

---

## SW N°5: QUERY OPTIMIZATION COST-BASED OPTIMIZATION

---

### **Exorcise 1:**

Schéma de la BD Entreprise

EMPLOYES(NE, nom, job, datemb, sal, comm., #ND, \*NEchef)

DEPARTEMENTS (ND, nom, ville)

Avec : Employés(500 ; 8)

Départements (5 ; 3)

On ajoute la sélectivité suivante : S (départements ; nom= « Comptabilité ») = 10%

1. Liste des employés par ordre alphabétique du département comptabilité avec leurs jobs et le nom de leur supérieur hiérarchique
  - a. Express the query in SQL
  - b. Give the corresponding algebraic tree (plan)
  - c. Give an optimized algebraic tree with cost calculation (in each node).

### **Exercise 02 :**

#### **Schéma de la BD**

Buveurs (NB, Nom, Prénom, Type)

Vins (NV, Cru, Millésime, Degré)

Producteurs (NP, Nom, Région)

Abuser (#NB, Date, Quantité, #NV)

Produire (#NP, #NV)

Avec

- 100 buveurs : B = (100 ; 4)
- 50 vins : V = (50 ; 4)
- 20 producteurs : P = (20 ; 3)
- 250 abuser : A = (250 ; 4)
- 75 produire : R = (75 ; 2)

On ajoute les sélectivités suivantes :

- Sélectivité S(Producteurs ; Région = "Bordeaux") = 25%
- Sélectivité S(Vins ; Degré >13) = 20%

1. On cherche tous les buveurs ayant bu un Bordeaux de degré supérieur ou égal à 13.

### **Exercice 03 :**

Considérons deux relations R1(A, B) et R2(B, C) avec les statistiques suivantes :

- R1 a 10000 tuples et R2 a 20000 tuples.
- La valeur maximale et minimale de A de R1 sont 0 et 100, respectivement.
- Il y a 200 valeurs distinctes de C de R2.
- Il existe 100 et 150 valeurs distinctes de B dans R1 et R2, respectivement.
- 11 pages mémoire sont disponibles.
- la taille de page disque est égale à 10 tuples.

Soit la requête suivante à exécuter :

```
SELECT *  
FROM R1, R2  
WHERE R1.A ≥ 20  
AND R1.A ≤ 30  
AND R2.C = 50  
AND R1.B = R2.B
```

1. Donner l'arbre algébrique de cette requête
2. Estimer le coût (d'entrées-sorties) pour cette requête.