UNIVERSITE ABOU BEKR BELKAID – TLEMCEN
FACULTE DES SCIENCES - DEPARTEMENT D'INFORMATIQUE

Administration des SGBD

Corrigé TD N°4 : Optimisation des Requêtes (L3 2018-2019)

Exercice 1

Soit les deux relations :

Cinema (NomCinema, Adresse, Gerant)

Salle (NomCinema#, NumSalle, Capacite, Type)

1. **Requête SQL qui donne les adresses des cinémas ayant des salles d'une capacité > 150** :

   - SELECT Adresse FROM Cinema WHERE NomCinema IN
     (SELECT NomCinema FROM Salle WHERE capacité > 150);

     Ou

   - SELECT Adresse FROM Cinema, Salle
     WHERE Capacité > 150 AND Cinema.NomCinema = Salle.NomCinema;

2. **Optimisation de la requête SQL précédente** :

   - SELECT Adresse FROM (SELECT NomCinema, Adresse FROM Cinema) AS A,
     (SELECT NomCinema FROM Salle WHERE Capacité > 150) AS B
     WHERE A.NomCinema = B.NomCinema;

   Règles d'optimisation utilisées :

   - Réduire le plus tôt la taille et le nombre de tuples manipulé :
     - Tuples moins nombreux (réduction de cardinalité) : grâce à la sélection.
     - Tuples plus petits (réduction de degré) : grâce à la projection.
   - Traiter en une seule fois les opérations de restriction et projection sur une même relation.

Exercice 2

Soit le schéma relationnel de la base de données :

Employe (Matricule, NomEmp, Poste, DateEmbauche, MatriculeSupérieur#, Salaire, CodeDept#)

Departement (CodeDept, NomDept, Lieu)

Projet (CodeProjet, NomProj)

Participation (Matricule#, CodeProjet#, Fonction)

A. **Requêtes SQL, requêtes en algèbre relationnelle ainsi que les différents plans d'exécution des questions suivantes :**

1. Matricule et nom des employés qui ont été embauchés avant le 1 janvier 2000.
   - SELECT Matricule, NomEmp FROM Employe WHERE DateEmbauche < 01/01/2000;
   - $\prod_{[Matricule, NomEmp]} (\sigma_{[DateEmbauche < 01/01/2000]} (Employe))$
   - R1← Sélectionner de la table Employe ceux dont la DateEmbauche est < 01/01/2000
     R2← Projeter R1 sur Matricule et NomEmp
2. Nom des employés avec le nom du département où ils travaillent.
   - SELECT NomEmp, NomDept FROM Employe, Departement
     WHERE Employe.CodeDept = Departement.CodeDept;
   - $\prod_{[NomEmp, NomDept]} (Employe \bowtie Departement)$
   - R1← Joindre Employe avec Departement par CodeDept
     R2←Projeter R1 sur NomEmp et NomDept
3. Nom des employés qui travaillent dans le département Comptabilité.
   - SELECT NomEmp FROM Employe, Departement WHERE NomDept= 'Comptabilite' And Employe.CodeDept = Departement.CodeDept;
   - $\prod_{NomEmp} (\sigma_{[NomDept= 'Comptabilite']} (Employe \bowtie Departement))$
   - R1← Joindre Employe avec Departement par CodeDept
     R2← Sélectionner de la table R1 les tuples dont le NomDept= Comptab
     R3← Projeter R2 sur NomEmp
4. Matricule des employés qui participent à tous les projets.
   - SELECT Matricule FROM Participation GROUP BY Matricule HAVING COUNT (DISTINCT (CodeProjet)) = (SELECT COUNT (CodeProjet) FROM Projet);
   - $(\prod_{[Matricule, CodeProjet]} Participation) \div (\prod_{CodeProjet} Projet)$
   - R1← Projeter Participation sur Matricule et CodeProjet
     R2← Projeter Projet sur CodeProjet
     R3← R1/ R2
5. Nom des employés qui ne participent à aucun projet.
   - SELECT NomEmp FROM Employe WHERE Matricule NOT IN
     (SELECT Matricule FROM Participation);

- $\prod_{NomEmp}$ (Employe ⋈ ($\prod_{Matricule}$ Employe – $\prod_{Matricule}$ Participation))

- R1← Projeter Employe sur Matricule
  R2← Projeter Participation sur Matricule
  R3← R1- R2
  R4← Joindre Employe avec R3 par Matricule
  R5← Projeter R4 sur NomEmp

6. Nom des départements qui ont à la fois au moins un ingénieur et au moins une secrétaire.

- (SELECT NomDept FROM Departement, Employe
  WHERE Poste = 'Ingénieur' AND Employe.CodeDept =Departement.CodeDept)

  INTERSECT

  (SELECT NomDept FROM Departement, Employe
  WHERE Poste = 'Secrétaire' WHERE Employe.CodeDept = Departement.CodeDept);
- ($\prod_{NomDept}$ ($\sigma_{[Poste= 'Ingénieur']}$ (Employe ⋈ Departement)))

  ∩

  ($\prod_{NomDept}$ ($\sigma_{[Poste='Secrétaire']}$ (Employe ⋈ Departement)))
- R1← Joindre Employe avec Departement par CodeDept
  R2← Sélectionner de la table R1 les tuples dont le Poste=Ingénieur
  R3← Projeter R2 sur NomDept
  R4← Sélectionner de la table R1 les tuples dont le Poste= Secrétaire
  R5← Projeter R4 sur NomDept
  R6← R3 ∩ R5

B. Optimisation des requêtes SQL et algèbre relationnelle précédentes, en s'appuyant sur la règle d'optimisation « ***Réduire le plus tôt possible la taille et le nombre de tuples manipulé, en anticipant les opérations de sélection et de projection*** », et génération de nouveaux plans d'exécutions optimaux :

1. Matricule et nom des employés qui ont été embauchés avant le 1 janvier 2000.

- SELECT Matricule, NomEmp FROM (SELECT Matricule, NomEmp, DateEmbauche FROM Employe WHERE DateEmbauche< 01/01/2000);
- $\prod_{[Matricule, NomEmp]}$ ($\sigma_{[DateEmbauche < 01/01/2000]}$ ($\prod_{[Matricule, NomEmp, DateEmbauche]}$ (Employe))
- R1← Projeter Employe sur Matricule, NomEmp et DateEmbauche
  R2← Sélectionner de la table R1 les employés dont la DateEmbauche est < 01/01/2000
  R3← Projeter R2 sur Matricule et NomEmp

2. Nom des employés avec le nom du département où ils travaillent.

   - SELECT NomEmp, NomDept FROM (SELECT NomEmp, CodeDept FROM Employe) AS E, (SELECT NomDept, CodeDept FROM Departement AS D)
     WHERE E.CodeDept = D.CodeDept;

   - $\Pi_{[NomEmp, NomDept]} (\Pi_{[NomEmp, CodeDept]} (Employe) \bowtie \Pi_{[NomDept, CodeDept]} (Departement))$

   - R1← Projeter Employe sur NomEmp et CodeDept
     R2← Projeter Departement sur NomDept et CodeDept
     R3← Joindre R1 avec R2 par CodeDept
     R4← Projeter R3 sur NomEmp et NomDept (Pour éjecter le CodeDept)

3. Nom des employés qui travaillent dans le département Comptabilité.

   - SELECT NomEmp FROM (SELECT NomEmp, CodeDept FROM Employe) AS E, (SELECT * FROM Departement WHERE NomDept= 'Comptabilite') AS D WHERE E.CodeDept = D.CodeDept;

   - $\Pi_{NomEmp} ((\Pi_{[NomEmp, CodeDept]} Employe) \bowtie (\sigma_{[NomDept= 'Comptabilite']} Departement))$

   - R1← Projeter Employe sur NomEmp, CodeDept
     R2← Sélectionner de la table Departement les tuples dont le NomDept= Comptabilite
     R3← Joindre R1 avec R2 par CodeDept
     R4← Projeter R3 sur NomEmp

4. Matricule des employés qui participent à tous les projets.

   - Requête et plan d'exécution optimal.

5. Nom des employés qui ne participent à aucun projet.

   - SELECT NomEmp FROM (SELECT Matricule, NomEmp FROM Employe) AS E1, (SELECT Matricule FROM Employe WHERE Matricule NOT IN (SELECT Matricule FROM Participation)) AS E2 WHERE E1.Matricule = E2.Matricule;

   - Requête en AR déjà optimisée et plan d'exécution optimal.

6. Nom des départements qui ont à la fois au moins un ingénieur et au moins une secrétaire.

   - SELECT Dep_Ing.NomDept FROM (SELECT NomDept FROM Employe, Departement WHERE Poste = 'Ingénieur' AND Employe.CodeDept=Departement.CodeDept) AS Dep_Ing,

(SELECT NomDept FROM Employe, Departement WHERE Poste = 'Secrétaire' AND Employe.CodeDept=Departement.CodeDept) AS Dep_Sec
WHERE Dep_Ing.NomDept = Dep_Sec.NomDept ;

- $(\prod_{NomDept} (\sigma_{[Poste= 'Ingénieur']} (Employe \bowtie Departement))) \bowtie (\prod_{NomDept} (\sigma_{[Poste= ' Secrétaire']} (Employe \bowtie Departement)))$

- R1← Joindre Employe avec Departement par CodeDept
  R2← Sélectionner de la table R1 les tuples dont le Poste=Ingénieur
  R3← Projeter R2 sur NomDept
  R4← Sélectionner de la table R1 les tuples dont le Poste= Secrétaire
  R5← Projeter R4 sur NomDept
  R6← Joindre R3 avec R5 par NomDept

Exercice 3

Soit le schéma ci-dessous :

Etudiant (IdEtd, NomEtd, AdrEtd, Res)

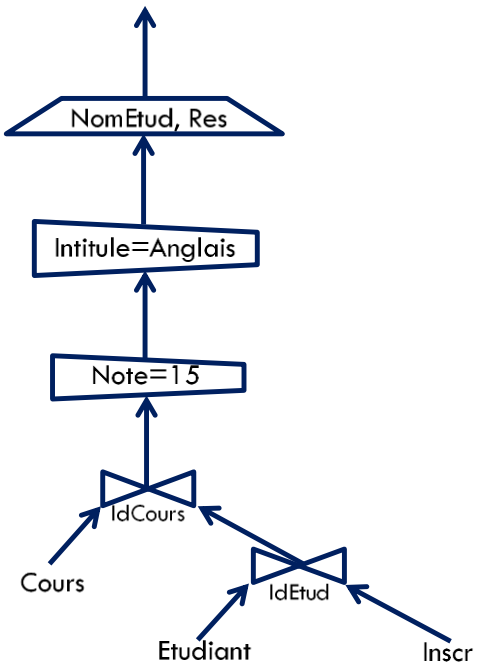
Formation (IdF, NomF, AdrF)

Cours (IdCours, IdF#, Intitule, Resp)

Inscr (NoCours#, NoEtd#, Note)

1. **Analyse de l'arbre ci-dessous et proposition d'amélioration** :

   - Réduire la table Etudiant en gardant IdEtd, NomEtd, Res.
   - Réduire la table Cours en gardant IdCours et Intitule.
   - Regrouper les 2 opérations de restriction (Libellé=Anglais) et (Note=15) avec la projection sur (NomEtd, Res).
   - L'ordre de jointure peut être modifié sachant les cardinalités des tables de jointure (Etudiant, Cours, Inscr).



2. **Traduction de l'arbre en algèbre relationnelle avant et après amélioration** :

Avant amélioration

- Π [NomEtd, Res] (σ [Intitule='anglais'] (σ [Note='15'] (σ [IdCours=NoCours] (Cours $\bowtie$ (σ [IdEtd=NoEtud] (Etudiant $\bowtie$ Inscr))))))

Aprés amélioration

- Π [NomEtd, Res] (σ [Intitule='anglais' and Note ='15'] (Π [NoCours, Intitule] (Cours) $\bowtie$ ($\Pi_{[IdEtd,NomEtd, Res]}$ (Etudiant) $\bowtie$ Inscr)))

3. **Optimisation des requêtes suivantes** :

- $\Pi_{[NomF, AdrF]}$ (σ [Resp='Mohamed'] (Cours $\bowtie$ Formation))

  Π [NomF, AdrF] ((Π **[IdF]** (σ **[Resp='Mohamed']** **Cours**)) $\bowtie$ Formation)

- SELECT NomF, Intitule, IdCours FROM Formation, Cours, Inscr, Etudiant WHERE (NomEtd = 'Amine') AND (Cours.IdCours = Inscr.NoCours) AND (Formation.IdF = Cours.IdF) AND (Etudiant.IdEtd = Inscr.NoEtd);

  SELECT NomF, Intitule, IdCours FROM Formation, Cours, Inscr, **(SELECT IdEtd FROM Etudiant WHERE NomEtd = 'Amine')**

  WHERE (Cours.IdCours = Inscr.NoCours) AND (Formation.IdF = Cours.IdF) AND (Etudiant.IdEtd = Inscr.NoEtd);

- SELECT NomEtd, Note, IdCours FROM Etudiant, Inscr, Cours, Formation WHERE (NomF = 'Informatique') AND (Intitule='ADM_BDD') AND (Cours.IdCours =Inscr.NoCours) AND (Formation.IdF =Cours.IdF) AND (Etudiant.IdEtd=Inscr.NoEtd);

```
SELECT NomEtd, Note, IdCours FROM Etudiant, Inscr,
(SELECT * FROM Cours WHERE Intitule='ADM_BDD'),
(SELECT IdF, NomF FROM Formation WHERE NomF= 'Informatique')
WHERE (Cours.IdCours = Inscr.NoCours) AND (Formation.IdF =Cours.IdF) AND
(Etudiant.IdEtd=Inscr.NoEtd);
```

NB : Les améliorations apportées ne sont pas exhaustives, en revanche, il y a d'autres améliorations qui peuvent être apportées

MATALLAH H