



**Questions:**

1. Quel est le résultat de chaque requête ?
2. Traduire chaque requête en une expression de l'algèbre relationnelle
3. Optimiser la requête en appliquant les règles de transformation

**Exercice 4 :**

Soit une relation  $R(A, B)$  contenant  $r$  tuples, et une relation  $S(B, C)$  contenant  $s$  tuples; on suppose que  $r, s > 0$ . Ne pas faire d'hypothèse sur les clés.

Pour chacune des expressions algébriques ci-dessous, donner le nombre minimal et maximal de tuples du résultat en fonction de  $r$  et  $s$ .

- a)  $\pi_{A,C}(R \bowtie S)$
- b)  $\pi_B R - (\pi_B R - \pi_B S)$
- c)  $(R \bowtie S) \bowtie R$
- d)  $\sigma_{A>B} R \cup \sigma_{A<B} R$

**Exercice 5 :**

On considère trois relations  $r_1(A, B, C)$ ,  $r_2(C, D, E)$ , et  $r_3(E, F)$  avec A, C, et E leurs clés primaires respectivement. Supposons que  $r_1$  comporte 1000 tuples,  $r_2$  comporte 1500 tuples et  $r_3$  comporte 750 tuples.

**Question :**

1. Donner une estimation de la taille de  $r_1 \bowtie r_2 \bowtie r_3$
2. donner une stratégie efficace pour calculer la jointure.

**Exercice 6 :**

Soit le schéma relationnel suivant :

Etudiants (Eid, nom, age, adresse)

Livre (Lid, titre, auteur)

Pret (Eid, Lid, date\_Pret)

Soit la requête SQL suivante :

```
SELECT E.nom
FROM Etudiant E, Livre L, Pret P
WHERE E.Eid = P.Eid AND L.Lid = P.Lid AND L.auteur = 'Agatha Christie' AND E.age > 12 AND E.age < 20
```

1. Que fait cette requête ?
2. Ecrire l'expression algébrique correspondant exactement à cette requête.
3. Donner un plan logique optimisé de cette requête.
4. On considère maintenant les données suivantes sur les tables:
  - La table Etudiant comporte 10000 enregistrements stockés sur 1000 pages
  - La table Livre comporte 50000 enregistrements stockés sur 5000 pages
  - La table Prêt comporte 300000 enregistrements stockés sur 15000 pages

**Question :** En supposant qu'il n'existe aucun index et que les tables ne sont pas triées, donner un plan physique (préciser l'algorithme à utiliser pour chaque opération et la cardinalité de chaque table de jointure).

5. Afin d'optimiser plus la requête, proposer deux indexes et un plan physique optimal exploitant ces indexes.

6. Ecrire la requête SQL qui renvoie le nombre de prêts effectués par chaque étudiant.

Pour exécuter la requête précédente, doit-on utiliser de la matérialisation des résultats ou du pipeline ? Justifier.

**Exercice 7 :**

On considère le schéma de base de données suivant :

Produit (Pid, Pnom, Prix)

Entreprise (Eid, Enom, Adresse, Pid#)

Les caractéristiques de cette BD sont comme suit :

- Cardinalité(Produit) = 20000 tuples, Taille(enregistrement) = 20 octets
- Cardinalité(Entreprise) = 1000 tuples, Taille(enregistrement) = 25 octets
- Taille(page) = 5000 octets
- Le tampon(buffer) mémoire réservé à cette base de données comporte 102 frames (un frame peut contenir une page)

1. Calculer le nombre de pages des tables Produit et Entreprise

2. Soit la requête

```
SELECT Pnom, Enom
```

```
FROM Produit, Entreprise
```

```
WHERE Produit.Pid = Entreprise.Pid AND Adresse = 'Constantine'
```