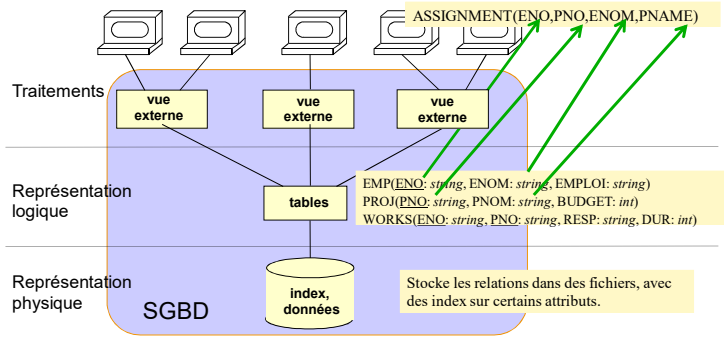
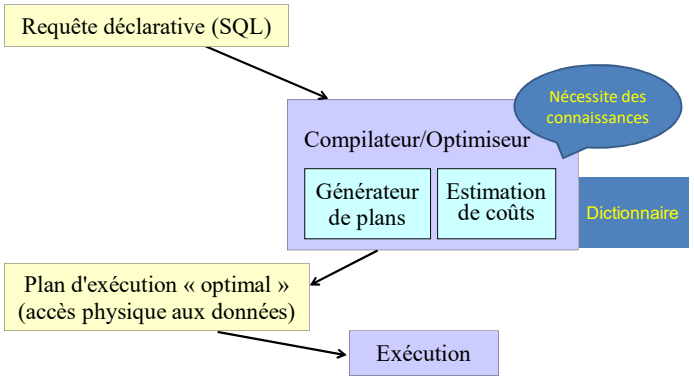
UFR 919 – licence d'informatique – cours LU3IN009 SGBD

Cours 2 : Algèbre relationnelle

Architecture ANSI/SPARC



Traitement de requêtes



Problème général

Comment évaluer une requête SQL efficacement ?

Le nom et le titre des employés qui travaillent dans des projets avec un budget > 250 et plus que 2 employés:

```
SELECT DISTINCT Ename, Title
  FROM Emp, Project, Works
 WHERE Budget > 250
   AND Emp.Eno=Works.Eno
   AND Project.Pno=Works.Pno
   AND Project.Pno IN (SELECT Pno
                 FROM Works
            GROUP BY Pno
              HAVING COUNT(*) > 2)
```

Problème

- Requête SQL = expression *déclarative*
- Plan d'exécution = programme *impératif*
  - Boucles, tests, …
  - Opérations sur des tables et des index : ***algèbre***
  - Génération de tables et index temporaires
- Problème : Comment trouver un plan d'exécution
  - Correct : il fait ce que la requête "dit"
  - **Efficace : il le fait vite!**

  Problème complexe car de nombreux plans possibles (encore plus si la BD est répartie)

Évaluation et optimisation de requêtes

- Traitement de requêtes SQL
- Algèbre relationnelle
- Optimisation de requêtes

Langages d'Interrogation Relationnels

- SQL :
  - langage « pratique » pour la programmation
- Calcul relationnel :
  - formules logiques qui décrivent le « sens » formel d'une requête SQL (sauf group by, order).
- Algèbre :
  - composition d'opérations qui décrit une exécution possible d'une requête SQL / calcul
  - Expression relationnelle (ex. la relation R) : retourne un ensemble de n-uplets (ex. le contenu de R)
  - « algèbre » : on reste dans l'espace des expr. rel.

Algèbres relationnelles

- Algèbre ensembliste : opérateurs sur des *ensembles* de n-uplets (relations)
- Algèbre physique : opérateurs *implantés* dans un SGBD
- Pour un opérateur logique il existe généralement plusieurs opérateurs physiques qui l'implantent (choix d'opérateurs)
- Certains opérateurs physiques n'ont pas d'opérateur équivalent au niveau des ensembles (tri, group-by, ...) ou alors il faut considérer une *algèbre étendue*
  - Algèbre ⇔ calcul algèbre étendue ⇔ SQL

Algèbre relationnelle étendue

- Inclus opérateurs pour group by, select avec doublon et order by
- Group by : opérateurs complexe à modéliser, surtout si clause having
- Select avec doublons : nécessite de changer de modèle (on ne travaille plus sur des ensembles, mais sur des « ensembles avec doublon »)
- Order by : nécessite de changer le modèle (on ne travaille plus sur des ensembles de n-uplets, mais sur des listes ordonnées)

⇒ Pour simplifier on étudiera principalement des requêtes sans group by, sans agrégats,sans doublons et sans order by

Algèbre relationnelle (ensembliste)

Opérateurs unaires :

<Opérateur>$_{<parametres>}$*<Opérande>* → *<Résultat>*

Opérateurs binaires :

<Opérande> <Opérateur>$_{<parametres>}$*<Opérande>* → *<Résultat>*

Langage *fermé* : les opérandes et les résultats sont *toujours* des relations (ensembles de n-uplets) → composition d'opérations

Opérateurs de l'algèbre relationnelle

Opérateurs de base :

- sélection
- projection
- produit cartésien
- opérations ensemblistes: union, différence
- *renommage*

Opérateurs dérivés :

- intersection
- jointure
- division

Sélection

Sélection d'un sous-ensemble de la relation opérande :

$$\sigma_F(R)$$

- *R* est une expression relationnelle
- *F* est une *formule logique sans quantificateur* composée de
- opérandes: constantes et attributs
- opérateurs de comparaison :<, >, =, ≠, ≤, ≥
- opérateurs logiques : ∧, ∨ , ¬

Résultat : *sous-ensemble* des n-uplets de *R* qui satisfont la formule F

En SQL ?

Exemple de sélection

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

$\sigma_{\text{TITLE='Elect. Eng.'}}$(EMP)

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E6	L. Chu	Elect. Eng.

Projection

Projection sur un ensemble d'attributs d'une relation

$$\pi_{A_1,\ldots,A_n}(R)$$

- R est une expr. relationnelle
- $\{A_1,\ldots, A_n\}$ est un sous-ensemble des attributs de R

Résultat : ensemble de n-uplets de R sans les attributs (colonnes) qui ne se trouvent pas dans $\{A_1,\ldots, A_n\}$

En SQL ?

Exemple de projection

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	*CAD/CAM*	250000
P4	Maintenance	310000
P5	*CAD/CAM*	500000

$\Pi_{\text{PNO,BUDGET}}$(PROJ)

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000
P5	500000

Π_{PNAME}(PROJ)

PNAME
Instrumentation
Database Develop.
CAD/CAM
Maintenance

Deux sémantiques :
- Ensembliste : élimination des *n-uplets doublons*
- SQL : avec doublons => distinct.

Pourquoi ?

Produit cartésien

Produit cartésien entre deux tables :

$$R \times S$$

- R est une table de degré k_1 , cardinalité n_1
- S est une table de degré k_2 , cardinalité n_2

Résultat : relation de *degré* $(k_1 + k_2)$ et contient $(n_1 * n_2)$ n-uplets, où chaque n-uplet est la *concaténation* d'un n-uplet de R avec un n-uplet de S.

En SQL
Select … from R, S
where <pas de lien entre R et S>
(« Monsieur c'est bloqué ! ») ?

Exemple de produit cartésien

Clé ?

EMP

8 n-uplets

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

PAY

4 n-uplets

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	45000
Programmer	60000

EMP × PAY

32 n-uplets

ENO	ENAME	EMP.TITLE	PAY.TITLE	SALARY
E1	J. Doe	Elect. Eng.	Elect. Eng.	55000
E1	J. Doe	Elect. Eng.	Syst. Anal.	70000
E1	J. Doe	Elect. Eng.	Mech. Eng.	45000
E1	J. Doe	Elect. Eng.	Programmer	60000
E2	M. Smith	Syst. Anal.	Elect. Eng.	55000
E2	M. Smith	Syst. Anal.	Syst. Anal.	70000
E2	M. Smith	Syst. Anal.	Mech. Eng.	45000
E2	M. Smith	Syst. Anal.	Programmer	60000
E3	A. Lee	Mech. Eng.	Elect. Eng.	55000
E3	A. Lee	Mech. Eng.	Syst. Anal.	70000
E3	A. Lee	Mech. Eng.	Mech. Eng.	45000
E3	A. Lee	Mech. Eng.	Programmer	60000
E8	J. Jones	Syst. Anal.	Elect. Eng.	55000
E8	J. Jones	Syst. Anal.	Syst. Anal.	70000
E8	J. Jones	Syst. Anal.	Mech. Eng.	45000
E8	J. Jones	Syst. Anal.	Programmer	60000

Union

Union *ensembliste* entre deux tables :

$R \cup S$

- R et S sont des relations *compatibles pour l'union* (même arité et domaines d'attributs)

Résultat : n-uplets qui sont dans R ou dans S

PAY1

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	*45000*

∪

PAY2

TITLE	SALARY
Syst. Anal.	73100
Mech. Eng.	*45000*
Programmer	60000

=

PAY1 ∪ PAY2

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	*45000*
Programmer	60000
Syst. Anal.	73100

Différence

Différence *ensembliste* entre deux tables :

$R - S$

- R et S sont des relations *compatibles* pour l'union.

Résultat : n-uplets qui sont dans R, mais pas dans S

PAY1

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	*45000*

–

PAY2

TITLE	SALARY
Syst. Anal.	73100
Mech. Eng.	*45000*
Programmer	60000

=

PAY1 – PAY2

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000

Renommage

Renommage de plusieurs attributs d'une table :

$$\rho_{A_1,\ldots,A_n \rightarrow B_1,\ldots,B_n}(R)$$

R est une relation

$\{A_1,\ldots, A_n\}$ est un sous-ensemble des attributs de R

$\{B_1,\ldots, B_n\}$ est un ensemble d'attributs

Résultat : une relation avec les mêmes n-uplets (le même contenu) où chaque attribut A_i a été renommé en B_i

On note aussi $R_{A1 \rightarrow B1,\ A2 \rightarrow B2\ldots}$

On peut aussi renommer la relation R en S, noté R_S

Exemple de renommage

- EMP(ENO, TITLE, ENAME)
- $\rho_{TITLE,ENAME \rightarrow JOB,\ NOM}(EMP)$ change le schéma de la relation EMP en EMP (ENO, NOM, JOB)
- ne change rien au contenu

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

$\rho_{TITLE,ENAME \rightarrow JOB,\ NOM}$ $\longrightarrow$

EMP

ENO	NOM	JOB
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

Opérateurs de l'algèbre relationnelle

Opérateurs de base :
- sélection
- projection
- produit cartésien
- opérations ensemblistes: union, différence
- *renommage*

Opérateurs dérivés :
- intersection
- jointure
- division

Intersection

Intersection de deux tables:

$$R \cap S = R - (R - S)$$

R, S sont deux tables compatibles (attributs deux à deux de même domaine)

Résultat : ensemble de n-uplets qui se trouvent à la fois dans R *et* dans S

PAY1

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	*45000*

$\cap$

PAY2

TITLE	SALARY
Syst. Anal.	73100
Mech. Eng.	*45000*
Programmer	60000

=

PAY1 $\cap$ PAY2

TITLE	SALARY
Mech. Eng.	*45000*

Jointure

Jointure entre deux tables R et S :

$$R \bowtie_F S = \sigma_F(R \times S)$$

R et S sont des relations (sans attributs en commun)

F est une formule logique composée d'au moins un atome de la forme $A_i\ \theta\ B_j$ où

$\theta \in \{<, >, =, \neq, \leq, \geq\}$, A_i est un attribut de R, B_j est un attribut de S

Résultat : sous-ensemble des n-uplets dans le produit cartésien *R* $\times$ *S qui satisfont la formule F*

Types de jointure

θ-jointure (théta-jointure)

la formule F utilise les comparaisons <, >, ≠, ≤, ≥

Equi-jointure

la formule F n'utilise que l'égalité : =

$R \bowtie_{R.A=S.B} S$

Jointure naturelle : R(X,Y), S(X, Y')

Equi-jointure où on élimine les attributs en communs

$R \bowtie S = \Pi_{R.X,\ R.Y,\ S.Y'}\, \sigma_F (R \times S) = \Pi_{S.X,\ R.Y,\ S.Y'}\, \sigma_F (R \times S)$

la condition de jointure F est R.X = S.X (X représente tous les attributs en commun entre R et S)

Exemple de jointure naturelle

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

PAY

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	45000
Programmer	60000

EMP ⋈ PAY

Clé ?

ENO	ENAME	TITLE	SALARY
E1	J. Doe	Elect. Eng.	55000
E2	M. Smith	Analyst	70000
E3	A. Lee	Mech. Eng.	45000
E4	J. Miller	Programmer	60000
E5	B. Casey	Syst. Anal.	70000
E6	L. Chu	Elect. Eng.	55000
E7	R. Davis	Mech. Eng.	45000
E8	J. Jones	Syst. Anal.	70000

Exemple de θ–jointure

EMP

ENO	ENAME	TITLE	CONTR
E1	J. Doe	Elect. Eng	12
E2	M. Smith	Syst. Anal.	12
E3	A. Lee	Mech. Eng.	12
E4	J. Miller	Programmer	24
E5	B. Casey	Syst. Anal.	24
E6	L. Chu	Elect. Eng.	36
E7	R. Davis	Mech. Eng.	36
E8	J. Jones	Syst. Anal.	12

WORKS

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E7	P5	Engineer	23
E8	P3	Manager	40

EMP ⋈$_{\text{EMP.ENO=WORKS.ENO} \wedge \text{CONTR<DUR}}$ WORKS

EMP. ENO.	ENAME	TITLE	WORKS. ENO	PNO	RESP	DUR	CONTR
E2	M. Smith	Syst. Anal.	E2	P1	Manager	24	12
E3	A. Lee	Mech. Eng.	E3	P4	Engineer	48	12
E6	L. Chu	Elect. Eng.	E6	P4	Manager	48	36
E8	J. Jones	Syst. Anal.	E8	P3	Manager	40	12

Résultat (en français) ?

Contrats et leurs employés embauchés sur une durée inférieure à la durée de leur participation au projet.

Division

Soient les relations R et S tels que *le schéma de R contient tous les attributs de S*

$R(A_1,\ldots,A_k,A_{k+1},\ldots,A_{k+n})$ est de degré $k+n$ et

$S(A_1,\ldots,A_k)$ de degré k.

La division de R par S :

$$T(A_{k+1},\ldots,A_{k+n}) = R \div S$$

est la « plus grande » relation de degré *n telle que*

$$T \times S \subseteq R.$$

$R \div S$ contient les tuples de schéma $(A_{k+1},\ldots,A_{k+n})$ qui sont associés, dans R, à tous les tuples de S.

Exemple de division

EMP

ENO	PNO	PNAME	BUDGET
E1	P1	Instrumentation	150000
E2	P1	Instrumentation	150000
E2	P2	Database Develop.	135000
E3	P1	Instrumentation	150000
E3	P4	Maintenance	310000
E4	P2	Instrumentation	150000
E5	P2	Instrumentation	150000
E6	P4	Maintenance	310000
E7	P3	CAD/CAM	250000
E8	P3	CAD/CAM	250000
E3	P2	Database Develop.	135000
E3	P3	CAD/CAM	250000

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

EMP ÷ PROJ

ENO
E3

Expression de la division

R

X	Y
x1	y1
x2	y1
x3	y1
x4	y1
x1	y2
x3	y2
x2	y3
x3	y3
x4	y3
x1	y4
x2	y4
x3	y4

S

X
x1
x2
x3

T

Y
y1
y4

Exprimons T en fonction de R et S…

Remarques sur l'algèbre non étendue

Permet de retrouver toutes les valeurs contenues dans la BD (n-uplets, n-uplets tronqués, n-uplets concaténés)

Et rien d'autre…

Certains opérateurs sont commutatifs ($R \bowtie \sigma_F(S)) = \sigma_F(R \bowtie S)$)

D'autres ne le sont pas $\Pi_X(R-S) != \Pi_X(R) - \Pi_X(S)$

Requêtes algébriques

Emp (Eno, Ename, Title, City)
Pay(Title, Salary)
Project(Pno, Pname, Budget, City)
Works(Eno, Pno, Resp, Dur)

Villes où il y a des employés ou des projets?

- $\Pi_{City}(Emp) \cup \Pi_{City}(Project)$

Villes où il y a des projets mais pas d'employés?

- $\Pi_{City}(Project) - \Pi_{City}(Emp)$

Requêtes algébriques

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, City)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

Noms des projets de budget > 225?

- $\Pi_{Pname}(\sigma_{Budget>225}(Project))$

Noms et budgets des projets où travaille l'employé E1?

- $\Pi_{Pname, Budget}(Project \bowtie (\sigma_{Eno='E1'}(Works)))$
- $\Pi_{Pname, Budget}(\sigma_{Project.Pno=Works.Pno}(Project \times \sigma_{Eno='E1'}(Works)))$

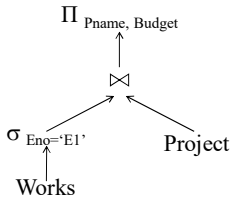
Employés qui travaillent dans chaque projet?

- $\Pi_{Eno, Pno}(Works) \div \Pi_{Pno}(Project)$

Arbre algébrique

Comme toute expression algébrique, peut se représenter sous forme d'arbre.
Utile pour manipuler les requêtes (optimisation, vues)
Exemple

$\Pi_{Pname, Budget}(Project \bowtie \sigma_{Eno='E1'}(Works)))$



Conclusion : algèbre relationnelle

- L'algèbre relationnelle définit un ensemble d'opérations pour interroger une BD relationnelle
- Les opérations peuvent être composées pour former des requêtes complexes
  - Avantage : facilite l'implantation d'un moteur de requêtes
  - Inconvénient : sémantique "opérationnelle"

L'algèbre « cache » la sémantique formelle (ensembliste) du modèle relationnel → calcul relationnel

Calcul de n-uplets et algèbre

- **Théorème:** Le calcul relationnel sûr et l'algèbre relationnelle ont une *puissance d'expression équivalente* (complétude relationnelle)
- *Autrement* : toutes les requêtes qu'on peut exprimer en utilisant l'algèbre relationnelle (sélection, projection, jointure, ...) peuvent être exprimées dans le calcul relationnel sûr et vice-versa.

Traduction: sélection, projection, jointure

Deux tables : R(A,B,C) S(C,D)

- $\Pi_{A,B}$ (R) $\equiv$ { t.A, t.B | R(t) }
- $\Pi_B(\sigma_{A=3}(R))$ $\equiv$ { t.B | R(t) $\wedge$ t.A = 3 }
- R $\bowtie$ S $\equiv$ { t.A, t.B, t.C, u.D | R(t) $\wedge$ S(u) $\wedge$ t.C = u.C }
- Π_C (R) - Π_C (S) $\equiv$ { t.C | R(t) $\wedge$ $\neg\exists$u (S(u) $\wedge$ t.C = u.C)

Traduction de la division

R(A,B,C,D) $\div$ S(C,D) s'exprime par la requête suivante :

R$\div$S = { x.A, x.B | R(x) $\wedge$
$\forall$u (S(u) $\rightarrow$ $\exists$v (R(v) $\wedge$
v.A = x.A $\wedge$ v.B = x.B $\wedge$
v.C = u.C $\wedge$ v.D = u.D)) }

Remarque : F $\rightarrow$ G est équivalent à $\neg$ F $\vee$ G

Requêtes algébriques (suite)

Emp (Eno, Ename, Title, City) **Project**(Pno, Pname, Budget, City)
Pay(Title, Salary) **Works**(Eno, Pno, Resp, Dur)

1. Projets ayant au moins deux employés?
2. Projets ayant exactement deux employés ?
3. Couples d'employé (même nom, même ville) ?
4. Quel grade (title) est le mieux payé ?
5. Quels sont les projets où tous les grades sont représentés ?
6. Quels employés n'habitent pas la(es) ville(s) où ils travaillent ?

Requêtes algébriques (TME)

- **Sponsorise(NSp, NJo, Somme),**
- **Joueur(NJo, Eq, Taille, Age),**
- **Equipe(NEq, Ville, Couleur, StP)**
- **Match(Eq1, Eq2, Date, St),**
- **Distance(St1, St2, NbKm)**

1. Quelles équipes ont déjà joué au stade préféré de l'équipe des Piépla ?
2. Quels sont les joueurs qui ne sont pas sponsorisés par Adadis ?
3. Quel est le(s) plus grand(s) joueur(s) sponsorisé par Adadis ?
4. A quelle date a eu lieu un match entre deux équipes sponsorisées par le même sponsor ?