




M114 – CM7

Algèbre Relationnelle

Denis PALLEZ

<http://denispallez.i3s.unice.fr>




Opérateurs utilisés

| Opérateurs Unaires | Opérateurs binaires (relations de même schéma) | Opérateurs binaires (schémas ≠) |
|--|--|------------------------------------|
| Projection (π) | Union (\cup) | Produit cartésien (\times) |
| Sélection (σ) ou Restriction | Intersection (\cap) | Quotient (\div) |
| Renommage (ρ) | Différence ($-$) | Jointure ($\bowtie_{condition}$) |

2015-2016

M114 – Intro. BD – <http://denispallez.i3s.unice.fr>

113




Définition

- En 1970, Codd définit l'Algèbre Relationnelle
 - Pour manipuler les relations (tables) du Modèle Relationnel
- Ensemble d'opérateurs qui s'appliquent aux relations (tables)
- Permet d'écrire une requête de manière algébrique ou de manière graphique
- À utiliser & optimiser AVANT SQL
- Opérateurs à 1 ou 2 opérandes
- Résultat : 1 relation

2015-2016

M114 – Intro. BD – <http://denispallez.i3s.unice.fr>

112



Base de données exemple

- CATEGORIE (IdCat, Libellé, TauxTva)
- PRODUIT (IdPro, IdCat#, Désignation, Marque, PrixUht, QStock)
- CLIENT (IdCli, Nom, Ville)
- VENTE (IdCli#, IdPro#, DateVente, Qté)
- COMMERCIAL (Matricule, Nom, Prénom, Ville)

2015-2016

M114 – Intro. BD – <http://denispallez.i3s.unice.fr>

114

Projection (π)

- Requête algébrique
 - $T = \pi_{A,B,C}(R)$
 - T ne contient que les attributs (de R) indiqués dans la liste
- Requête SQL
 - SELECT A,B,C FROM R ;
- Graphe algébrique



Sélection (σ) ou Restriction

- Requête algébrique
 - $T = \sigma_{\text{condition de restriction}}(R)$
 - T ne contient que les enregistrements de R qui vérifient la condition
- Requête SQL
 - SELECT * FROM R WHERE condition de restriction ;
- Graphe algébrique



Exemple

- Client (IdCli, Nom, Ville)
- $T = \pi_{\text{IdCli}, \text{Nom}}(\text{Client})$

| CLIENT | | |
|--------|--------|--------|
| IdCli | Nom | Ville |
| 1 | Dupont | Nice |
| 5 | Durand | Paris |
| 8 | Arthur | Nice |
| 9 | Martin | Toulon |

DECOUPAGE VERTICAL

| $T = \pi_{\text{IdCli}, \text{Nom}}(\text{Client})$ | |
|---|--------|
| IdCli | Nom |
| 1 | Dupont |
| 5 | Durand |
| 8 | Arthur |
| 9 | Martin |

Exemple

Client

| IdCli | Nom | Ville |
|-------|--------|-------|
| 1 | Dupont | Nice |
| 5 | Durand | Paris |
| 8 | Arthur | Nice |

DECOUPAGE HORIZONTAL

$T = \sigma_{\text{Ville} = \text{'Nice'}}(\text{Client})$

| IdCli | Nom | Ville |
|-------|--------|-------|
| 1 | Dupont | Nice |
| 8 | Arthur | Nice |

Question ?

- On veut afficher les Noms des clients niçois
- Requête algébrique
 - $\pi_{\text{Nom}}(\sigma_{\text{Ville} = \text{'Nice'}}(\text{Client}))$
- En SQL
 - SELECT Nom FROM Client WHERE Ville = 'Nice';
- Graphe algébrique

| CLIENT | | |
|--------|--------|--------|
| IdCli | Nom | Ville |
| 1 | Dupont | Nice |
| 5 | Durand | Paris |
| 8 | Arthur | Nice |
| 9 | Martin | Toulon |

| $\sigma_{\text{Ville} = \text{'Nice'}}(\text{Client})$ | | |
|--|--------|-------|
| IdCli | Nom | Ville |
| 1 | Dupont | Nice |
| 8 | Arthur | Nice |

| $\pi_{\text{Nom}}(\sigma_{\text{Ville} = \text{'Nice'}}(\text{Client}))$ | |
|--|--|
| Nom | |
| Dupont | |
| Arthur | |

Opérateurs utilisés

| Opérateurs Unaires | Opérateurs binaires (relations de même schéma) | Opérateurs binaires (schémas ≠) |
|---------------------------------------|--|---|
| Projection (π) | Union (\cup) | Produit cartésien (\times) |
| Sélection (σ) ou Restriction | Intersection (\cap) | Quotient (\div) |
| Renommage (ρ) | Différence ($-$) | Jointure ($\bowtie_{\text{condition}}$) |

Renommage (ρ)

- Requête algébrique
 - $T = \rho_{A/newA}(R)$
Renomme l'attribut **A** en **newA** dans la relation R
 - $T = \rho(R/R2)$
Renomme la relation R en R2
- Requête SQL
 - SELECT *, **A AS NEWA** FROM R ;
 - SELECT * FROM **R AS R2** ;
- Graphe algébrique

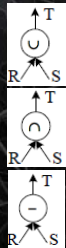


Uni-compatibilité

- 2 tables sont dites **uni-compatibles** \Leftrightarrow
 - Elles ont le même degré
 - les domaines de leurs attributs sont 2 à 2 compatibles
- Exemple
 - PRODUIT (no-prod : NumAuto(Entier), Nom : texte, prixUHT : réel)
 - MP (no-mp : entier, libellé : texte, prixAchat : money)

Union, Intersection, Différence

- Union
 - $T = R \cup S$
- Intersection
 - $T = R \cap S$
- Différence
 - $T = R - S$
 - $R - S \neq S - R$!!!!
- R et S doivent au mieux avoir le même schéma, au moins être uni-compatibles



Opérateurs utilisés

| Opérateurs Unaires | Opérateurs binaires (relations de même schéma) | Opérateurs binaires (schémas ≠) |
|--|---|---|
| Projection (π) Sélection (σ) ou Restriction Renommage (ρ) | Union (\cup) Intersection (\cap) Différence ($-$) | Produit cartésien (\times) Quotient (\div) Jointure ($\bowtie_{condition}$) |

Exemple

R

| IdCli | Nom |
|-------|--------|
| 1 | Dupont |
| 5 | Durand |
| 8 | Arthur |

S

| IdCli | Nom |
|-------|--------|
| 3 | Blanc |
| 8 | Arthur |

T=R ∪ S

| IdCli | Nom |
|-------|--------|
| 1 | Dupont |
| 3 | Blanc |
| 5 | Durand |
| 8 | Arthur |

T=R ∩ S

| IdCli | Nom |
|-------|--------|
| 8 | Arthur |

T=R - S

| IdCli | Nom |
|-------|--------|
| 1 | Dupont |
| 5 | Durand |

T=S - R

| IdCli | Nom |
|-------|-------|
| 3 | Blanc |

Produit cartésien

- Requête algébrique
 - $T = R \times S$
 - « Multiplier » chaque tuple de R par chaque tuple de S
- Requête SQL
 - `SELECT * FROM R, S ;`
- Graphe algébrique



Exemple

R

| IdCli | Nom | Ville |
|-------|--------|--------|
| 1 | Dupont | Nice |
| 5 | Durand | Paris |
| 8 | Arthur | Nice |
| 9 | Martin | Toulon |

S

| numFacture | dateFact | IdCli |
|------------|----------|-------|
| 10 | 12/8/92 | 8 |
| 13 | 15/8/92 | 1 |

$T = R \times S$

| R.IdCli | Nom | Ville | Num Facture | dateFact | S.IdCli |
|---------|--------|--------|-------------|----------|---------|
| 1 | Dupont | Nice | 10 | 12/8/92 | 8 |
| 1 | Dupont | Nice | 13 | 15/8/92 | 1 |
| 5 | Durand | Paris | 10 | 12/8/92 | 8 |
| 5 | Durand | Paris | 13 | 15/8/92 | 1 |
| 8 | Arthur | Nice | 10 | 12/8/92 | 8 |
| 8 | Arthur | Nice | 13 | 15/8/92 | 1 |
| 9 | Martin | Toulon | 10 | 12/8/92 | 8 |
| 9 | Martin | Toulon | 13 | 15/8/92 | 1 |

Exemple

- R (n° avion#, n° pilote#) → dividende
- S (n° pilote) → diviseur
- T (n° avion) → quotient

| R | n° avion | n° pilote |
|---|----------|-----------|
| 1 | 10 | 10 |
| 1 | 13 | 13 |
| 5 | 10 | 10 |
| 8 | 10 | 10 |
| 8 | 13 | 13 |
| 9 | 13 | 13 |

| S | n° pilote |
|----|-----------|
| 10 | 10 |
| 13 | 13 |

| T = R ÷ S | n° avion |
|-----------|----------|
| 1 | 10 |
| 8 | 13 |

- Quels sont les avions qui sont pilotés par tous les pilotes ?

Division

- Requête algébrique
 - $T = R \div S$
 - $R(A_1, A_2, \dots, A_p, A_{p+1}, \dots, A_n) S(A_{p+1}, \dots, A_n)$
 - $T(A_1, A_2, \dots, A_p)$ contient tous les tuples tq leur concaténation à chaque tuple de S donne toujours un tuple de R
- Requête SQL
 - Plusieurs solutions possibles
 - <http://sqlpro.developpez.com/cours/divrelationnelle/>
 - Solution retenue
 - Select ... where not exists (Select ... where not exists (select))
- Graphe algébrique



Jointure

- Requête algébrique
 - $T = R \bowtie_{condition} S$
 - $R \bowtie_{condition} S \equiv \sigma_{condition}(R \times S)$
- Requête SQL
 - SELECT * FROM R INNER JOIN S ON (condition de jointure) ;
- Graphe algébrique



Exemple

| R | | | S | | |
|-------|--------|--------|------------|----------|-------|
| IdCli | Nom | Ville | numFacture | dateFact | IdCli |
| 1 | Dupont | Nice | 10 | 12/8/92 | 8 |
| 5 | Durand | Paris | 13 | 15/8/92 | 1 |
| 8 | Arthur | Nice | | | |
| 9 | Martin | Toulon | | | |

$R \bowtie_{R.IdCli = S.IdCli} S$

| R.IdCli | Nom | Ville | numFacture | date | S.IdCli |
|---------|--------|-------|------------|---------|---------|
| 1 | Dupont | Nice | 13 | 15/8/92 | 1 |
| 8 | Arthur | Nice | 10 | 12/8/92 | 8 |

Exemple complet

- Donner les noms des clients ayant acheté des produits de la marque IBM

CATEGORIE (IdCat, Libellé, TauxTva)
 PRODUIT (IdPro, IdCat#, Désignation, Marque, PrixUht, QStock)
 CLIENT (IdCli, Nom, Ville)
 VENTE (IdCli#, IdPro#, DateVente, Qté)
 COMMERCIAL (Matricule, Nom, Prénom, Ville)

– Graphe algébrique

– Requête algébrique correspondante

• $\pi_{Nom} ((\sigma_{Marque = 'IBM'} (Client \bowtie_{IdCli} Vente \bowtie_{IdPro} Produit)))$

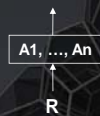
– Requête SQL

• SELECT Nom
 FROM (Client join Vente on (Client.IdCli=Vente.IdCli))
 join Produit on (Produit.IdPro=Vente.IdPro)
 WHERE Marque='IBM' ;



Tris et Regroupements

- Tris
 - Graphe algébrique



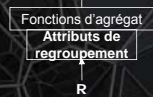
– Requête SQL

- SELECT * FROM R
- ORDER BY A1, ..., An ;

– Requête algébrique

- ?

- Regroupements
 - Graphe algébrique



– Requête SQL

- SELECT Fonctions d'agrégat
 FROM R
 GROUP BY Attribut ;

– Exemple

- SELECT Ville, COUNT(IdCli)
 FROM CLIENT
 GROUP BY Ville ;

Optimisation

- Donner les noms des clients ayant acheté des produits de la marque IBM
 - Graphe algébrique



– Requête algébrique correspondante

- $\pi_{Nom} (Client \bowtie_{IdCli} Vente \bowtie_{IdPro} (\sigma_{Marque = 'IBM'} (Produit)))$