

**Administration des SGBD**  
**TD N°4 : (L3 2017-2018)**

**Exercice 1** Soit les relations suivantes :

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

DEPARTEMENT (CodeDept, NomDept, Lieu)

PROJET (CodeProjet, NomProj)

PARTICIPATION (Matricule#, CodeProjet#, Fonction)

**Donnez les requêtes suivantes en SQL et en algèbre relationnelle, puis proposer les arbres algébriques correspondants ainsi que les différents plans d'exécution de ces requêtes :**

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
```

  
 $\Pi_{[Matricule, NomEmp]}(\sigma_{[DateEmbauche < 01/01/2000]}(Employee))$   
 $R1 \leftarrow$  Sélectionner de la table *Employe* ceux dont la *DateEmbauche* est < 01/01/2000  
 $R2 \leftarrow$  Projeter  $R1$  sur *Matricule* et *NomEmp*
- $\Pi_{[Matricule, NomEmp]}(\sigma_{[DateEmbauche < 01/01/2000]}(\Pi_{[Matricule, NomEmp, DateEmbauche]}(Employee)))$   
 $R1 \leftarrow$  Projeter *EMPLOYE* sur *Matricule*, *NomEmp* et *DateEmbauche*  
 $R2 \leftarrow$  Sélectionner de la table  $R1$  les employés dont la *DateEmbauche* est < 01/01/2000  
 $R3 \leftarrow$  Projeter  $R2$  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
```

  
 $\Pi_{[NomEmp, NomDept]}(EMPLOYE \bowtie DEPARTEMENT)$   
 $R1 \leftarrow$  Joindre *EMPLOYE* avec *DEPARTEMENT* par *CodeDept*  
 $R2 \leftarrow$  Projeter  $R1$  sur *NomEmp* et *NomDept*
- $\Pi_{[NomEmp, NomDept]}(\Pi_{[NomEmp, CodeDept]}(EMPLOYE) \bowtie \Pi_{[NomDept, CodeDept]}(DEPARTEMENT))$   
 $R2 \leftarrow$  Projeter *EMPLOYE* sur *NomEmp* et *CodeDept*  
 $R2 \leftarrow$  Projeter *DEPARTEMENT* sur *NomDept* et *CodeDept*  
 $R3 \leftarrow$  Joindre  $R1$  avec  $R2$  par *CodeDept*  
 $R4 \leftarrow$  Projeter  $R3$  sur *NomEmp* et *NomDept* (Pour éjecter le *CodeDept*)

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

- ```
SELECT NomEmp
FROM EMPLOYE, DEPARTEMENT
WHERE NomDept= 'Comptabilite' And EMPLOYE.CodeDept = DEPARTEMENT.CodeDept
```

  
 $\Pi_{NomEmp}(\sigma_{[NomDept= 'Comptabilite']}(EMPLOYE \bowtie DEPARTEMENT))$   
 $R1 \leftarrow$  Joindre *EMPLOYE* avec *DEPARTEMENT* par *CodeDept*  
 $R2 \leftarrow$  Sélectionner de la table  $R1$  les tuples dont le *NomDept*= Comptabilite  
 $R2 \leftarrow$  Projeter  $R2$  sur *NomEmp*

- SELECT NomEmp  
FROM (SELECT NomEmp, CodeDept FROM EMPLOYE) AS E, (SELECT \* FROM DEPARTEMENT WHERE  
NomDept= 'Comptabilite') AS D  
WHERE E.CodeDept = D.CodeDept

- $\Pi_{\text{NomEmp}} (\text{EMPLOYE} \bowtie (\sigma_{[\text{NomDept} = \text{'Comptabilite'}]} \text{DEPARTEMENT}))$   
 $R1 \leftarrow \text{Projeter EMPLOYE sur NomEmp, CodeDept}$   
 $R2 \leftarrow \text{Sélectionner de la table DEPARTEMENT les tuples dont le NomDept= Comptabilite}$   
 $R3 \leftarrow \text{Joindre R1 avec R2}$   
 $R4 \leftarrow \text{Projeter R3 sur NomEmp}$

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)
- $(\Pi_{[\text{Matricule}, \text{CodeProjet}]} \text{PARTICIPATION}) \div (\Pi_{\text{CodeProjet}} \text{PROJET})$   
 $R1 \leftarrow \text{Projeter Participation sur Matricule et CodeProjet}$   
 $R2 \leftarrow \text{Projeter Projet sur CodeProjet}$   
 $R3 \leftarrow R1 / R2$

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

- SELECT NomEmp  
FROM EMPLOYE  
WHERE Matricule NOT IN (SELECT Matricule FROM PARTICIPATION)

Ou

- 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
- $\Pi_{\text{NomEmp}} (\text{EMPLOYE} \bowtie (\Pi_{\text{Matricule}} \text{EMPLOYE} - \Pi_{\text{Matricule}} \text{PARTICIPATION}))$   
 $R1 \leftarrow \text{Projeter EMPLOYE sur Matricule}$   
 $R2 \leftarrow \text{Projeter PARTICIPATION sur Matricule}$   
 $R3 \leftarrow R1 - R2$   
 $R4 \leftarrow \text{Joindre EMPLOYE avec R3}$   
 $R5 \leftarrow \text{Projeter R4 sur NomEmp}$

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) ;

Ou

- SELECT Dept\_Ing.NomDept FROM  
(SELECT NomDept FROM DEPARTEMENT, EMPLOYE  
WHERE Poste = 'Ingénieur' AND EMPLOYE.CodeDept =DEPARTEMENT.CodeDept ) AS Dept\_Ing,  
(SELECT NomDept FROM DEPARTEMENT, EMPLOYE  
WHERE Poste = 'Secrétaire') WHERE EMPLOYE.CodeDept =DEPARTEMENT.CodeDept) AS Dept\_Sec  
WHERE Dept\_Ing.NomDept = Dept\_Sec.NomDept ;

- $(\Pi_{\text{NomDept}} (\sigma_{[\text{Poste} = \text{'Ingénieur'}]} (\text{EMPLOYEE} \bowtie \text{DEPARTEMENT}))) \cap (\Pi_{\text{NomDept}} (\sigma_{[\text{Poste} = \text{'Secrétaire'}]} (\text{EMPLOYEE} \bowtie \text{DEPARTEMENT})))$

$R1 \leftarrow$  Joindre EMPLOYEE avec DEPARTEMENT par CodeDept  
 $R2 \leftarrow$  Sélectionner de la table R1 les tuples dont le Poste=Ingénieur  
 $R3 \leftarrow$  Projeter R2 sur NomDept  
 $R4 \leftarrow$  Sélectionner de la table R1 les tuples dont le Poste= Secrétaire  
 $R5 \leftarrow$  Projeter R4 sur NomDept  
 $R6 \leftarrow$  Joindre R3 avec R5 par NomDept

## Exercice 2 Soit deux relations :

CINEMA (NomCinema, Adresse, Gerant)

SALLE (NomCinema#, NumSalle, Capacite, Type)

### 1. Donner la 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. Optimiser votre requête SQL.

- ```
SELECT Adresse
FROM (SELECT NomCinema, Adresse FROM CINEMA) AS A,
(SELECT NomCinema FROM SALLE WHERE capacité > 150) AS B
WHERE A.NomCinema = B.NomCinema
```

## Exercice 3 Soit le schéma ci-dessous :

Etudiant (IdEtud, NomEtud, AdrEtud, Etat)

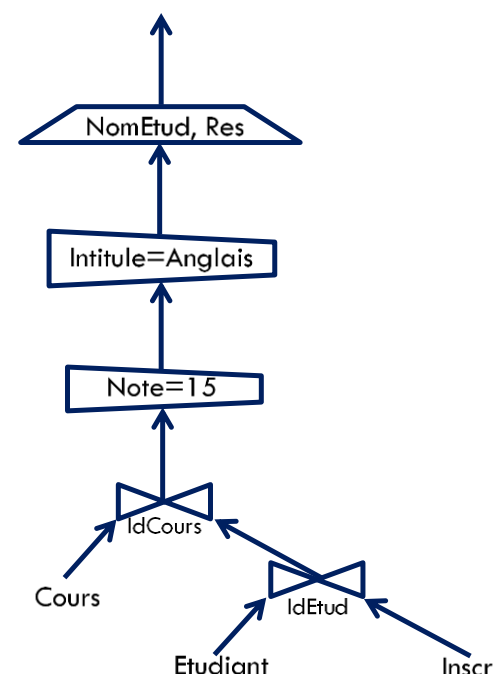
Formation (IdF, NomF, AdrF)

Cours (IdCours, IdF#, Intitule, Resp)

Inscr (IdCours#, IdEtud#, Note)

### 1. Analyser l'arbre suivant et essayer de l'améliorer :

- 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.



## 2. Traduire l'arbre en algèbre relationnel avant et après amélioration

### Avant amélioration

$\prod_{[NomEtd, Res,]} (\sigma_{[Intitule='anglais']} (\sigma_{[Note='1.5']} (\sigma_{[IdCours=NoCours]} (Cours \bowtie (\sigma_{[IdEtd=NoEtd]} (Etudiant \bowtie Inscr))))))$

### Après amélioration

$\prod_{[NomEtd, Res,]} (\sigma_{[Intitule='anglais' \text{ and } Note = '1.5']} (\prod_{[NoCours, Intitule]} (Cours) \bowtie (\prod_{[IdEtd, NomEtd, Res]} (Etudiant) \bowtie Inscr) ))$

## 3. Optimisez les requêtes suivantes :

- a.  $Q1 = \prod_{[NomF, AdrF]} \sigma_{[Formation.IdF = Cours.IdF]} (\sigma_{[Resp='Mohamed']} Cours \times Formation)$

$Q1 = \prod_{[NomF, AdrF]} \sigma_{[Formation.IdF = Cours.IdF]} (\sigma_{[Resp='Mohamed']} (\prod_{[IdF, Resp]} Cours) \times Formation)$

- b.  $Q2 = \text{SELECT NomF, Intitule, IdCours FROM Formation, Cours, Inscr, Etudiant WHERE (NomEtd = 'Amine')} \text{ AND (Cours.IdCours = Inscr.NoCours) AND (Formation.IdF = Cours.IdF) AND (Etudiant.IdEtd = Inscr.NoEtd)}$

$Q2 = \text{SELECT NomF, Intitule, IdCours FROM Formation, Cours, Inscr, (SELECT IdEtd FROM Etudiant WHERE NomEtd = 'Amine')} \text{ WHERE (Cours.IdCours = Inscr.NoCours) AND (Formation.IdF = Cours.IdF) AND (Etudiant.IdEtd = Inscr.NoEtd)}$

- c.  $Q3 = \text{SELECT NomEtd, Note, IdCours FROM Etudiant, Inscr, Cours, Formation WHERE (NomF = 'Informatique')} \text{ AND (Intitule='ADM_BDD')} \text{ AND (Cours.IdCours = Inscr.NoCours) AND (Formation.IdF = Cours.IdF) AND (Etudiant.IdEtd = Inscr.NoEtd)}$

$Q3 = \text{SELECT NomEtd, Note, IdCours FROM Etudiant, Inscr, (SELECT * FROM Cours WHERE Intitule='ADM_BDD'), (SELECT IdF, NomF FROM Formation WHERE NomF = 'Informatique')} \text{ WHERE (Cours.IdCours = Inscr.NoCours) AND (Formation.IdF = Cours.IdF) AND (Etudiant.IdEtd = Inscr.NoEtd)}$

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

**MATALLAH H**