Chapitre 4 (1ère partie)

Langages de consultation des bases de données relationnelles

Introduction

Les langages de consultation sont les outils qui permettent de:

- *définir*
- *et manipuler les données d'une BD.*

Types de langages utilisés par les SGBD – Relationnelles:

- *Le langage SQL*
- *Le langage QBE*
- *Le langage QUEL*

Utilises les concepts suivant:

- *Algèbre relationnelle*
- *Calcul relationnel*

Classification des langages de consultation

Deux classes :

- *Les langages procéduraux*
- *Les langages non-procéduraux*

Un langage procédural permet de définir:

- *l'information*
- *une méthode de recherche dans la base*
- *la façon d'accéder à l'information*

Ex: algèbre relationnelle

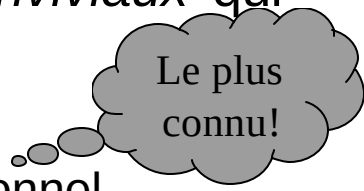
Un langage non-procédural permet de définir:

- *l'information désirée, mais le système de base de données se charge de la procédure de recherche.*

Ex: calcul relationnel

Classification des langages de consultation

Les SGBD sont munis de langages de consultation plus *conviviaux* qui sont les langages commerciaux.



Exemples:

- SQL : combinaison de l'algèbre relationnel et du calcul relationnel,
- QBE : utilise le calcul relationnel sur le domaine,
- QUEL : utilise le calcul relationnel sur les tuples.

Les différentes commandes offertes par ces langages sont classifiées en deux catégories:

- Le langage de définition des données (commandes de définition, de modification et d'élimination des tables de base.)
  - ***DDL*** *pour "Data Definition Langage".*
- Le langage de manipulation des données (commandes pour la manipulation des données.)
  - ***DML*** *pour "Data Manipulation Langage".*

Algèbre relationnelle

♦un langage de haut niveau (ses opérations sont appliqués sur des relations entières)

♦Pour une requête, c'est la responsabilité de l'utilisateur de spécifier dans quel ordre exécuter les opérations de la requête.

opérateurs de l'algèbre relationnel:

- *sélection*
- *projection*
- *produit cartésien*
- *union*
- *différence*

} *opérateur de base*

- *intersection*
- *théta-jonction*
- *équijonction*
- *quotient relationnel (division)*

Tous les opérateurs de l'algèbre relationnelle **définissent** ou **créent** une nouvelle relation.

Algèbre relationnelle

2 types d'opérateurs:

- *Les opérateurs unaires:* *portent sur une seule relation*
  - sélection
  - projection
- *Les opérateurs binaires:* *portent sur un couple de relations*
  - produit cartésien
  - union
  - différence
  - intersection
  - théta-jonction
  - équijonction
  - division

Algèbre relationnelle : Sélection

LA SÉLECTION : σ

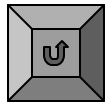
- *opérateur unaire*
- *permet de choisir les tuples qui satisfont un prédicat donné*
- *notée :* $s_{prédicat}$ *(relation)* *où*
  - le *prédicat* est une expression algébrique qui combine un attribut avec un symbole de comparaison (=, <, >).
  - plusieurs *prédicats* peuvent être combinés par les opérateurs logiques : " et = ∩ " et " ou = ∪ "

Algèbre relationnelle : Sélection

Exemple 1 : Trouver les tuples de la relation Crédit pour lesquels l'agence est Perryridge.

$$\sigma_{\text{agence = Perryridge}}\ \textbf{(Crédit)}$$

Agence	Prêt	Client	Montant
Downtown	17	John	1000
Redwood	23	Smith	2000
Perryridge	15	Hayes	1500
Downtown	14	Jackson	1500
Mianus	93	Curry	500
Round Hill	11	Turner	900
Pownal	29	Williams	1200
North Town	16	Adams	1300
Downtown	18	Johnson	2000
Perryridge	25	Glenn	2500
Brighton	10	Brooks	2200



Résultat : une nouvelle relation de 2 tuples.

Agence	Prêt	Client	Montant
Perryridge	15	Hayes	1500
Perryridge	25	Glenn	2500

Algèbre relationnelle : Sélection

Exemple 2 : Trouver les tuples de la relation Crédit dont le montant est supérieur à 1200 $

$$\sigma_{\text{montant} > 1200}(\text{Crédit})$$

Agence	Prêt	Client	Montant
Downtown	17	John	1000
Redwood	23	Smith	2000
Perryridge	15	Hayes	1500
Downtown	14	Jackson	1500
Mianus	93	Curry	500
Round Hill	11	Turner	900
Pownal	29	Williams	1200
North Town	16	Adams	1300
Downtown	18	Johnson	2000
Perryridge	25	Glenn	2500
Brighton	10	Brooks	2200

Résultat : Nouvelle relation de 7 tuples.

Agence	Prêt	Client	Montant
Redwood	23	Smith	2000
Perryridge	15	Hayes	1500
Downtown	14	Jackson	1500
North Town	16	Adams	1300
Downtown	18	Johnson	2000
Perryridge	25	Glenn	2500
Brighton	10	Brooks	2200

Algèbre relationnelle : Sélection

Exemple 3 :Trouver les tuples de la relation Crédit pour lesquels l'agence est Perryridge dont le montant est supérieur à 1200 $

$\sigma_{\text{agence = Perryridge } \cap \text{ montant > 1200}}$ (Crédit)

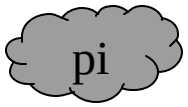
Agence	Prêt	Client	Montant
Downtown	17	John	1000
Redwood	23	Smith	2000
Perryridge	15	Hayes	1500
Downtown	14	Jackson	1500
Mianus	93	Curry	500
Round Hill	11	Turner	900
Pownal	29	Williams	1200
North Town	16	Adams	1300
Downtown	18	Johnson	2000
Perryridge	25	Glenn	2500
Brighton	10	Brooks	2200

Résultat : Nouvelle relation de 2 tuples.

Agence	Prêt	Client	Montant
Perryridge	15	Hayes	1500
Perryridge	25	Glenn	2500

Algèbre relationnelle : Projection

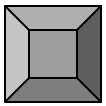
LA PROJECTION : Π



- *opérateur unaire*
- *recopie la relation avec seulement les attributs spécifiés*
- *notée :* $P_{attribut\ 1,...,\ attribut\ n}$ *(relation)*

Exemple : *Faire la liste des tuples de la relation Crédit concernant les clients avec leurs montants.*

$\Pi_{montant,client}$ (Crédit)



Résultat : *Nouvelle relation de 11 tuples qui contient les colonnes montant et client seulement.*

Montant	Client
1000	Jones
2000	Smith
1500	Hayes
1500	Jackson
500	Curry
900	Turner
1200	Williams
1300	Adams
2000	Johnson
2500	Glenn
2200	Brooks

Algèbre relationnelle : Projection

Une opération peut aussi s'effectuer sur le résultat d'une opération!

Exemple : Trouver la liste des clients qui ont le même nom que leur banquier pour la relation Affectation

$$\Pi_{client}\ (\sigma_{client\ =\ employé}\ (\text{Affectation}))$$

2 opérateurs:
Π σ

Client	Employé
Turner	Johnson
Hayes	Jones
Johnson	Johnson

La relation Affectation

Algèbre relationnelle : Projection

Résultat intermédiaire : Nouvelle relation de 1 tuple qui contient le nom du client et de l'employé.

Client	Employé
Johnson	Johnson

Résultat de $\sigma_{client = employé}$ (Affectation)

Résultat final : Nouvelle relation de 1 tuple qui contient le nom du client seulement.

Client
Johnson

Résultat de Π_{client} ($\sigma_{client = employé}$ (Affectation))

Algèbre relationnelle : Produit cartésien

LE PRODUIT CARTÉSIEN : X

- *opérateur binaire*
- *combinaison des tuples de deux relations*

  Si nous avons n1 tuples pour r1

  et si n2 tuples pour r2

  alors r = r1 X r2 comptera n1 * n2 tuples possibles

  Ex: r1 a 2 tuples

  r2 a 3 tuples

  r = r1 X r2, donc 2 * 3 = 6 tuples (tableau)

Algèbre relationnelle : Produit cartésien

Exemple 1 : Trouver le produit cartésien entre les relations Affectation et Clientèle.

Client	Employé
Turner	Johnson
Hayes	Jones
Johnson	Johnson

La relation Affectation

Client	Rue	Localité
Jones	Main	Harrison
Smith	North	Rye
Hayes	Main	Harrison
Curry	North	Rye
Lindsay	Park	Pittsfield
Turner	Putnam	Stamford
Williams	Nassau	Princeton
Adams	Spring	Pittsfiel
Johnson	Alma	Palo Alto
Glenn	Sand Hill	Woodside
Brooks	Senator	Brooklyn
Green	Walnut	Stamford

La relation Clientèle

Résultat : Une nouvelle relation qui combine la relation Affectation à la relation Clientèle.

- *Les tuples sont obtenus en combinant toutes les paires de tuples possibles, 3 tuples pour Affectation et 12 tuples pour Clientèle*

Affectation X Clientèle: 3*12 = 36 tuples possibles

Affectation X Clientèle

Affectation Client	Affectation Employé	Clientèle Client	Clientèle Rue	Clientèle Localité
Turner	Johnson	Jones	Main	Harrison
Turner	Johnson	Smith	North	Rye
Turner	Johnson	Hayes	Main	Harrison
Turner	Johnson	Curry	North	Rye
Turner	Johnson	Lindsay	Park	Pittsfield
Turner	Johnson	Turner	Putnam	Stamford
Turner	Johnson	Williams	Nassau	Princeton
Turner	Johnson	Adams	Spring	Pittsfield
Turner	Johnson	Johnson	Alma	Palo Alto
Turner	Johnson	Glenn	Sand Hill	Woodside
Turner	Johnson	Brooks	Senator	Brooklyn
Turner	Johnson	Green	Walnut	Stamford
Hayes	Jones	Jones	Main	Harrison
Hayes	Jones	Smith	North	Rye
Hayes	Jones	Hayes	Main	Harrison
Hayes	Jones	Curry	North	Rye
Hayes	Jones	Lindsay	Park	Pittsfield
Hayes	Jones	Turner	Putnam	Stamford
Hayes	Jones	Williams	Nassau	Princeton
Hayes	Jones	Adams	Spring	Pittsfield
Hayes	Jones	Johnson	Alma	Palo Alto
Hayes	Jones	Glenn	Sand Hill	Woodside
Hayes	Jones	Brooks	Senator	Brooklyn
Hayes	Jones	Green	Walnut	Stamford
Johnson	Johnson	Jones	Main	Harrison
Johnson	Johnson	Smith	North	Rye
Johnson	Johnson	Hayes	Main	Harrison
Johnson	Johnson	Curry	North	Rye
Johnson	Johnson	Lindsay	Park	Pittsfield
Johnson	Johnson	Turner	Putnam	Stamford
Johnson	Johnson	Williams	Nassau	Princeton
Johnson	Johnson	Adams	Spring	Pittsfield
Johnson	Johnson	Johnson	Alma	Palo Alto
Johnson	Johnson	Glenn	Sand Hill	Woodside
Johnson	Johnson	Brooks	Senator	Brooklyn
Johnson	Johnson	Green	Walnut	Stamford

Algèbre relationnelle : Produit cartésien

- *permet de combiner l'information de plusieurs tableaux afin d'effectuer des recherches plus complexes.*
- *Mais il peut conduire à des relations immenses qui prennent beaucoup d'espace mémoire.*

Exemple 2 : Faire la liste de tous les clients de l'employé Johnson et la ville où ils résident.

$$\Pi_{\text{Affectation.client, Clientèle.localité}}(\sigma_{\text{Affectation.client = clientèle.client}}(\sigma_{\text{Affectation.employé= Johnson}}(\text{Affectation X Clientèle})))$$

Algèbre relationnelle : Produit cartésien

Résultat intermédiaire 1:

Affectation Client	Affectation Employé	Clientèle Client	Clientèle Rue	Clientèle Localité
Turner	Johnson	Jones	Main	Harrison
Turner	Johnson	Smith	North	Rye
Turner	Johnson	Hayes	Main	Harrison
Turner	Johnson	Curry	North	Rye
Turner	Johnson	Lindsay	Park	Pittsfield
Turner	Johnson	Turner	Putnam	Stamford
Turner	Johnson	Williams	Nassau	Princeton
Turner	Johnson	Adams	Spring	Pittsfield
Turner	Johnson	Johnson	Alma	Palo Alto
Turner	Johnson	Glenn	Sand Hill	Woodside
Turner	Johnson	Brooks	Senator	Brooklyn
Turner	Johnson	Green	Walnut	Stamford
Johnson	Johnson	Jones	Main	Harrison
Johnson	Johnson	Smith	North	Rye
Johnson	Johnson	Hayes	Main	Harrison
Johnson	Johnson	Curry	North	Rye
Johnson	Johnson	Lindsay	Park	Pittsfield
Johnson	Johnson	Turner	Putnam	Stamford
Johnson	Johnson	Williams	Nassau	Princeton
Johnson	Johnson	Adams	Spring	Pittsfield
Johnson	Johnson	Johnson	Alma	Palo Alto
Johnson	Johnson	Glenn	Sand Hill	Woodside
Johnson	Johnson	Brooks	Senator	Brooklyn
Johnson	Johnson	Green	Walnut	Stamford

$\sigma_{\text{Affectation.employé = Johnson}}$ **(Affectation X Clientèle)**

Résultat intermédiaire 2:

Nouvelle relation de 2 tuples où Affectation.Client = Clientèle.Client

Résultat finale :

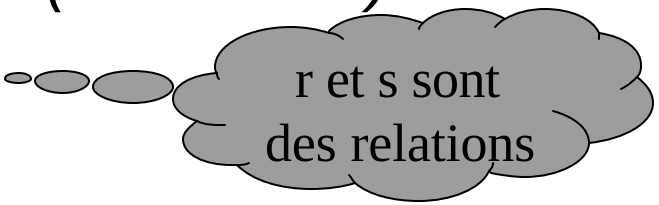
Nouvelle relation qui contient l'information suivante :

Affectation Client	Clientèle Localité
Turner	Stamford
Johnson	Palo Alto

Algèbre relationnelle : Union

L'UNION : U

- *opérateur binaire*
- *union logique (booléenne) de relations compatibles*
- *notée: r U s*
- *règles :*



  - r et s doivent présenter le même nombre d'attributs
  - les domaines du i ième attribut de r et du i ième attribut de s doivent être identiques
  - ne créé pas de doublons (tuples identiques)

Algèbre relationnelle : Union

Exemple : Faire la liste de tous les clients de Perryridge qui possèdent un compte ou qui ont obtenu un prêt ou les deux).

$r1 = \Pi_{client}(\sigma_{agence = Perryridge}(Crédit))$

Agence	Prêt	Client	Montant
Downtown	17	Jones	1000
Redwood	23	Smith	2000
Perryridge	15	Hayes	1500
Downtown	14	Jackson	1500
Mianus	93	Curry	500
Round Hill	11	Turner	900
Pownal	29	Williams	1200
North Town	16	Adams	1300
Downtown	18	Johnson	2000
Perryridge	25	Glenn	2500
Brighton	10	Brooks	2200

$r2 = \Pi_{client}(\sigma_{agence = Perryridge}(Dépôt))$

Agence	Compte	Client	Position
Downtown	101	Johnson	500
Mianus	215	Smith	700
Perryridge	102	Hayes	400
Round Hill	305	Turner	350
Perryridge	201	Williams	900
Redwood	222	Lindsay	700
Brighton	217	Green	750

Résultat intermédiaire : r1 = Hayes, Glenn

Client
Hayes
Glenn

r2 = Hayes, Williams

Client
Hayes
Williams

Résultat final : r1 U r2

Client
Hayes
Glenn
Williams

Remarquez que cette dernière relation ne contient pas de doublon.

Algèbre relationnelle : Différence

DIFFÉRENCE : -

- *opérateur binaire*
- *différence ensembliste (booléenne) de relations compatibles*
- *permet de trouver les tuples qui vérifient une relation sans en vérifier une autre (ex: tous les tuples de R qui ne sont pas dans S)*
- *notée : r – s*

Algèbre relationnelle : Différence

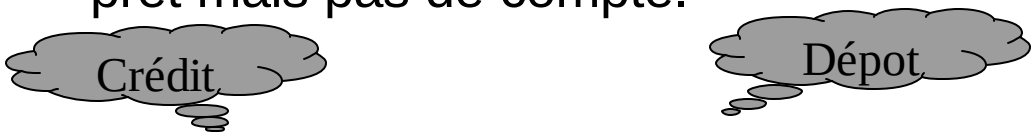
Exemple 1 : Faire la liste de tous les clients de Perryridge qui ont un compte mais pas de prêt.

À partir des résultats précédents de r2 et r1, nous avons

r2 - r1 = { Hayes, Williams } - { Hayes, Glenn }

Résultat final : Williams

Exemple 2 : Faire la liste de tous les clients de Perryridge qui ont un prêt mais pas de compte.



r1 - r2 = { Hayes, Glenn } - { Hayes, Williams }

Résultat final : Glenn

Algèbre relationnelle : Intersection

L'INTERSECTION : Ç

- *opérateur binaire*
- *opération booléenne: ce qui est commun à deux ensembles*
- *notée: r Ç s = r - (r - s) = s - (s - r)*

Exemple : Faire la liste de tous les clients de Perryridge qui ont un compte et un prêt.

À partir des résultats précédents de r1 et r2, nous avons

1. *r1 Ç r2 = { Hayes, Glenn } Ç { Hayes, Williams }*
2. *r - (r - s) = { Hayes, Glenn } - ({ Hayes, Glenn } - { Hayes, Williams })*
3. *s - (s - r) = { Hayes, Williams } - ({ Hayes, Williams } - { Hayes, Glenn })*

Résultat final : Hayes

Algèbre relationnelle : Theta-jonction

LA THETA-JONCTION : $\hat{o}C½_Q$

- *opérateur binaire*
- *permet de simplifier les consultations avec le produit cartésien*
- *combinaison du produit cartésien et de la sélection*
- *notée : $r\hat{o}C½_Q\ s = s_Q\ (r\ X\ s)$ où Q est le prédicat de la sélection*

Avantages:

- *Évite de construire un tableau mémoire immense qui correspondrait au produit cartésien.*
- *Combine en même temps l'association des données des différentes relations avec la sélection.*

Algèbre relationnelle : Theta-jonction

Exemple : Faire la liste de tous les clients de Perryridge qui ont un prêt avec leurs localités.

$$r = \Pi_{\text{Crédit. client, Clientèle.localité}} (\text{Crédit} \mid X \mid_{\theta} \text{Clientèle})$$

où θ = *Crédit.agence = Perryridge Ç Crédit.client = Clientèle.client*

Résultat :

Client	Localité
Hayes	Harrison
Glenn	Woodside

Algèbre relationnelle : Équijonction

L'ÉQUIJONCTION : ô C½

- *opérateur binaire*
- *cœur de la théorie des bases de données*
- *Utilisé lorsque le prédicat de la theta-jonction présente une opération d'égalité sur les valeurs des attributs de même types.*
- *notée : r ô C½ s*

  où pour chaque attribut A des relations r et s, r.A = s.A

Algèbre relationnelle : Équijonction

Séquence d'opération :

1. Sélectionne les tuples de même valeur selon les attributs spécifiés,
2. Combine chaque paire de tuples en un seul tuple par union des schémas relationnels
3. Effectue la projection spécifiée

Algèbre relationnelle : Équijonction

Exemple : Faire la liste de tous les clients qui ont un prêt avec leurs localités.

- *avec la théta-jonction on note :*

$$r1 = \Pi_{\text{Crédit. client, Clientèle.localité}}(\text{Crédit} \mid X \mid_{\text{Crédit.client = Clientèle.client}} \text{Clientèle})$$

- *avec l'équijonction on note :*

$$r2 = \Pi_{\text{Crédit. client, Clientèle.localité}}(\text{Crédit} \mid X \mid \text{Clientèle})$$

où le prédicat de la sélection de l'équijonction est sous-entendu *Crédit.client = Clientèle.client*

Algèbre relationnelle : Équijonction

Crédit

Agence	Prêt	Client	Montant
Downtown	17	Jones	1000
Redwood	23	Smith	2000
Perryridge	15	Hayes	1500
Downtown	14	Jackson	1500
Mianus	93	Curry	500
Round Hill	11	Turner	900
Pownal	29	Williams	1200
North Town	16	Adams	1300
Downtown	18	Johnson	2000
Perryridge	25	Glenn	2500
Brighton	10	Brooks	2200

Clientèle

Client	Rue	Localité
Jones	Main	Harrison
Smith	North	Rye
Hayes	Main	Harrison
Curry	North	Rye
Lindsay	Park	Pittsfield
Turner	Putnam	Stamford
Williams	Nassau	Princeton
Adams	Spring	Pittsfiel
Johnson	Alma	Palo Alto
Glenn	Sand Hill	Woodside
Brooks	Senator	Brooklyn
Green	Walnut	Stamford

Résultat:

Client	Localité
Jones	Harrison
Smith	Rye
Hayes	Harrison
Curry	Rye
Turner	Stamford
Williams	Princeton
Adams	Pittsfield
Johnson	Palo Alto
Glenn	Woodside
Brooks	Brooklyn

Algèbre relationnelle : Quotient relationnel

LE QUOTIENT RELATIONNEL : ¸

- *opérateur binaire*
- *permet de traiter la requête "pour tous les tuples" d'une consultation qui apparaissent dans la relation r en combinant chaque tuple de la relation s*
- *notée : r ¸ s*
- *Un tuple t appartient au quotient r ¸ s, si les valeurs en t apparaissent dans r en combinaison avec chaque tuple de s.*

Algèbre relationnelle : Quotient relationnel

Exemple :

Trouver les NAS des employés travaillant sur tous les projets dans lequel travaille John Smith. (référence p.76)

$$SMITH \leftarrow \sigma_{FNAME=\text{'John'} \text{ AND } LNAME=\text{'Smith'}}(EMPLOYEE)$$

$$SMITH\_PNOS \leftarrow \pi_{PNO}(WORKS\_ON \mid X \mid_{ESSN=SSN} SMITH)$$

Projection de tous les numéros de projets associés à John Smith

SMITH_PNOS

PNO
1
2

Algèbre relationnelle : Quotient relationnel

$SSN\_PNOS \leftarrow P_{PNO,ESSN}(WORKS\_ON)$

SSN_PNOS

ESSN	PNO
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10

...	...
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987987987	30
987987987	20
888665555	20

Projection des numéros de projets et NAS de la table WORKS_ON

$SSNS \leftarrow SSN\_PNOS \div SMITH\_PNOS$

SSNS

SSN
123456789
453453453

Projection des NAS qui travaillent sur tous les mêmes projets que Smith.

Algèbre relationnelle : Quotient relationnel

Démonstration en utilisant les opérateur de base :

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - r)$$

On peut aussi écrire :

$$r(R) \div s(S) = T_1 - T_2$$

posons Y = R - S

Ensemble des attributs de R qui ne sont pas dans S

donc, $T_1 = \Pi_Y(r)$

et donne le NAS de tous les tuples de r (16 tuples)

Algèbre relationnelle : Quotient relationnel

- *Décomposons $T_2 = P_Y((s \; X \; T_1) - r)$*

  $(s \; X \; T_1)$: Ensemble (“ fabriqué artificiellement ”) où l ’on retrouve au moins une occurrence de tous les sous-ensembles de tuples que l’on recherche dans r (32 tuples)

$s \; X \; T_1$

ESSN	PNO
123456789	1
123456789	2
123456789	1
123456789	2
666884444	1
666884444	2
453453453	1
453453453	2
...	...

...	...
333445555	1
333445555	2
333445555	1
333445555	2
...	...
999887777	1
999887777	2
...	...

...	...
888665555	1
888665555	2

Algèbre relationnelle : Quotient relationnel

((s X T_1) - r) = Tous les tuples de (s X T_1) qui ne sont pas dans r.

Π_Y((s X T_1) - r) : Projections de tous les R-S (ESSN) qui ne remplissent pas les conditions.

(s X T_1) - r

ESSN	PNO
666884444	1
666884444	2
333445555	1
999887777	1
999887777	2
888665555	1
888665555	2

P_Y((s X T_1) – r)

ESSN
666884444
666884444
333445555
999887777
999887777
888665555
888665555

Algèbre relationnelle : Quotient relationnel

Résultat final : SSNS ← SSN_PNOS ÷ SMITH_PNOS

T_1 - T_2 : Tous les tuples de T_1 qui ne sont pas dans T_2

T_1

ESSN	…
123456789	333445555
123456789	999887777
666884444	999887777
453453453	987987987
453453453	987987987
333445555	987987987
333445555	987987987
333445555	888665555

T_2

ESSN
666884444
666884444
333445555
999887777
999887777
888665555
888665555

SSNS

ESSN
123456789
453453453

Algèbre relationnelle : opérateur arithmétiques

+ - * /

- *utilisés dans les expressions algébriques de:*
  - Projection
  - Sélection
  - Condition de jointure

- *Utilisés sur les valeurs numériques des relations*

Exemple: Donnez une hausse de 5% sur le salaire des employés.

$$P_{nom,\ salaire*0.5}\ (employé)$$

Algèbre relationnelle : opérateur de chaînes de caractères

LIKE

- *utilisé pour faire une comparaison*
- *paramètres*
  - _ un caractère quelconque à la position déterminée
  - % une chaîne de caractère de longueur quelconque

Exemples: Donnez la liste des employés dont…

- *le nom de famille est Tremblay.*

  $s_{nom\ LIKE\ 'Tremblay'}(Employé)$
- *le nom de famille commence par 'A'.*

  $s_{nom\ LIKE\ 'A\%'}(Employé)$
- *la 2ième lettre du nom de famille est un 'u'.*

  $s_{nom\ LIKE\ '\_u\%'}(Employé)$

Algèbre relationnelle : opérateur de groupement

- *fonction non standard de l'algèbre relationnel*
- *équivalente au «GROUP BY» du langage SQL.*
- *noté :* **<attributs de regroupement>** $\acute{A}$ **<Liste de fonctions>** ***(RELATION)***
  - **<attributs de regroupement>** : attributs appartenant à la relation
  - **<Liste de fonctions>** : fonctions d'agrégation qui permettent de calculer des valeurs à partir des tuples.

  Ces fonctions sont :
  - AVG : Moyenne des tuples
  - COUNT: Nombre de tuples
  - MAX : La valeur maximum
  - MIN : La valeur minimum
  - SUM : La somme des valeurs
- *La relation résultante contient les attributs de regroupement plus un attribut pour chacun des éléments de la liste de fonctions.*

Algèbre relationnelle : opérateur de groupement

Exemple : Pour chaque département, retrouver le numéro de département, le nombre d'employé et le salaire moyen.

$$R(DNO, NB\_EMPLOYEES, AVG\_SALARY) \leftarrow DNO\,\mathfrak{I}_{COUNT\ SSN,\ AVERAGE\ SALARY}\,(EMPLOYE)$$

Attribut de regroupement

Résultat :

R	DNO	NB_EMPLOYEES	AVG_SALARY
	5	4	30000
	4	3	32000
	1	1	60000

Note : Si la liste d'attributs n'est pas spécifiée, les noms d'attributs sont formés par concaténation du nom de la fonction et le nom de l'attribut.

$$DNO\,\mathfrak{I}_{COUNT\ SSN,\ AVERAGE\ SALARY}\,(EMPLOYEE)$$

DNO	COUNT_SSN	AVERAGE_SALARY
5	4	30000
4	3	32000
1	1	60000

Algèbre relationnelle : opérateur de groupement

Exemple (suite) :

Si aucun attribut de regroupement n'est spécifié, les fonctions s'appliquent sur les valeurs d'attributs de tous les tuples à la fois.

La relation résultante n'aura donc qu'un seul tuple.

$$Á_{\text{COUNT SSN, AVERAGE SALARY}} (\text{EMPLOYE})$$

Résultat :

COUNT_SSN	AVERAGE_SALARY
8	40666

Calcul Relationnel

Le Calcul Relationnel …

- *est un langage non procédural*
- *décrit l'information à extraire sans spécifier la séquence de recherche.*

Il existe 2 formes de calcul relationnel :

- sur tuples
- sur domaines

Calcul Relationnel sur tuples

Forme de consultation : { t | P(t) }

- *représente l'ensemble de tous les tuples tels que le prédicat P s'applique à t.*
- *P(t) est aussi appelée formule.*

  Une formule est constituée de variables tuples :

  - liées par opérateurs : " il existe ∃ " **ou** " pour tout ∀ "

    ou

  - libres (non liées)

Ex : $t \in$ Crédit $\cap \exists$ s (t[client] = s [client])

- **t** est une variable libre et **s** est une variable liée.

Rappel: *t[A] : valeur du tuple t sur l'attribut A*

t *Î r : tuple t appartient à la relation r*

Calcul Relationnel sur tuples

- *Une formule du calcul relationnel sur tuples est constituée d'atomes.*
- *Un atome s'identifie sous l'une des formes suivantes :*
  - $s \in r$
    - *où s est un tuple variable et r une relation*
  - $s[x] \Theta u[y]$
    - *où Θ opérateur de comparaison (=, <, >, etc)*
    - *où s et u sont des tuples variables*
    - *x et y attributs de s et u*
  - $s[x] \Theta c$
    - *où s est un tuple variable*
    - *x attribut de s*
    - *C est une constante du domaine relatif à l'attribut x*

Calcul Relationnel sur tuples

Une formule est élaborée à partir des atomes selon les règles suivantes:

- *un atome est une formule*
- *si P est une formule, alors les résultats des opérations unaires sur P sont aussi des formules.*
- *si P et Q sont des formules, alors P È Q et P Ç Q sont aussi des formules.*

Calcul Relationnel sur tuples

Exemple 1 :

Nous avons les relations :

- *Crédit(agence, prêt, client, montant)*
- *Clientèle(client, rue, localité)*

Trouver l'agence, le prêt, le client et le montant des prêts de plus de 1200$ (On recherche les tuples entiers)

$$t \mid t \in \text{Crédit} \cap t[\text{montant}] > 1200$$

Calcul Relationnel sur tuples

Exemple 2 :

Trouver tous les clients qui ont des prêts de plus 1200$

$$t \mid \exists s (s \in \text{Crédit} \cap (t[\text{client}] = s[\text{client}]) \cap (s[\text{montant}] > 1200))$$

S'énonce :

L'ensemble de tous les tuples t tels qu'il existe un tuple s de la relation Crédit pour lequel les valeurs de t et de s sur l'attribut client sont égaux et la valeur de s sur l'attribut montant soit > 1200$.

Calcul Relationnel sur tuples

Exemple 3 : Trouver tous les clients avec leur localité qui ont des prêts à Perryridge.

$$t \mid \exists\, s\, (\, s \in \text{Crédit} \cap (t\,[\text{client}] = s\,[\text{client}])$$

$$\cap\, (\, s\,[\text{agence}] = \text{Perryridge})$$

$$\cap\ \exists\, u\, (\, u \in \text{Clientèle} \cap (u\,[\text{client}] = s\,[\text{client}])$$

$$\cap\, (\, t\,[\text{localité}] = u\,[\text{localité}]\,)$$

S'énonce : L'ensemble de tous les tuples t tels qu'il existe un tuple s de la relation Crédit pour lequel les valeurs de t et de s sur l'attribut client sont égaux et la valeur de s sur l'attribut agence soit Perryridge, et tels qu'il existe un tuple u de la relation Clientèle pour lequel les valeurs de u et de s sur l'attribut client sont égaux et la valeur de t et de u sur l'attribut localité sont égaux.

Calcul Relationnel sur domaines

- *Les variables prennent leurs valeurs sur le domaine d'un attribut plutôt que sur un tuple entier.*
- *Le calcul relationnel sur domaines est relié au calcul relationnel sur tuples.*
- *Une consultation s'exprime sous la forme :*

$$\{ <x_1, x_2, ..., x_n> \mid P(x_1, x_2, ..., x_n) \}$$

où les x_i, 1 <= i <= n représentent les variables d'un domaine et P est une formule composée d'atomes.

Un atome s'identifie sous l'une des formes suivantes :

- $<x_1, x_2, ..., x_n> \in$ r où r est une relation entre n attributs et x_i sont des variables ou des constantes d'un domaine.
- x Θ y, où x et y sont des variables de domaine et Θ est un opérateur de comparaison (=, <, >, etc...)
- x Θ c, où x est une variable du domaine et c est une constante.

NOTE: Les formules sont élaborées à l'aide des mêmes règles que pour le calcul relationnel sur tuples.

Calcul Relationnel sur domaines

Exemples :

Soit les schémas relationnels suivants :

- *Crédit (agence, prêt, client, montant)*
- *Dépôt (agence, compte, client, position)*
- *Clientèle (client, rue, localité)*

1. Trouver l'agence, le prêt, le client et le montant des prêts de plus de 1200 $:

$$\{ \langle a,p,c,m \rangle \mid \langle a,p,c,m \rangle \in \text{Crédit} \cap m > 1200 \}$$

Calcul Relationnel sur domaines

2. Trouver les clients dont le montant des prêts est de plus de 1200 $:

$\{ \langle c \rangle \mid \exists\, a,p,m\ (\langle a,p,c,m \rangle \in \text{Crédit} \cap m > 1200) \}$

3. Trouver les clients qui ont un compte ou un emprunt à l'agence de Perryridge :

$\{ \langle c \rangle \mid \exists\, a,p,m\ (\langle a,p,c,m \rangle \in \text{Crédit} \cap a = \text{Perryridge}) \cup \exists\, a,co,p\ (\langle a,co,c,p \rangle \in \text{Dépôt} \cap a = \text{Perryridge}) \}$