# Chapitre 5

# Optimisations de requêtes

## 5.1 Généralités sur l'optimisation des requêtes

*Objectif de l'évaluation des requêtes :*

- Évaluateur traduit une requête SQL en un programme appelé plan d'exécution
- Plan d'exécution se compose :
  - Des opérations de l'algèbre relationnelle avec quelques extensions (tri, agrégat)
  - D'opérations de transfert de données
  - De primitives de synchronisation
- Facteurs d'optimisation (Optimiser quelles ressources) :
  - CPU, nombre d'E/S, taille mémoire nécessaire, coût de transfert, gain dû au parallélisme

*Etapes de l'évaluation*

1. Traduction de la requête SQL en algèbre relationnelle
2. Tracer l'arbre algébrique
3. Déduire les plans d'exécutions pour l'arbre algébrique
4. Calculer les coûts des plans
5. Sélectionner un plan

## 5.2 Traduction de la requête SQL en algèbre relationnelle

Rappel de vocabulaire : Algèbre relationnelle

1. La projection $\pi$ : $\pi_{Nom,adresse}$.
2. La sélection $\sigma$ : $\sigma_{adresse='Bejaia'}$.
3. Le produit cartésien x : R x S.
4. La jointure $\infty$ : Client $\infty_{numero=no\_client} Vente$.
5. L'union $\cup$ : $R \cup S$.
6. La déférence - : R - S.
7. Intersection, division, etc

*Exemple :* Requête SQL
SELECT Nom, Bureau
FROM Employe, Departement
WHERE Employe.numoD = Departement.numoD AND Employe.salaire > 1000

*Traduction en algèbre relationnelle*

$$\pi_{Nom,Bureau}(\sigma_{E.salaire>1000}(Employe \infty_{E.numD=D.numD} Departement))$$

## 5.3 Tracer l'arbre algébrique

— Visualiser graphiquement une requête relationnelle traduite en une expression équivalente de l'algèbre relationnelle.
— *Feuilles :* tables utilisées dans la requête
— *Noeuds intermédiaires :* opérations algébriques
— *Noeud racine :* dernière opération algébrique avant le retour du résultat

— L'opération d'un noeud intermédiaire est déclenchée quand ses opérandes sont disponibles.

On remplace alors le noeud par la relation produite par l'opération.
*Exemple :*

$$\pi_{Nom,Bureau}(\sigma_{E.salaire>1000}(Employe \infty_{E.numD=D.numD} Departement))$$

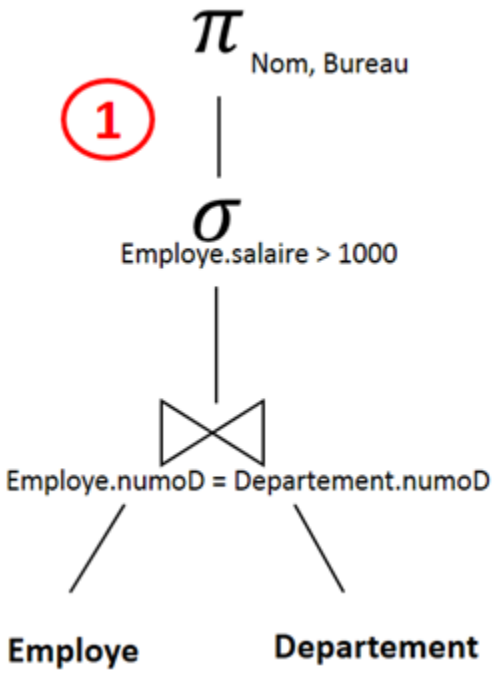


FIGURE 5.1 – l'arbre algébrique

## 5.4 Déduire les plans d'exécutions pour l'arbre algébrique

Le principe général de l'optimisation repose sur le constat suivant :

1. Les opérations unaires produisent des tables plus petites que la table d'origine.

2. Les opérations binaires produisent des tables plus grandes que la table d'origine. Autrement dit, ce sont les produits cartésiens des jointures qui accroissent la taille des tables intermédiaires.
   ⇒ *Idée :* Supprimer un maximum de lignes et de colonnes avant de faire les jointures et faire les jointures avant les produits cartésiens.

### 5.4.1 Règles de transformation : (La base mathématique)

— *Règle 1 :* Commutativité et associativité de $\infty$ et x :

$$E_1 \infty E_2 = E_2 \infty E_1$$
$$(E_1 \infty E_2 ) \infty E_3 = E_1 \infty (E_2 \infty E_3)$$

Si $(E_1 \infty E_2)$ est plus grand que $(E_2 \infty E_3)$,choisir $E_1 \infty (E_2 \infty E_3)$

$$E_1 \text{ x } E_2 = E_2 \text{ x } E_1$$
$$(E_1 \text{ x } E_2 ) \text{ x } E_3 = E_1 \text{ x } (E_2 \text{ x } E_3)$$

— *Règle 2 :* Cascade de projections

$$\pi_{A_1,...,A_n}(\pi_{B_1,...,B_m}(E)) = \pi_{A_1,...,A_n}(E)$$

— *Règle 3 :* Cascade de sélections

$$\sigma_{F_1}(\sigma_{F_2}(E)) = \sigma_{F_1 \wedge F_2}(E)$$

— *Règle 4 :* Commutativité de la sélection avec la projection

Si F ne porte que sur $A_1, ..., A_n$

$$\pi_{A_1,...,A_n}(\sigma_F(E)) = \sigma_F(\pi_{A_1,...,A_n}(E))$$

Si F porte aussi sur $B_1, ..., B_m$

$$\pi_{A_1,...,A_n}(\sigma_F(E)) = \pi_{A_1,...,A_n}(\sigma_F(\pi_{A_1,...,A_n,B_1,...,B_m}(E)))$$

— *Règle 5 :* Commutativité de la sélection avec x, $\cup$, -, $\infty$

$$\sigma_F(E_1 OP E_2) = \sigma_F(E_1) OP \sigma_F(E_2)$$

— *Règle 6 :* Commutativité de la projection avec $\cup$, x

$$\pi_{A_1,...,A_n}(E_1 OP E_2) = \pi_{A_1,...,A_n}(E_1) OP \pi_{A_1,...,A_n}(E_2)$$

L'optimisation intuitive se résume à :

1. Faire toutes les restrictions spécifiques pour limiter le nombre de tuples.
2. Faire toutes les projections mono-tables possibles pour limiter le nombre d'attributs.
3. Faire les jointures et après chaque jointure, les projections possibles.

4. Faire les "distinct", les "tris", les "group by", les fonctions de groupe.

En utilisant les règles de transformations précédentes, on arrive à l'algorithme suivant :

1. Séparer les restrictions comportant plusieurs prédicats (R3).
2. Faire descendre les restrictions le plus bas possible (R4, R5).
3. Regrouper les restrictions successives portant sur une même relation (R3).
4. Séparer les projections comportant plusieurs prédicats (R2).
5. Faire descendre les projections le plus bas possible (R4, R6).
6. Regrouper les projections successives (R2).

*Exemple :*

$$\pi_{Nom,Bureau}(\sigma_{E.salaire>1000}(Employe \infty_{E.numD=D.numD} Departement))$$

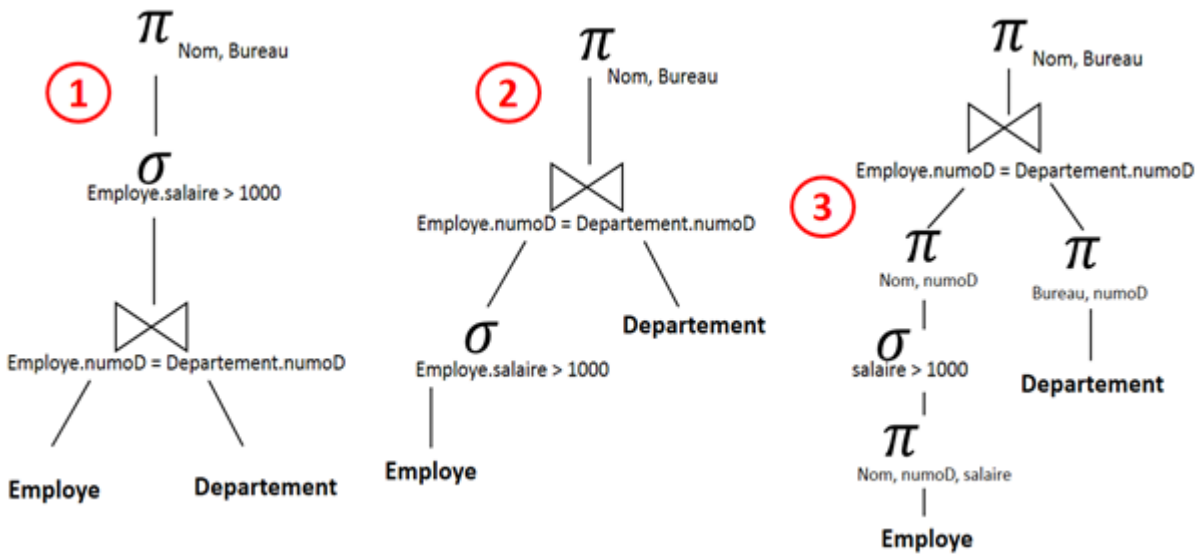


FIGURE 5.2 – (1)l'arbre algébrique (Plan initial) (2) Plan optimisé 1 (3) Plan optimisé 2

## 5.5 Calculer les coûts des plans

Le coût d'un plan dépend des algorithmes du calcul du coût des opérateurs algébriques. Il existe plusieurs algorithmes, donc, le coût peut changer d'un algorithme à l'autre.
On s'intéresse au coût des opérateurs $\infty$ et $\sigma$ en terme d'Entrées/Sortie
*Algorithme :*

- coût de la sélection : $\sigma_{condition}(R)$
  - coût (Entrée) = Lecture de tous les tuples- Tuples (R).
  - coût (Sortie) = Les tuples qui vérifient la condition.

- coût de la jointure : $R_1 \infty R_2$
  - coût (Entrée) = (Chargement) Lecture de R1, puis lecture de R2 = Tuples(R1)+( Tuples(R1)x Tuples(R2))
  - coût (Sortie) = Les tuples qui vérifient la condition : <= Tuples(R1)x Tuples(R2)

coût Total = $\sum_{i=1}^{n} OP_i$

*Exemple :*

$$\pi_{Nom,Bureau}(\sigma_{E.salaire>1000}(Employe \infty_{E.numD=D.numD} Departement))$$

*Hypothèses :*

1. Il y a 300 lignes dans "Employe"
2. Il y a 12 lignes dans "Departement"
3. On suppose qu'il n'y a que 20% d'employés ayant un salaire > 1000

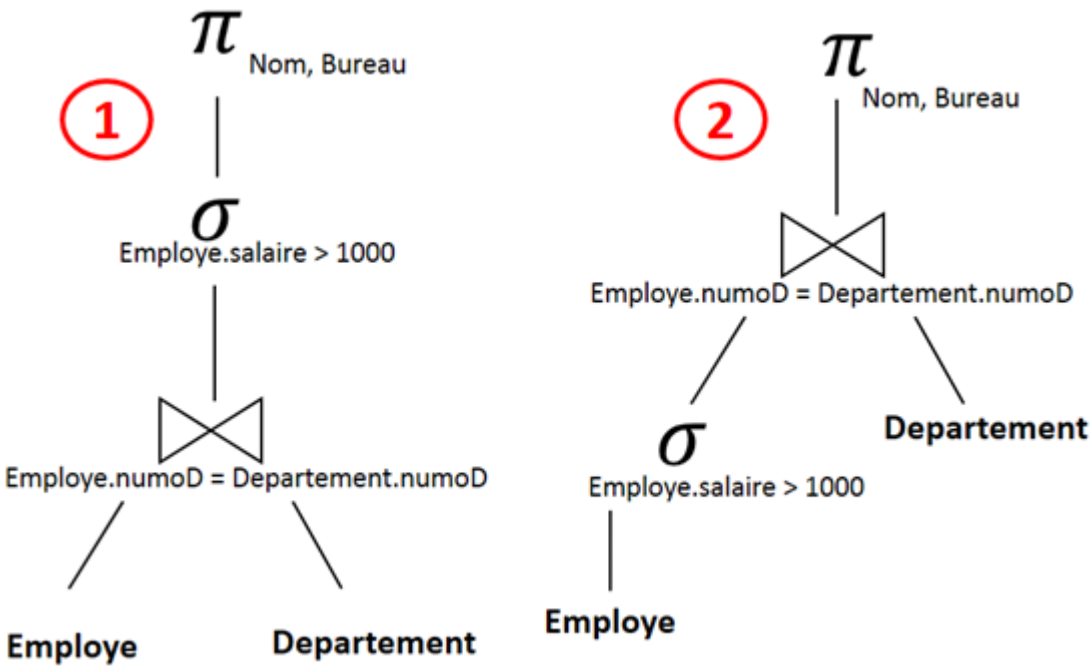


FIGURE 5.3 – (1) Plan initial (2) Plan optimisé

*Coût plan initial*

- Coût de la jointure :
  - coût (Entrée) = 300 + (300*12) = 3900E
  - coût (Sortie) = 300 S (pour chaque employé, un seul Département)
  - Total = 3900 + 300 = 4200 E/S
- Coût de la sélection :
  - coût (Entrée) = 300 E
  - coût (Sortie) = 300 * 0.2 = 60 S
  - Total = 300 + 60 = 360 E/S
- Coût Total = 4200 E/S + 360 E/S = 4560 E/S

*Coût plan optimisé*

- Coût de la sélection :
  - coût (Entrée) = 300 E
  - coût (Sortie) = 300 * 0.2 = 60 S
  - Total = 300 + 60 = 360 E/S
- Coût de la jointure :
  - coût (Entrée) = 60 + (60*12) = 780E
  - coût (Sortie) = 60 S (pour chaque employé, un seul Département)
  - Total = 780 + 60 = 840 E/S
- Coût Total = 360E/S + 840 E/S = 1200 E/S

## 5.6 Sélectionner un plan

Le principe général pour sélectionner un plan parmi d'autres consiste à sélectionner celui qui permet de limiter le nombre d'accès disques, soit, en première approximation, le nombre de tuples parcourus.