élimination des doublons
 - Complexité = $\text{card}(R)$. *0 si inclut dans une sélection*
 - Taille du résultat : $[1 : \text{card}(R)]$
- Jointure (naturelle ou thêta) entre R et S
 - Balayer R et pour chaque tuple de R faire :
Balayer S et comparer chaque tuple de S avec celui de R
 - Complexité = $\text{card}(R) \times \text{card}(S)$
 - Taille du résultat : $[0 : \text{card}(R) \times \text{card}(S)]$



-
- Des questions ?