

**TD1 - Séance 2- Cours 2 Algèbre relationnelle :**  
**Exercices et exemples de cours - Produits et**  
**jointures de la BD university – (Requête Optimale)**

Soit le schéma de la base de données university (MPD) suivante :

Faculty(facId, name, department, rank)

Class(classNo, *facId*, schedule, room)

Student(stuId, lastName, firstName, major, credits)

Enroll(stuId, classNo, grade)

Soit les 4 tables (relations) en intention en extension suivantes :

Relation Student(stuId, lastName, firstName, major, credits)

stuId	lastName	firstName	major	credits
S1001	Smith	Tom	History	90
S1002	Chin	Ann	Math	36
S1005	Lee	Perry	History	3
S1010	Burns	Edward	Art	63
S1013	McCarthy	Owen	Math	0
S1015	Jones	Mary	Math	42
S1020	Rivera	Jane	CSC	15

La relation Faculty (facId, name, department, rank)

facId	name	department	rank
F101	Adams	Art	Professor
F105	Tanaka	CSC	Instructor
F110	Byrne	Math	Assistant
F115	Smith	History	Associate
F221	Smith	CSC	Professor

La relation Enroll (stuId, classNumber, grade)

stuId	classNumber	grade
S1001	ART103A	A
S1001	HST205A	C
S1002	ART103A	D
S1002	CSC201A	F
S1002	MTH103C	B
S1010	ART103A	
S1010	MTH103C	
S1020	CSC201A	B
S1020	MTH101B	A

La relation Class(classNumber, facId, schedule, room)

classNumber	facId	schedule	room
ART103A	F101	MWF9	H221
CSC201A	F105	TuThF10	M110
CSC203A	F105	MThF12	M110
HST205A	F115	MWF11	H221
MTH101B	F110	MTuTh9	H225
MTH103C	F110	MWF11	H225

Demande 1:

Les deux requêtes suivantes sont équivalentes :

✓ R1 (version1):

- 1) SELECT Student WHERE lastName='Chin' AND firstName ='Ann' GIVING Temp1
- 2) Temp1 JOIN Enroll GIVING Temp2
- 3) PROJECT Temp2 OVER (classNo, grade) GIVING Answer

✓ R2 (version2) :

$\Pi_{classNo, grade}((\sigma_{lastName='Chin' \wedge firstName='Ann'}(Student)) \Join Enroll)$

Question :

Rédigez la requête précédente en utilisant uniquement les opérateurs fondamentaux de l'algèbre relationnelle (sélection, projection, produit cartésien, différence, renommage, etc.). À chaque étape, décrivez explicitement la relation résultante :

[1] En intention allégée.

[2] En extension

Réponse :

T0 : temp0  $\leftarrow \sigma_{lastName='Chin' \wedge firstName='Ann'}(Student)$

En intention :

temp0(studId, lastName, firstName, major, credits)

En extension : (5C \* 1L)

temp0

studId	lastName	firstName	major	credits
S1002	Chin	Ann	Math	36

T1 :

temp1  $\leftarrow$  temp0 x Enroll

En intention :

temp1(temp0.studId, lastName, firstName, major, credits, enroll.studId, classNumber, grade)

En extension : (8C \*9L)

temp1

Temp0.studId	lastName	firstName	major	credits	enroll.studId	classNumber	grade
S1002	Chin	Ann	Math	36	S1001	ART103A	A
S1002	Chin	Ann	Math	36			
S1002	Chin	Ann	Math	36			
S1002	Chin	Ann	Math	36			
S1002	Chin	Ann	Math	36			
S1002	Chin	Ann	Math	36			
S1002	Chin	Ann	Math	36			
S1002	Chin	Ann	Math	36			
S1002	Chin	Ann	Math	36	S1020	MTH101B	A

**Temp2**  $\leftarrow \sigma_{temp0.studId = enroll.studId} (temp1)$

**En intention :**

**Temp2(temp0.studId, lastName, firstName, major, credits, enroll.studId, classNumber, grade)**

**En extension : (8C \*3L)**

**Temp2**

Temp0.studId	lastName	firstName	major	credits	enroll.studId	classNumber	grade
S1002	Chin	Ann	Math	36	S1002	ART103A	D
S1002	Chin	Ann	Math	36	S1002	CSC201A	F
S1002	Chin	Ann	Math	36	S1002	MTH103C	B

**temp3**  $\leftarrow \Pi_{temp0.studId, lastName, firstName, major, credits, classNumber, grade} (temp2)$

**En intention :**

**Temp3(temp0.studId, lastName, firstName, major, credits, classNumber, grade)**

**En extension : (7C \*3L)**

**Temp3**

Temp0.studId	lastName	firstName	major	credits	classNumber	grade
S1002	Chin	Ann	Math	36	ART103A	D
S1002	Chin	Ann	Math	36	CSC201A	F
S1002	Chin	Ann	Math	36	MTH103C	B

**T2:**

**answer**  $\leftarrow \Pi_{classNo, grade} (temp3)$

**En intention :**

**answer (classNumber, grade)**

**En extension : (2C \*3L)**

**answer**

classNumber	grade
ART103A	D
CSC201A	F

MTH103C

B

**Question 1 : synthèse et optimisation**

**Sans sélection préalable :**

Si nous effectuons la jointure sans appliquer la sélection sur la table Student (c'est-à-dire que nous joignons toutes les lignes de Student avec toutes les lignes de Enroll), cela impliquerait une jointure entre les 7 lignes de Student et les 9 lignes de Enroll. Cela donne :

7×9=63 comparaisons

**Avec sélection préalable :**

Si nous appliquons d'abord la sélection pour filtrer uniquement l'étudiant "Ann Chin" dans la table Student, la sélection va réduire la table Student à une seule ligne.

Ensuite, cette seule ligne de Student sera jointe avec la table Enroll, ce qui implique seulement les comparaisons nécessaires pour l'id de l'étudiant (supposons que l'étudiant "Ann Chin" soit inscrit dans 9 cours). Ainsi, la jointure donne :

1×9=9 comparaisons

**Résumé des performances :**

- Sans sélection préalable : 63 comparaisons
- Avec sélection préalable : 9 comparaisons

**Conclusion :**

L'optimisation avec la sélection préalable réduit le nombre de comparaisons de 63 à 9, ce qui est un gain de performance significatif, surtout lorsque le nombre d'enregistrements dans les tables augmente. Dans ce cas précis, appliquer la sélection avant la jointure améliore l'efficacité de la requête de manière notable.



## Questions/TAF-R :

Refaire la même synthèse pour les deux requêtes suivantes :

✓ **R3(version3):**

$\Pi_{\text{classNo}, \text{grade}}(\Pi_{\text{stuld}}(\sigma_{\text{lastName}='Chin' \wedge \text{firstName}='Ann'}(\text{Student})) \mid x \mid \text{Enroll})$

✓ **R4(version4):**

$\Pi_{\text{classNo}, \text{grade}}(\sigma_{\text{lastName}='Chin' \wedge \text{firstName}='Ann'}(\text{Student} \mid x \mid \text{Enroll}))$

**Bilan et synthèse d'optimisation !**

**Donner la requête optimale. Justifier !**

Le code dans le langage SQL de cette demande est :

**Jointure implicite (norme SQL-89)**

```
SELECT E.classNo, E.grade
FROM Student S, Enroll E
WHERE S.stuld = E.stuld
      AND S.lastName = 'Chin'
      AND S.firstName = 'Ann';
```

**Ou, Jointure explicite (norme SQL-92 et versions suivantes)**

```
SELECT E.classNo, E.grade
FROM Student S
      JOIN Enroll E ON S.stuld = E.stuld
      WHERE S.lastName = 'Chin' AND S.firstName = 'Ann';
```

**Ou, avec une sous-requête :**

```
SELECT E.classNo, E.grade
```

```
FROM (
  SELECT S.stuld
  FROM Student S
  WHERE S.lastName = 'Chin'
        AND S.firstName = 'Ann'
```

```
) SubQuery
```

```
JOIN Enroll E ON SubQuery.stuld = E.stuld;
```

**Demande 2** : Nous souhaitons trouver les identifiants de tous les étudiants inscrits dans les classes du professeur Adams.

Donner la version optimale en AR. Il faut suivre la même démarche de la demande 1.

Donner le code en SQL. Avec jointure implicite et explicite.

**Demande 3** : L'opération Rename/ rename operator,  $\rho$  (CM)

Les données de cette demande c'est dans le TD1.

«Trouvez l'ID et le nom des instructeurs qui gagnent plus que l'instructeur dont l'ID est 10101 et moins que l'instructeur dont l'ID 12121»

AR - Donner deux versions :

- 1) nous voulons faire tout cela en plusieurs expressions
- 2) nous voulons faire tout cela en une seule expression (comme en CM)

SQL – Donner le code en SQL