Bases de Données Avancées, Partie 1 : Algèbre Relationnelle

Thomas Gerald

September 8, 2025

Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, CNRS
thomas.gerald@lisn.upsaclay.fr

C'est quoi ?

L'algèbre relationnelle (AR) est un langage formel pour la construction de requêtes. Les requêtes sont construites à partir d'une collection d'opérateurs.

- Chaque requête accepte comme arguments une **collection d'instances de relations**
- Une requête retourne **une instance de relation**

Les requêtes relationnelle décrivent une procédure pour la calcul de la réponse basée sur l'ordre des opérateurs dans la requête !!!

Objectifs

- Connaître les opérateurs de l'AR et leur résultats
- Être capable de retranscrire en langage naturel une expression en AR et vice-versa
- Être capable de transcrire en SQL une expression en AR et vice-versa
- Étant données deux requêtes définies à l'aide de AR, établir leur équivalence
- Modéliser sous forme d'arbre un expression AR

Opérateurs

- σ : La sélection (filtrage)
- Π : La projection (sélection des colonnes)
- $\cup$: L'union
- $\cap$: L'intersection
- $-$: La Différence
- $\times$: Le produit cartésien
- $\bowtie$: La jointure

Algèbre Relationnelle: Exemple du cours

- Etu(<u>*eid*: **`serial`**</u>, *nom*: **`str`**, *prenom*: **`str`**, *mail*: **`str`**)
- Cours(<u>*cid*: **`serial`**</u>, *titre*: **`str`**)
- Eval(<u>*nid*: **`serial`**</u>, *etu_id*: **`int`**, *cours_id*: **`int`**, *note*: **`float`**)

Etu

eid	nom	prenom	mail
1	Zeblouse	Agathe	az@*****
2	Huai	Odile	oh@*****
3	Peuplu	Jean	jp@*****
4	Hochon	Paul	hp@*****
5	Gator	Ali	ag@*****

Cours

cid	titre
1	Bases de données avancées
2	Mathématiques
3	Anglais

Eval

nid	etu_id	cours_id	note
1	1	2	0
4	3	2	15
5	3	3	6

Une relation :

On notera R une instance d'une relation, on appelle :

- Ses attributs (les colonnes) : $a_1, a_2 \dots a_n$
- Ses enregistrements : $\{e_1, e_2, \dots, e_m\}$

La relation R est définie comme un ensemble d'enregistrements où

$$\begin{aligned} R(a_1, a_2, \dots a_m) &= \{e_1, \qquad e_2, \qquad \dots, e_m \qquad \} \\ &= \{(e_{11}, e_{12}, \dots e_{1n}), (e_{21}, e_{22}, \dots e_{2n}), \dots, (e_{m1}, e_{m2}, \dots e_{mn})\} \end{aligned}$$

avec e_{ij} appartenant au domaine définit pour l'attribut (colonne) a_j. On notera $domaine(a_j)$ le domaine de a_j (flotant, entier, chaine de caractères).

Les enregistrements sont distincts en algèbre relationnelle $e_i, e_j \in R$ et $i \neq j \Rightarrow e_i \neq e_j$

Une requête, une fonction ?

Soit q une requête et $(R_1, R_2, \ldots R_n) \in \mathcal{A}^n$ un tuple d'instance de relations, l'application de la requête q sur le tuple R_i :

$$\begin{aligned} q\colon \mathcal{A}^n &\longrightarrow \mathcal{A} \\ (R_1, R_2, \ldots, R_n) &\longmapsto T \end{aligned}$$

$\rightarrow T$ est ici une relation issue de la requête appliqué sur le tuple de relations !!!

Les opérateurs unaires

On travail avec des opérateurs ne prenant en compte qu'une unique relation et non un tuple de relations !

- La sélection σ : Filtrage des enregistrements
- La projection Π : Sélection des attributs

La sélection :

Filtrage des résultats d'une relation selon une condition, on la note

$$\sigma_{condition}(R)$$

Exemple :

$\sigma_{note \geq 6}(Eval)$ (les notes supérieures à 6 dans la table Eval)

Eval			
nid	etu_id	cours_id	note
1	1	2	0
4	3	2	15
5	3	3	6

$\rightarrow$

T			
nid	etu_id	cours_id	note
4	3	2	15
5	3	3	6

La sélection :

Soit $R(a_1, a_2, \dots a_m)$ une relation, on appelle sélection l'opération :

$$\begin{aligned} \sigma_{condition} : \mathcal{A} &\longrightarrow \mathcal{A} \\ R &\longmapsto \{e_i \in R | condition(e_i)\} \end{aligned}$$

En d'autres termes, la relation retournée est la relation T composée des enregistrements ou la condition est vérifiée !!

Algèbre Relationnelle: La sélection (plus formellement)

La sélection :

Soit $R(a_1, a_2, \ldots a_m)$ une relation, on appelle sélection l'opération :

$$\begin{aligned} \sigma_{condition} : \mathcal{A} &\longrightarrow \mathcal{A} \\ R &\longmapsto \{e_i \in R | condition(e_i)\} \end{aligned}$$

En d'autres termes, la relation retournée est la relation T composée des enregistrements ou la condition est vérifiée !!

Exemple :

On considère la condition $(a_1 = 2) \wedge \neg(a_3 = "Jean")$:

$$\sigma_{(a_1=2)\wedge\neg(a_3="Jean")}(R) = \{e_i | e_i \in R \text{ tq } (e_{i1} = 2) \wedge \neg(e_{i3} = "Jean")\}$$

$\rightarrow$ les enregistrements de R où la première colonne est égale à 2 et où la troisième colonne n'est pas "JEAN"

La projection :

Sélection d'un sous ensemble d'attributs pour une relation :

$$\Pi_{a_i,a_j,\ldots,a_k}(R)$$

Exemple :

$\Pi_{nom,prenom}(etu)$ (les noms et prénoms de la table R)

Etu			
eid	nom	prenom	mail
1	Zeblouse	Agathe	az@*****
2	Huai	Odile	oh@*****
3	Peuplu	Jean	jp@*****
4	Hochon	Paul	hp@*****
5	Gator	Ali	ag@*****

$\rightarrow$

T	
nom	prenom
Zeblouse	Agathe
Huai	Odile
Peuplu	Jean
Hochon	Paul
Gator	Ali

La projection :

Soit $R(a_1, a_2, \dots a_m)$ une relation, on appelle projection l'opération :

$$\begin{aligned} \Pi_{a_j, a_k, \dots, a_l} : \mathcal{A} &\longrightarrow \mathcal{A} \\ R &\longmapsto \{(e_{ij}, e_{ik}, \dots, e_{e_l}) | e_i \in R\} \end{aligned}$$

En d'autres termes, la relation retournée est la relation T composée de tous les enregistrements de R sur les attributs définies!!

- Etu(*eid*: **`serial`**, *nom*: **`str`**, *prenom*: **`str`**, *mail*: **`str`**)
- Cours(*cid*: **`serial`**, *titre*: **`str`**)
- Eval(*nid*: **`serial`**, *etu_id*: **`int`**, *cours_id*: **`int`**, *note*: **`float`**)

Pour les exercices suivants trouver la requête AR associée

Question 1

Sélectionnez les étudiants **Ali Gator** et **Paul Hochon**

Question 2

Trouver une requête **AR** équivalente à la requête SQL suivante :

```
SELECT nom, prenom, mail FROM etu
```

Question 3

Sélectionnez les identifiants des étudiants où la note est supérieure à 10

Question 1

Sélectionnez les étudiants Ali Gator et Paul Hochon

Question 1

Sélectionnez les étudiants **Ali Gator** et **Paul Hochon**

$\rightarrow \sigma_{(nom="Gator" \wedge prenom="Ali")\vee(nom="Hochon" \wedge prenom="Paul")}(etu)$

Question 1

Sélectionnez les étudiants Ali Gator et Paul Hochon

$\rightarrow \sigma_{(nom=''Gator''\wedge prenom=''Ali'')\vee(nom=''Hochon''\wedge prenom=''Paul'')}(etu)$

Question 2

Trouver une requête AR équivalente à la requête SQL suivante :

Question 1

Sélectionnez les étudiants **Ali Gator** et **Paul Hochon**

$\rightarrow \sigma_{(nom="Gator" \wedge prenom="Ali") \vee (nom="Hochon" \wedge prenom="Paul")}(etu)$

Question 2

Trouver une requête **AR** équivalente à la requête SQL suivante :

```
SELECT nom, prenom, mail FROM etu
```

Question 1

Sélectionnez les étudiants **Ali Gator** et **Paul Hochon**

$\rightarrow \sigma_{(nom="Gator" \wedge prenom="Ali") \vee (nom="Hochon" \wedge prenom="Paul")}(etu)$

Question 2

Trouver une requête **AR** équivalente à la requête SQL suivante :

```
SELECT nom, prenom, mail FROM etu
```

$\rightarrow \Pi_{nom,prenom,mail}(etu)$

Question 1

Sélectionnez les étudiants **Ali Gator** et **Paul Hochon**

$\rightarrow \sigma_{(nom="Gator" \wedge prenom="Ali") \vee (nom="Hochon" \wedge prenom="Paul")}(etu)$

Question 2

Trouver une requête **AR** équivalente à la requête SQL suivante :

```sql
SELECT nom, prenom, mail FROM etu
```

$\rightarrow \Pi_{nom,prenom,mail}(etu)$

Question 3

Sélectionnez les identifiants des étudiants où la note est supérieure ou égale à 10

Question 1

Sélectionnez les étudiants **Ali Gator** et **Paul Hochon**

$\rightarrow \sigma_{(nom="Gator" \wedge prenom="Ali") \vee (nom="Hochon" \wedge prenom="Paul")}(etu)$

Question 2

Trouver une requête **AR** équivalente à la requête SQL suivante :

```sql
SELECT nom, prenom, mail FROM etu
```

$\rightarrow \Pi_{nom,prenom,mail}(etu)$

Question 3

Sélectionnez les identifiants des étudiants où la note est supérieure ou égale à 10

$\rightarrow \Pi_{etu\_id}(\sigma_{note \geq 10}(note))$

Composition des opérateurs

Les opérateurs se composes !!!

Composition des opérateurs

Les opérateurs se composes !!!

- On peut écrire $\sigma_{condition}(\Pi_{a_i}(R))$

Composition des opérateurs

Les opérateurs se composes !!!

- On peut écrire $\sigma_{condition}(\Pi_{a_i}(R))$
- Ou bien plusieurs sélections $\sigma_{condition_1}(\sigma_{condition_2}(R))$

Composition des opérateurs

Les opérateurs se composes !!!

- On peut écrire $\sigma_{condition}(\Pi_{a_i}(R))$
- Ou bien plusieurs sélections $\sigma_{condition_1}(\sigma_{condition_2}(R))$

Importance de l'ordre des compositions ?

La requête AR suivante est-elle possible sur $R(a_1, a_2, a_3, a_4)$?

$$\sigma_{a_4>100}\Big(\Pi_{a_1,a_2,a_3}(\sigma(a_2 = 5)(R))\Big)$$

Importance de l'ordre des compositions ?

La requête AR suivante est-elle possible sur $R(a_1, a_2, a_3, a_4)$?

$$\sigma_{a_4>100}\left(\Pi_{a_1,a_2,a_3}(\sigma(a_2 = 5)(R))\right)$$

⚠ $\Pi_{a_1,a_2,a_3}(\sigma(a_2 = 5)(R))$ donne une relation $R'(a_1, a_2, a_3)$

Importance de l'ordre des compositions ?

La requête AR suivante est-elle possible sur $R(a_1, a_2, a_3, a_4)$?

$$\sigma_{a_4>100}\left(\Pi_{a_1,a_2,a_3}(\sigma(a_2 = 5)(R))\right)$$

⚠ $\Pi_{a_1,a_2,a_3}(\sigma(a_2 = 5)(R))$ donne une relation $R'(a_1, a_2, a_3)$
$\rightarrow \sigma_{a_4>100}(R')$ ne dispose pas de l'attribut a_4

Importance de l'ordre des compositions ?

La requête AR suivante est-elle possible sur $R(a_1, a_2, a_3, a_4)$?

$$\sigma_{a_4>100}\Big(\Pi_{a_1,a_2,a_3}(\sigma(a_2 = 5)(R))\Big)$$

⚠ $\Pi_{a_1,a_2,a_3}(\sigma(a_2 = 5)(R))$ **donne une relation** $R'(a_1, a_2, a_3)$
$\rightarrow \sigma_{a_4>100}(R')$ **ne dispose pas de l'attribut** a_4

Les opérations ne sont pas forcément commutatives entre elles

L'opérateur ρ

Cette opération permet de nommer une relation à partir d'une expression. Elle prend deux arguments $\rho(A, B)$

- A étant le nom de la relation avec une liste des attributs
- B étant une expression d'algèbre relationnelle

L'opérateur ρ

Cette opération permet de nommer une relation à partir d'une expression. Elle prend deux arguments $\rho(A, B)$

- A étant le nom de la relation avec une liste des attributs
- B étant une expression d'algèbre relationnelle

Exemple

On veut nommer R la relation contenant les attributs pid, eid pour les notes supérieures à 10 de la table `eval` :

L'opérateur ρ

Cette opération permet de nommer une relation à partir d'une expression. Elle prend deux arguments $\rho(A, B)$

- A étant le nom de la relation avec une liste des attributs
- B étant une expression d'algèbre relationnelle

Exemple

On veut nommer R la relation contenant les attributs pid, eid pour les notes supérieures à 10 de la table `eval` :

$$\rho(R(1 \rightarrow pid, 2 \rightarrow eid), \sigma_{note>10}(eval))$$

L'intersection

Comme vue dans le cours de SQL il s'agit de créer une relation des enregistrements communs de deux autres relations, soit deux relation $R(a_1, a_2, \ldots, a_n)$ et $S(a_1, a_2, \ldots, a_n)$ elle s'écrit:

$$R \cap S$$

⚠ Les attributs de chacune des tables doivent être de même domaine

L'intersection

Comme vue dans le cours de SQL il s'agit de créer une relation des enregistrements communs de deux autres relations, soit deux relation $R(a_1, a_2, \ldots, a_n)$ et $S(a_1, a_2, \ldots, a_n)$ elle s'écrit:

$$R \cap S$$

⚠ Les attributs de chacune des tables doivent être de même domaine

Exemple :

$R = \Pi_{etu_id}(\sigma_{note<5}(Eval))$, $S = \Pi_{etu_id}(\sigma_{note>15}(Eval))$

$$R \cap S$$

L'intersection

Comme vue dans le cours de SQL il s'agit de créer une relation des enregistrements communs de deux autres relations, soit deux relation $R(a_1, a_2, \ldots, a_n)$ et $S(a_1, a_2, \ldots, a_n)$ elle s'écrit:

$$R \cap S$$

⚠ Les attributs de chacune des tables doivent être de même domaine

Exemple :

$R = \Pi_{etu_id}(\sigma_{note<5}(Eval)), S = \Pi_{etu_id}(\sigma_{note>15}(Eval))$

$$R \cap S$$

$\rightarrow$ Les étudiants ayant au moins une note supérieure à 15 et une note inférieure à 5

L'intersection :

Soit $R(a_1, a_2, \ldots, a_n)$ et $S(a_1, a_2, \ldots, a_n)$ deux relations de même schéma on appelle l'intersection l'opération :

$$\begin{aligned} R \cap S \colon \mathcal{A}^2 &\longrightarrow \mathcal{A} \\ (R, S) &\longmapsto \{e_i | e_i \in R, e_i \in S\} \end{aligned}$$

L'union :

Soit $R(a_1, a_2, \ldots, a_n)$ et $S(a_1, a_2, \ldots, a_n)$ deux relations de même schéma on appelle l'union l'opération :

$$\begin{aligned} R \cup S \colon \mathcal{A}^2 &\longrightarrow \mathcal{A} \\ (R, S) &\longmapsto \{e_i | e_i \in R \vee e_i \in S\} \end{aligned}$$

L'union :

Soit $R(a_1, a_2, \ldots, a_n)$ et $S(a_1, a_2, \ldots, a_n)$ deux relations de même schéma on appelle l'union l'opération :

$$\begin{aligned} R \cup S \colon \mathcal{A}^2 &\longrightarrow \mathcal{A} \\ (R, S) &\longmapsto \{e_i | e_i \in R \vee e_i \in S\} \end{aligned}$$

La différence :

Soit $R(a_1, a_2, \ldots, a_n)$ et $S(a_1, a_2, \ldots, a_n)$ deux relations de même schéma on appelle la différence l'opération :

$$\begin{aligned} R - S \colon \mathcal{A}^2 &\longrightarrow \mathcal{A} \\ (R, S) &\longmapsto \{e_i | e_i \in R \wedge e_i \notin S\} \end{aligned}$$

Produit Cartésien :

Soit $R(a_1, a_2, \ldots, a_n)$ et $S(b_1, b_2, \ldots, b_m)$ deux relations de schéma différents on appelle le produit cartésien l'opération :

$$\begin{aligned} R \times S \colon \mathcal{A}^2 &\longrightarrow \mathcal{A} \\ (R, S) &\longmapsto \{(e^R_{i1}, \ldots, e^R_{in}, e^S_{j1}, \ldots, e^S_{jn}) | e_i \in R, e_j \in S\} \end{aligned}$$

La relation résultante comporte les attributs $(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m)$

Le schéma des relations peut être différent

Algèbre Relationnelle: θ-jointure

θ-jointure :

Soit $R(a_1, a_2, \ldots, a_n)$ et $S(b_1, b_2, \ldots, b_m)$ deux relations de schéma différents on appelle le θ-jointure l'opération :

$$\begin{aligned} R \bowtie_{a_k \theta b_l} S \colon \mathcal{A}^2 &\longrightarrow \mathcal{A} \\ (R, S) &\longmapsto \{e_i \in E \times S | e_i.a_k \theta e_i.b_l\} \end{aligned}$$

La relation résultante comporte les attributs $(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m)$

Le schéma des relations peut être différent

Un exemple en SQL

```
SELECT * FROM etu, eval WHERE etu.eid > eval.etu_id
```

Équi-jointure:

Soit $R(a_1, a_2, \ldots, a_n)$ et $S(b_1, b_2, \ldots, b_m)$ deux relations de schéma différents on appelle équi-jointure l'opération :

$$\begin{aligned} R \bowtie_{a_k=b_l} S \colon \mathcal{A}^2 &\longrightarrow \mathcal{A} \\ (R, S) &\longmapsto \{e_i \in E \times S | e_i.a_k = e_i.b_l\} \end{aligned}$$

La relation résultante comporte les attributs $(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m)$

$\rightarrow$ C'est une theta jointure sur une égalité !!!

Un exemple en SQL

```
SELECT * FROM etu, eval WHERE etu.eid = eval.etu_id
```

Équi-jointure:

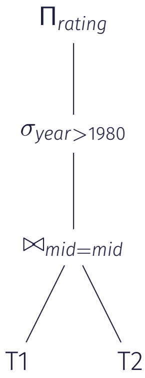
Soit $R(a_1, a_2, \ldots, a_n)$ et $S(b_1, b_2, \ldots, b_m)$ deux relations de schéma différents mais partageant un ensemble d'attribut on appelle le jointure naturelle l'opération d'équi-jointure sur tous les attributs communs. La relation résultante comporte les attributs $(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m | b_i \notin A_R \cup A_S)$

Un exemple en SQL

Considérons eval_b, la relation produite par :

```
SELECT etu_id AS eid, cours_id AS cid, note FROM eval
```

```
SELECT * FROM etu NATURAL JOIN eval_b
```



Arbre algèbrique

Représentation arborescente d'une expression algébrique.

Algèbre Relationnelle: Écriture sous forme d'arbre

Π_{rating}

$\sigma_{year>1980}$

$\bowtie_{mid=mid}$

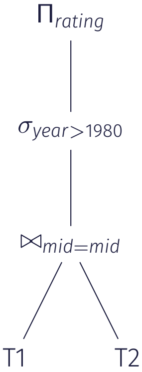
T1 T2

Arbre algèbrique

Représentation arborescente d'une expression algébrique.

Dans un SGBD : Des fonctions implémentent les opérateurs

- Un ordre dans l'execution des opérateurs
- Différentes manières d'exécuter les opérations
- Stockage RAM des relations intermédiaires



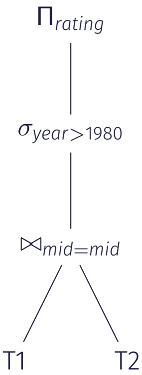
Arbre algèbrique

Représentation arborescente d'une expression algébrique.

Dans un SGBD : Des fonctions implémentent les opérateurs

- Un ordre dans l'execution des opérateurs
- Différentes manières d'exécuter les opérations
- Stockage RAM des relations intermédiaires

$\rightarrow$ On parle alors de **plans d'execution**



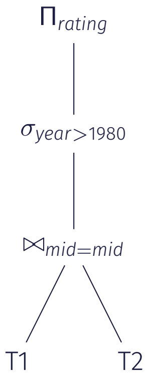
Arbre algèbrique

Représentation arborescente d'une expression algébrique.

Représentation arborescente des plans d'execution : Les opérations auront les priorités suivantes :

- Pour un nœud, l'élément de gauche sera calculé avant
- L'ordre des calculs se fait des feuilles vers la racine

Algèbre Relationnelle: Écriture sous forme d'arbre



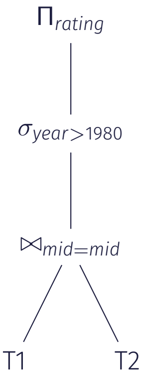
Arbre algèbrique

Représentation arborescente d'une expression algébrique.

Représentation arborescente des plans d'execution : Les opérations auront les priorités suivantes :

- Pour un nœud, l'élément de gauche sera calculé avant
- L'ordre des calculs se fait des feuilles vers la racine

On peut réécrire l'arbre suivant comme :



Arbre algèbrique

Représentation arborescente d'une expression algébrique.

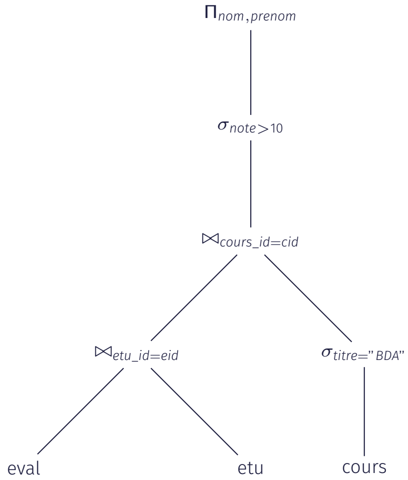
Représentation arborescente des plans d'execution : Les opérations auront les priorités suivantes :

- Pour un nœud, l'élément de gauche sera calculé avant
- L'ordre des calculs se fait des feuilles vers la racine

On peut réécrire l'arbre suivant comme :

$$\Pi_{rating}(\sigma_{year<1980}(T_1 \bowtie_{mid} T_2))$$

Algèbre Relationnelle: Écriture sous forme d'arbre



Question 1

Quelle expression relationnelle est-elle décrite par cet arbre ?

Question 2

Donnez une requête SQL qui pourrait être équivalente à cette expression !

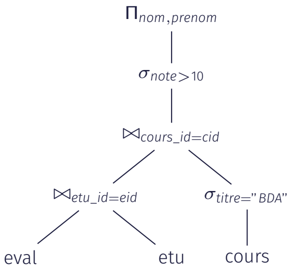
Question 3

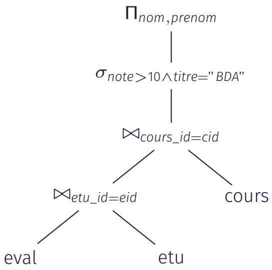
Expliquez en langage naturel ce que fait la requête !

Question 4

Dans quel ordre sont effectuées les opérations ?

Algèbre Relationnelle: Écriture sous forme d'arbre





- Les deux expressions sont équivalentes (même résultat)
- Les plans d'execution seront différents (temps d'execution différent)

$\rightarrow$ L'ordre des opérations impacte le temps d'execution

- Les deux expressions sont équivalentes (même résultat)
- Les plans d'execution seront différents (temps d'execution différent)

→ L'ordre des opérations influe sur le temps d'execution

Dans un SGBD il faut, évaluer le temps de plusieurs plans d'executions !!!

- Les deux expressions sont équivalentes (même résultat)
- Les plans d'execution seront différents (temps d'execution différent)

$\rightarrow$ L'ordre des opérations influe sur le temps d'execution

Dans un SGBD il faut, évaluer le temps de plusieurs plans d'executions !!!

$\downarrow$

Retrouver les expressions équivalentes !!!

Pourquoi c'est important ?

- Plusieurs écritures pour une même requête !!
- L'ordre des opérations peut influer sur le coût d'une requête !!
- À la base de l'optimisations des requêtes

Retrouver les expressions équivalentes

- Équivalences simples sur peu d'opérations
- Appliquer plusieurs équivalences simple

→ Explorer l'espace des expressions équivalentes

⚠ Tester toutes les équivalences possibles

Retrouver les expressions équivalentes

- Équivalences simples sur peu d'opérations
- Appliquer plusieurs équivalences simple

$\longrightarrow$ Explorer l'espace des expressions équivalentes

⚠ Tester toutes les équivalences possibles $\longrightarrow$ Coûteux (temps/mémoire)

Retrouver les expressions équivalentes

- Équivalences simples sur peu d'opérations
- Appliquer plusieurs équivalences simple

$\rightarrow$ Explorer l'espace des expressions équivalentes

⚠ Tester toutes les équivalences possibles $\rightarrow$ Coûteux (temps/mémoire)
$\rightarrow$ Des algorithmes parcourant seulement une sous-partie de l'espace des solutions (A*, algorithmes génétiques)

Sélection et conjonction :

La sélection sur une conjonction est équivalent à la composition de sélection sur chaque "partie" de la conjonction :

Sélection et conjonction :

La sélection sur une conjonction est équivalent à la composition de sélection sur chaque "partie" de la conjonction :

$$\sigma_{c_1 \wedge c_2}(R) = \sigma_{c_1}(\sigma_{c_2}(R))$$

Sélection et conjonction :

La sélection sur une conjonction est équivalent à la composition de sélection sur chaque "partie" de la conjonction :

$$\sigma_{c_1 \wedge c_2}(R) = \sigma_{c_1}(\sigma_{c_2}(R))$$

Exemple:

Sélectionner les identifiants où $nom =' Zeblouse'$ et $prenom =' Agathe'$

- $\Pi_{eid}(\sigma_{nom='Zeblouse' \wedge prenom='Agathe'}(Etu))$
- $\Pi_{eid}(\sigma_{nom='Zeblouse'}(\sigma_{prenom='Agathe'}(Etu)))$

Exemple (arborescence):

Sélectionner les identifiants où $nom =' Zeblouse'$ et $prenom =' Agathe'$

Algèbre Relationnelle: Sélection et Conjonction

Exemple (arborescence):

Sélectionner les identifiants où $nom =' Zeblouse'$ et $prenom =' Agathe'$

$$\Pi_{eid}$$

$$|$$

$$\sigma_{nom='Zeblouse' \wedge prenom='Agathe'}$$

$$|$$

$$\mathrm{Etu}$$

Exemple (arborescence):

Sélectionner les identifiants où $nom =' Zeblouse'$ et $prenom =' Agathe'$

$$
\begin{array}{c}
\Pi_{eid} \\
| \\
\sigma_{nom='Zeblouse' \wedge prenom='Agathe'} \\
| \\
\text{Etu}
\end{array}
\quad \rightarrow \quad
\begin{array}{c}
\Pi_{eid} \\
| \\
\sigma_{nom='Zeblouse'} \\
| \\
\sigma_{prenom='Agathe'} \\
| \\
\text{Etu}
\end{array}
$$

Commutativité de la sélection

$\sigma_{C_1}(\sigma_{C_2}(R)) = \sigma_{C_2}(\sigma_{C_1}(R))$

NB : cette propriété se déduit de la précédente

Exemple:

Sélectionner les identifiants où $nom =' Zeblouse'$ et $prenom =' Agathe'$

Commutativité de la sélection

$\sigma_{C_1}(\sigma_{C_2}(R)) = \sigma_{C_2}(\sigma_{C_1}(R))$

NB : cette propriété se déduit de la précédente

Exemple:

Sélectionner les identifiants où $nom =' Zeblouse'$ et $prenom =' Agathe'$

- $\Pi_{eid}(\sigma_{prenom='Agathe'}(\sigma_{nom='Zeblouse'}(Etu)))$
- $\Pi_{eid}(\sigma_{nom='Zeblouse'}(\sigma_{prenom='Agathe'}(Etu)))$

Algèbre Relationnelle: Commutativité de la sélection

Exemple (arborescence):

Sélectionner les identifiants où $nom =' Zeblouse'$ et $prenom =' Agathe'$

Π_{eid}

|

$\sigma_{prenom='Agathe'}$

|

$\sigma_{nom='Zeblouse'}$

|

Etu

$\rightarrow$

Algèbre Relationnelle: Commutativité de la sélection

Exemple (arborescence):

Sélectionner les identifiants où $nom =' Zeblouse'$ et $prenom =' Agathe'$

$$\begin{array}{ccc}
\Pi_{eid} & & \Pi_{eid} \\
| & & | \\
\sigma_{prenom='Agathe'} & & \sigma_{nom='Zeblouse'} \\
| & \rightarrow & | \\
\sigma_{nom='Zeblouse'} & & \sigma_{prenom='Agathe'} \\
| & & | \\
\text{Etu} & & \text{Etu}
\end{array}$$

Simplification des projections

$\Pi_{A_1}(\Pi_{A_2}(\dots\Pi_{A_n}(R)\dots)) = \Pi_{A_1}(R)$, si $A_1 \subset A_2 \subset \dots \subset A_n$

Simplification des projections

$\Pi_{A_1}(\Pi_{A_2}(\dots\Pi_{A_n}(R)\dots)) = \Pi_{A_1}(R)$, si $A_1 \subset A_2 \subset \dots \subset A_n$

Exemple:

Sélectionner les noms et prénoms de tous les utilisateurs

Simplification des projections

$\Pi_{A_1}(\Pi_{A_2}(\ldots\Pi_{A_n}(R)\ldots)) = \Pi_{A_1}(R)$, si $A_1 \subset A_2 \subset \cdots \subset A_n$

Exemple:

Sélectionner les noms et prénoms de tous les utilisateurs

- $\Pi_{nom,prenom}(\Pi_{eid,nom,prenom}(Etu))$
- $\Pi_{nom,prenom}(Etu)$

Exemple (arborescence):

Sélectionner les noms et prénoms de tous les utilisateurs

Exemple (arborescence):

Sélectionner les noms et prénoms de tous les utilisateurs

$$\Pi_{nom,prenom}$$

$$\Pi_{eid,nom,prenom}$$

$$\text{Etu}$$

$$\rightarrow$$

Algèbre Relationnelle: Simplification des projections

Exemple (arborescence):

Sélectionner les noms et prénoms de tous les utilisateurs

$$\begin{array}{c} \Pi_{nom,prenom} \\ | \\ \Pi_{eid,nom,prenom} \\ | \\ \text{Etu} \end{array} \quad \rightarrow \quad \begin{array}{c} \Pi_{nom,prenom} \\ | \\ \text{Etu} \end{array}$$

projections et selection

$\Pi_{A_1}(\sigma_{c_1}(R_1)) = \sigma_{c_1(R_1)}(\Pi_{A_1}(R))$ si c_1 sur attribut de A_1

Exemple:

Sélectionner les "eid" des utilisateurs ayant un eid supérieur à 2

- $\Pi_{eid}(\sigma_{eid>2}(Etu))$
- $\sigma_{eid>2}(\Pi_{eid}(Etu))$

Algèbre Relationnelle: projection et selection

Exemple:

Sélectionner les “eid” des utilisateurs ayant un eid supérieur à 2

$$\Pi_{eid}$$

$$\sigma_{eid>2}$$

Etu

$$\rightarrow$$

Exemple:

Sélectionner les “eid” des utilisateurs ayant un eid supérieur à 2

$$\begin{tikzcd}
\Pi_{eid} \arrow[d, dash] & & \sigma_{eid>2} \arrow[d, dash] \\
\sigma_{eid>2} \arrow[d, dash] & \rightarrow & \Pi_{eid} \arrow[d, dash] \\
\text{Etu} & & \text{Etu}
\end{tikzcd}$$

Sélection sur un produit cartésien

$\sigma_{c_1}(R_1 \times R_2) = R_1 \bowtie_{c_1} R_2$

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

Sélection sur un produit cartésien

$\sigma_{c_1}(R_1 \times R_2) = R_1 \bowtie_{c_1} R_2$

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

- $\sigma_{Etu.eid=Eval.eid}(Etu \times Eval)$

Sélection sur un produit cartésien

$\sigma_{C_1}(R_1 \times R_2) = R_1 \bowtie_{C_1} R_2$

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

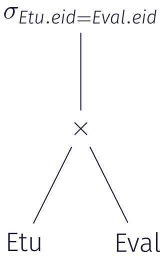
- $\sigma_{Etu.eid=Eval.eid}(Etu \times Eval)$
- $Etu \bowtie_{Etu.eid=Eval.eid} Eval$

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

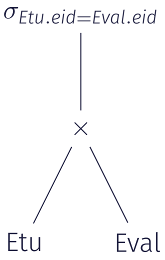


$\rightarrow$

Algèbre Relationnelle: Produit cartésien, sélection et jointure

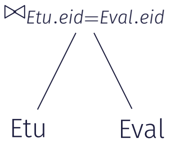
Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs



$\rightarrow$

avec arborescence



Commutativité des jointures

$R_1 \bowtie_{c_1} R_2 = R_2 \bowtie_{c_1} R_1$ (mais aussi du produit cartésien)

Commutativité des jointures

$R_1 \bowtie_{c_1} R_2 = R_2 \bowtie_{c_1} R_1$ (mais aussi du produit cartésien)

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

- $Eval \bowtie_{Eval.eid=Etu.eid} Etu$

Commutativité des jointures

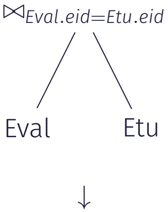
$R_1 \bowtie_{c_1} R_2 = R_2 \bowtie_{c_1} R_1$ (mais aussi du produit cartésien)

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

- $Eval \bowtie_{Eval.eid=Etu.eid} Etu$
- $Etu \bowtie_{Etu.eid=Eval.eid} Eval$

Représentation Arborescente



Commutativité des jointures

$R_1 \bowtie_{c_1} R_2 = R_2 \bowtie_{c_1} R_1$ (mais aussi du produit cartésien)

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

- $Eval \bowtie_{Eval.eid=Etu.eid} Etu$
- $Etu \bowtie_{Etu.eid=Eval.eid} Eval$

Algèbre Relationnelle: Commutativité des jointures

Commutativité des jointures

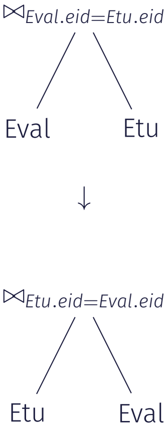
$R_1 \bowtie_{c_1} R_2 = R_2 \bowtie_{c_1} R_1$ (mais aussi du produit cartésien)

Exemple:

Sélectionner tous les noms et prénoms et les notes des utilisateurs

- $Eval \bowtie_{Eval.eid=Etu.eid} Etu$
- $Etu \bowtie_{Etu.eid=Eval.eid} Eval$

Représentation Arborescente



Associativité des équi-jointures

$(R_1 \bowtie R_2) \bowtie R_3 = R_1 \bowtie (R_2 \bowtie R_3)$

Exemple:

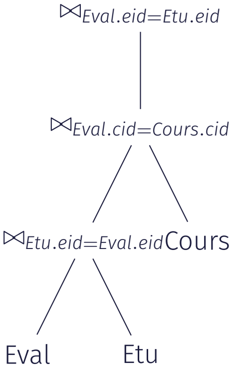
Sélectionner tous les noms et prénoms, cours et notes des utilisateurs

- $(Etu \bowtie_{Etu.eid=Eval.eid} Eval) \bowtie_{Eval.cid=Cours.cid} Cours$
- $Etu \bowtie_{Etu.eid=Eval.eid} (Eval \bowtie_{Eval.cid=Cours.cid} Cours)$

Algèbre Relationnelle: Associativité des équi-jointures

Exemple (arborescence):

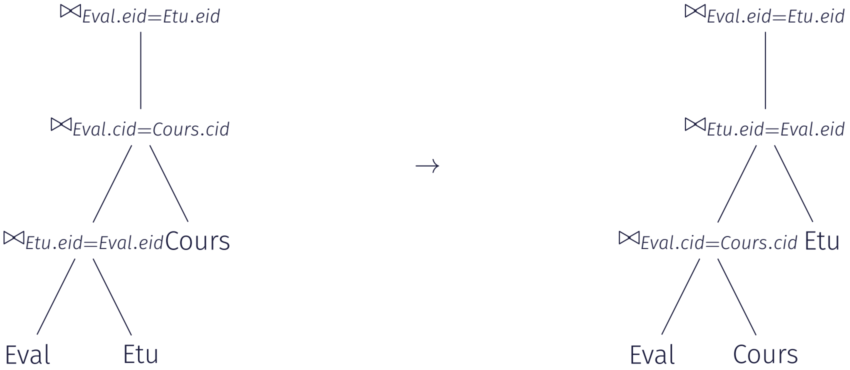
Sélectionner tous les noms et prénoms, cours et notes des utilisateurs



$\rightarrow$

Exemple (arborescence):

Sélectionner tous les noms et prénoms, cours et notes des utilisateurs



Sélection et jointure

$\sigma_{c_1}(R_1 \bowtie_{c_2} R_2) = \sigma_{c_1}(R_1) \bowtie_{c_2} R_2$ Si c_1 porte sur R_1

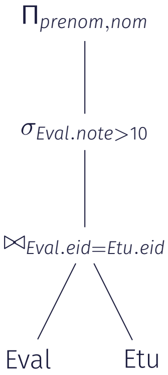
Exemple:

Sélectionner tous les noms, prénoms des utilisateurs ayant au moins une note > 10

- $\Pi_{prenom,nom}(\sigma_{Eval.note>10}(Eval \bowtie_{Eval.eid=Etu.eid} Etu))$
- $\Pi_{prenom,nom}(\sigma_{Eval.note>10}(Eval) \bowtie_{Eval.eid=Etu.eid} Etu)$

Exemple (arborescence):

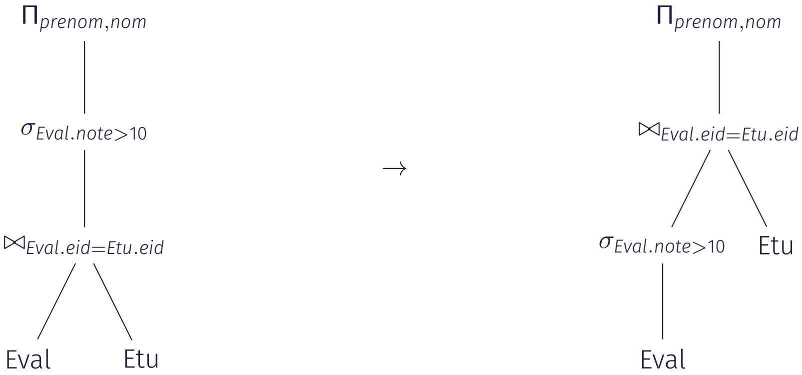
Sélectionner tous les noms, prénoms des utilisateurs ayant au moins une note > 10



$\rightarrow$

Exemple (arborescence):

Sélectionner tous les noms, prénoms des utilisateurs ayant au moins une note > 10



Algèbre Relationnelle: Résumé des équivalences

- $\sigma_{c_1 \wedge c_2}(R) = \sigma_{c_1}(\sigma_{c_2}(R))$: Conjonction et sélection
- $\sigma_{c_1}(\sigma_{c_2}(R)) = \sigma_{c_2}(\sigma_{c_1}(R))$: Commutativité des sélections
- $\Pi_{a_1}(\Pi_{a_2}(\ldots \Pi_{a_n}(R) \ldots)) = \Pi_{a_1}(R)$: Simplification des projections
- $\Pi_{a_1}(\sigma_{c_1}(R_1)) = \sigma_{c_1}(R_1)(\Pi_{a_1}(R))$: Selection-projection (si c_1 porte sur a_1)
- $\sigma_{c_1}(R_1 \times R_2) = R_1 \bowtie_{c_1} R_2$: La sélection sur un produit cartésien est équivalent à une theta-jointure sur la condition

Algèbre Relationnelle: Équivalences supplémentaires

- $R_1 \cup R_2 = R_2 \cup R_1$ commutativité de l'union
- $R_1 \cap R_2 = R_2 \cap R_1$ commutativité de l'intersection
- $(R_1 \cup R_2) \cup R_3 = R_1 \cup (R_2 \cup R_3)$: Associativité de l'union
- $(R_1 \cap R_2) \cap R_3 = R_1 \cap (R_2 \cap R_3)$: Associativité de l'intersection
- $\Pi_a(R_1 \cup R_2) = \Pi_a(R_1) \cup \Pi_a(R_2)$: Distributivité de la projection sur l'union
- $\sigma_{c_1}(R_1 \cup R_2) = \sigma_{c_1}(R_1) \cup \sigma_{c_1}(R_2)$: Distributivité de la sélection sur l'union (mais aussi l'intersection et la différence)

Exercice

Les deux requêtes ci-dessous sont-elles équivalentes ?

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(eval \bowtie_{eid=etu\_id} etu)))$

Réponse

Exercice

Les deux requêtes ci-dessous sont-elles équivalentes ?

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(eval \bowtie_{eid=etu\_id} etu)))$

Réponse

1. $\Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(R))) \Leftrightarrow \Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \wedge cours\_id=1}(R)))$

Exercice

Les deux requêtes ci-dessous sont-elles équivalentes ?

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \land cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(eval \bowtie_{eid=etu\_id} etu)))$

Réponse

1. $\Pi_{nom,prenom}(\sigma_{eid=1 \land cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(R))) \Leftrightarrow \Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \land cours\_id=1}(R)))$
2. $\Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \land cours\_id=1}(R))) \Leftrightarrow \Pi_{nom,prenom}(\sigma_{eid=1 \land cours\_id=1}(R))$

Exercice

Les deux requêtes ci-dessous sont-elles équivalentes ?

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(eval \bowtie_{eid=etu\_id} etu)))$

Réponse

1. $\Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(R))) \Leftrightarrow \Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \wedge cours\_id=1}(R)))$
2. $\Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \wedge cours\_id=1}(R))) \Leftrightarrow \Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(R))$
3. $\sigma_{eid=1 \wedge cours\_id=1}(eval \bowtie_{eid=etu\_id} etu) \Leftrightarrow \sigma_{eid=1}(\sigma_{cours\_id=1}(eval \bowtie_{eid=etu\_id} etu))$

Exercice

Les deux requêtes ci-dessous sont-elles équivalentes ?

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(eval \bowtie_{eid=etu\_id} etu)))$

Réponse

1. $\Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(R))) \Leftrightarrow \Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \wedge cours\_id=1}(R)))$
2. $\Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \wedge cours\_id=1}(R))) \Leftrightarrow \Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(R))$
3. $\sigma_{eid=1 \wedge cours\_id=1}(eval \bowtie_{eid=etu\_id} etu) \Leftrightarrow \sigma_{eid=1}(\sigma_{cours\_id=1}(eval \bowtie_{eid=etu\_id} etu))$
4. $\sigma_{eid=1}(\sigma_{cours\_id=1}(eval \bowtie_{eid=etu\_id} etu)) \Leftrightarrow \sigma_{eid=1}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} etu)$

Exercice

Les deux requêtes ci-dessous sont-elles équivalentes ?

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \land cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(eval \bowtie_{eid=etu\_id} etu)))$

Réponse

1. $\Pi_{nom,prenom}(\sigma_{eid=1 \land cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(R))) \Leftrightarrow \Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \land cours\_id=1}(R)))$
2. $\Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \land cours\_id=1}(R))) \Leftrightarrow \Pi_{nom,prenom}(\sigma_{eid=1 \land cours\_id=1}(R))$
3. $\sigma_{eid=1 \land cours\_id=1}(eval \bowtie_{eid=etu\_id} etu) \Leftrightarrow \sigma_{eid=1}(\sigma_{cours\_id=1}(eval \bowtie_{eid=etu\_id} etu))$
4. $\sigma_{eid=1}(\sigma_{cours\_id=1}(eval \bowtie_{eid=etu\_id} etu)) \Leftrightarrow \sigma_{eid=1}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} etu)$
5. $\sigma_{eid=1}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} etu) \Leftrightarrow \sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu)$

Exercice

Les deux requêtes ci-dessous sont-elles équivalentes ?

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(eval \bowtie_{eid=etu\_id} etu)))$

Réponse

1. $\Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(\Pi_{nom,prenom,eid,cours\_id}(R))) \Leftrightarrow \Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \wedge cours\_id=1}(R)))$
2. $\Pi_{nom,prenom}(\Pi_{nom,prenom,eid,cours\_id}(\sigma_{eid=1 \wedge cours\_id=1}(R))) \Leftrightarrow \Pi_{nom,prenom}(\sigma_{eid=1 \wedge cours\_id=1}(R))$
3. $\sigma_{eid=1 \wedge cours\_id=1}(eval \bowtie_{eid=etu\_id} etu) \Leftrightarrow \sigma_{eid=1}(\sigma_{cours\_id=1}(eval \bowtie_{eid=etu\_id} etu))$
4. $\sigma_{eid=1}(\sigma_{cours\_id=1}(eval \bowtie_{eid=etu\_id} etu)) \Leftrightarrow \sigma_{eid=1}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} etu)$
5. $\sigma_{eid=1}(\sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} etu) \Leftrightarrow \sigma_{cours\_id=1}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu)$
6. $A \Leftrightarrow B$

Exercice

Les deux requêtes ci-dessous sont-elles équivalentes ?

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=2}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \vee cours\_id=2}(eval \bowtie_{eid=etu\_id} etu))$

Réponse

Eval			
nid	etu_id	cours_id	note
1	1	2	0
4	3	2	15
5	3	3	6

Etu			
eid	nom	prenom	mail
1	Zeblouse	Agathe	az@*****
2	Huai	Odile	oh@*****
3	Peuplu	Jean	jp@*****
4	Hochon	Paul	hp@*****
5	Gator	Ali	ag@*****

Algèbre Relationnelle: Équivalence exercice

Données

Eval			
nid	etu_id	cours_id	note
1	1	2	0
4	3	2	15
5	3	3	6

Etu			
eid	nom	prenom	mail
1	Zeblouse	Agathe	az@*****
2	Huai	Odile	oh@*****
3	Peuplu	Jean	jp@*****
4	Hochon	Paul	hp@*****
5	Gator	Ali	ag@*****

- $A = \Pi_{nom,prenom}(\sigma_{cours\_id=2}(eval) \bowtie_{eid=etu\_id} \sigma_{eid=1}(etu))$
- $B = \Pi_{nom,prenom}(\sigma_{eid=1 \vee cours\_id=2}(eval \bowtie_{eid=etu\_id} etu))$

Résultats

A	
nom	prenom
Zeblouse	Agathe

B	
nom	prenom
Zeblouse	Agathe
Peuplu	Jean

Soit les deux requêtes **équivalentes** suivantes:

- $A = \sigma_{R_1.id=1}(R_1 \bowtie_{R_1.a=R_2.b} R_2)$
- $B = (\sigma_{R_1.id=1}(R_1) \bowtie_{R_1.a=R_2.b} R_2)$

Si $\sigma_{R_1.id=1}(R_1)$ ne contient qu'un seul enregistrement et que R_1 et R_2 contiennent respectivement 1000 enregistrements et 500 enregistrements sont issus de la jointure $C = R_1 \bowtie_{R_1.a=R_2.b} R_2$

Combien d'opérations pour A ?

- Itérations pour la jointure $\rightarrow 1000 \times 1000$
- Parcourir tous les enregistrements pour $\sigma_{R_1.id=1}(C) \rightarrow 500$

Total $1000 \times 1000 + 500 \approx 10^6$

Soit les deux requêtes **équivalentes** suivantes:

- $A = \sigma_{R_1.id=1}(R_1 \bowtie_{R_1.a=R_2.b} R_2)$
- $B = (\sigma_{R_1.id=1}(R_1) \bowtie_{R_1.a=R_2.b} R_2)$

Si $\sigma_{R_1.id=1}(R_1)$ ne contient qu'un seul enregistrement et que R_1 et R_2 contiennent respectivement 1000 enregistrements et 500 enregistrements sont issus de la jointure $C = R_1 \bowtie_{R_1.a=R_2.b} R_2$

Combien d'opérations pour B ?

- Parcourir tous les enregistrements pour $\sigma_{R_1.id=1}(R_1) \rightarrow 1000$
- Itérations pour la jointure $\rightarrow 1000 \times 1$ (car 1 élément $\sigma_{R_1.id=1}(R_1)$)

Total $1000 + 1000 = 2000$

Algèbre Relationnelle: Vers l'optimisation (intuition)

Soit les deux requêtes **équivalentes** suivantes:

- $A = \sigma_{R_1.id=1}(R_1 \bowtie_{R_1.a=R_2.b} R_2)$
- $B = (\sigma_{R_1.id=1}(R_1) \bowtie_{R_1.a=R_2.b} R_2)$

Si $\sigma_{R_1.id=1}(R_1)$ ne contient qu'un seul enregistrement et que R_1 et R_2 contiennent respectivement 1000 enregistrements et 500 enregistrements sont issus de la jointure $C = R_1 \bowtie_{R_1.a=R_2.b} R_2$

B **beaucoup plus efficace que** A **!!!**

Algèbre Relationnelle: Conclusion

- Un “langage” pour définir des requêtes sur des bases de données relationnelles
- Transcrire du SQL en un expression d’algèbre relationnelle
- Estimer si deux expressions d’AR sont équivalentes