

Corrigé TD 3

Exercice 01:

1) D'une façon générale : Soit R et S deux relations de degrés respectifs r et s , telles que $r > s$ et $S \neq 0$.

$$R \div S = \pi_{1,2,\dots,r-s}(R) - \pi_{1,2,\dots,r-s}((\pi_{1,2,\dots,r-s}(R) \times S) - R).$$

Dans notre cas: $R1 \div R2 = \pi_{1,2,\dots,r-s}(R1) - \pi_{1,2,\dots,r-s}((\pi_{1,2,\dots,r-s}(R1) \times R2) - R1).$

Donc on aura:

$$R1 \div R2 = \pi_A(R1) - \pi_A((\pi_A(R1) \times R2) - R1).$$

Explication par un exemple : Soit

R : table des affectations (N° _employés, N° _projet)

S : table des projets (N° _projet)

T : table résultante de la division

Donc : $T = R \div S$ est équivalent à :

$T_1 \leftarrow \pi_Y(R)$: T_1 = les N° employé de la table « affectations »

$T_2 \leftarrow \pi_Y((S \times T_1) - R)$: T_2 = les N° employé non affectés à au moins un projet
 : $S \times T_1$ = supposer que tous les employés ont été affectés à tous les projets
 : R = les vraies affectations
 : $(S \times T_1) - R$ = les affectations non faites

$T_3 \leftarrow T_1 - T_2$: liste des employés qui ont travaillé (été affectés) dans tous les projets

2) Dans le cas où certains SGBDR n'acceptent pas les mots clés **EXCEPT** et **INTERSECT**, on doit imaginer une alternative en SQL qui assure le même travail de ces deux mots clé :

$R1 \cap R2$ en SQL: **SELECT * FROM R1 INTERSECT SELECT * FROM R2**

```
SELECT * FROM R1
INTERSECT
SELECT * FROM R2
```

Equivalent à:

```
SELECT DISTINCT value FROM R1
WHERE value IN (SELECT value FROM R2);
```

A noter : la colonne « **value** » est à remplacer par la colonne de votre choix. La commande **DISTINCT** n'est pas obligatoire, mais est la plupart du temps utile pour éviter d'afficher plusieurs fois les mêmes valeurs.

Dans notre cas :

```
SELECT *
FROM R1
WHERE A IN (SELECT A FROM R2) AND
      B IN (SELECT B FROM R2);
```

Ou bien:

```
SELECT *
FROM R1
WHERE EXISTS (SELECT *
FROM R2
WHERE R1.A = R2.A AND R1.B = R2.B)
```

$R1 - R2$ en SQL: **SELECT * FROM R1 EXCEPT SELECT * FROM R2**

```
SELECT *
FROM R1
WHERE A NOT IN (SELECT A FROM R2) AND
      B NOT IN (SELECT B FROM R2);
```

Ou bien:

```
SELECT *
FROM R1 WHERE NOT EXISTS
(SELECT *
FROM R2
WHERE R1.A = R2.A AND R1.B = R2.B)
```

Exercice 02 :

- 1) Nom et prénom des clients qui ont acheté au moins un produit de la marque '**DELL**' .

```
SELECT DISTINCT C.NOM, PRENOM  
FROM CLIENT AS C, PRODUIT AS P, VENTE AS V  
WHERE (P.MARQUE='DELL') AND (P.NOPRODUIT=V.NOPRODUIT) AND (C.NOCLEINT=V.NOCLIENT) ;
```

- 2) Numéro, nom et prix des produits non vendus en **Avril 2010** (En utilisant la clause **NOT EXISTS**).

```
SELECT NOPRODUIT, NOM, PRIX  
FROM PRODUIT P  
WHERE NOPRODUIT NOT EXISTS (SELECT NOPRODUIT FROM VENTE V  
WHERE DATEVTE >= #01/04/2010# AND DATEVTE <= #30/04/2010# AND  
P.NOPRODUIT = V.NOPRODUIT)
```

- 3) Numéro du produit le plus vendus en **2010**.

```
SELECT NOPRODUIT, COUNT(*)  
FROM VENTE  
GROUP BY NOPRODUIT  
HAVING COUNT(*)= (SELECT MAX (nbr) FROM (Select Count (*) as Nbr From VENTE  
WHERE (DATEVTE>=#01/01/2010# And DATEVTE <= #31/12/2010#)  
GROUP BY NOPRODUIT));
```

- 4) Numéro des produits non vendus (En utilisant **OUTER JOIN**).

```
SELECT Produit.noproduit  
FROM VENTE FULL OUTER JOIN PRODUIT ON (VENTE.noproduit = PRODUIT.noproduit)  
WHERE VENTE.noproduit IS NULL ;
```

- 5) Numéro des clients qui ont acheté les mêmes marques que le client numéro **C100**.

```
SELECT V2.NOCLIENT  
FROM PRODUIT AS P, VENTE AS V1, VENTE AS V2  
WHERE ( (V1.NOCLIENT= C100 AND (V1.NOPRODUIT= P.NOPRODUIT)) AND  
(V2.NOPRODUIT=P.NOPRODUIT) AND (V1.MARQUE = V2.MARQUE AND V1.NOCLEINT<> V2.NOCLIENT) ) ;
```

Exercice 03: Soit le schéma relationnel suivant :

EMP (matrE, nom, poste, dateEmb, MatrSup, salaire, NoDept)

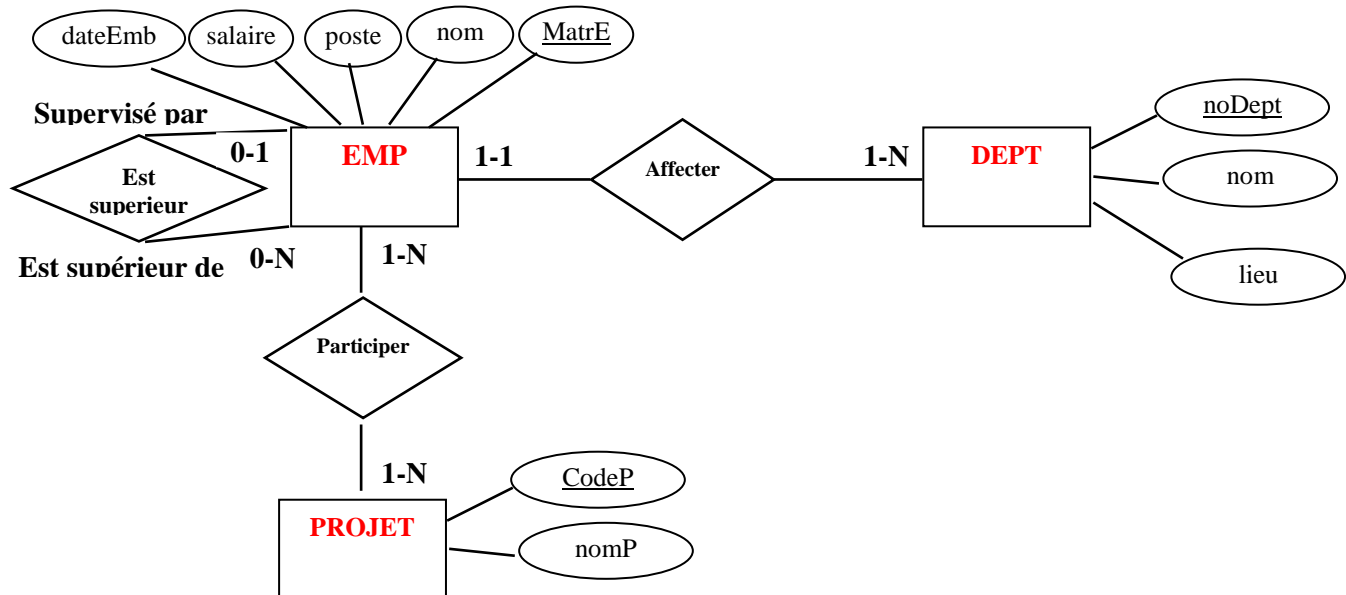
DEPT (noDept, nom, lieu)

PROJET (codeP, nomP)

PARTICIPATION (matrE, codeP)

a) Le diagramme EA qui correspond à ce schéma relationnel est le suivant :

Il faut remarquer que **MatrSup** est une clé étrangère, donc : (relation récursive).



b) Les deux clés candidates pour la relation **DEPT** sont: **noDept** ET **nom** (si les noms sont uniques)

c) Le résultat en français de chaque requête est comme suit :

Requête (1) : `SELECT * FROM EMP
WHERE matrSup = NULL;`

- La liste des employés qui n'ont pas de supérieur.

Requête (2) :

$\pi_{nom, codeP} ((\pi_{noDept, matrE} (EMP) \bowtie DEPT) \bowtie PARTICIPATION) \div \pi_{codeP} (PROJET)$

- Le Nom des départements qui ont participé à tous les projets.

Requête (3) :

`SELECT * FROM DEPT
WHERE Nodept not in (SELECT NoDept FROM EMP WHERE
dateEmb >= '01/10/2006');`

- La liste des départements qui n'ont pas embauchés depuis le 01/10/2006.

Requête (4) :

`SELECT * FROM DEPT
WHERE Nodept in (SELECT NoDept FROM EMP WHERE dateEmb
< '01/10/2006');`

- La liste des départements qui ont embauchés jusqu'au 01/10/2006.

Requête (5) : `SELECT * FROM EMP WHERE nom LIKE '%P4%';`

- La liste des employés dont le nom contient P4.

d) Les Requêtes en SQL :

1. Noms des employés qui ont un salaire supérieur au salaire de leur supérieur.

```
SELECT E1.NOM
FROM EMP E1, EMP E2
WHERE (E1.Matrsup= E2.MatrE) AND (E1.SALAIRE > E2.SALAIRE);
```

2. Numéros des départements dont tous les employés sont des ingénieurs (**poste = 'Ingenieur'**).

```
SELECT E1.NODEPT
FROM EMP AS E1
WHERE (E1.POSTE = 'Ingenieur') and (E1.POSTE = ALL (SELECT Distinct E2.POSTE
                                                    FROM EMP AS E2
                                                    WHERE E1.NODEPT=E2.NODEPT));
```

3. Numéros des départements qui ont plus de deux ingénieurs.

```
SELECT NODEPT, COUNT(*)
FROM EMP
WHERE POSTE = 'INGENIEUR'
GROUP BY NODEPT
HAVING COUNT(*) > 2 ;
```

4. Nom du département qui a le plus grand nombre des employés.

```
SELECT Dept.nom, COUNT(*) AS nbr_emp
FROM emp, dept
Where emp.nodept = Dept.nodept
GROUP BY Dept.nom
HAVING Count(*) = ( SELECT MAX (nbr) FROM( SELECT COUNT(*) as nbr FROM emp Group By nodept));
```

5. Nom des départements qui ont des ingénieurs (**poste = 'ingenieur'**) et des comptables (**poste = 'comptable'**).

```
SELECT DISTINCT D1.NOM
FROM EMP AS E1, EMP AS E2, DEPT AS D1
WHERE ( (D1.NODEPT= E1.NODEPT) AND ( D1.NODEPT= E2.NODEPT ) AND
(E1.POSTE = 'ingenieur' AND E2.POSTE = 'comptable') );
```

6. La liste des employés qui ne participent à aucun projet.

```
SELECT *
FROM EMP
WHERE MATRE NOT IN (SELECT MATRE
                    FROM PARTICIPATION);
```

7. Codes des projets auxquels participe au moins un employé du **département 10**.

```
SELECT Codep
FROM Participation AS P, Emp AS E
WHERE (P.noemp=E.noemp AND nodept='10');
```

8. Matricule des employés qui ont au moins un subordonné qui participe à chacun des projets (pas nécessairement le même subordonné).

```
SELECT DISTINCT Matrsup
FROM EMP as E1 INNER JOIN PARTICIPATION as P1 ON E1.Matrsup=P1.noemp
WHERE codep IN (SELECT codep
                FROM EMP as E2 INNER JOIN PARTICIPATION as P2 ON E2.noemp=P2.noemp
                WHERE E2.noemp=E1.Matrsup);
```

9. Nom des employés qui ont le plus grand salaire (n'utiliser ni la fonction de synthèse max ni min).

```
SELECT nom
FROM emp
WHERE salaire NOT IN (SELECT e1.salaire
                      FROM emp e1, emp e2
                      WHERE e1.salaire<e2.salaire);
```

10. Nom des départements qui participent dans tous les projets.

```
SELECT D.NOM
FROM DEPT AS D
WHERE NOT EXISTS
    (SELECT *
     FROM EMP AS E
     WHERE NOT EXISTS
        (SELECT *
         FROM PROJET AS PR
         WHERE NOT EXISTS
            (SELECT *
             FROM PARTICIPATION AS P
             WHERE P.MATRE= E.MATRE
                   AND P.CODEP=PR.CODEP
                   AND D.NODEPT <> E.NODEPT));
```

Donc, le résultat est composé du Nom de chaque **département** pour le quel il n'existe pas un **employé** pour le quel il n'existe pas un **projet** pour lequel il n'existe pas une **participation**.

11. La fonction qui se trouve dans plus de deux départements.

```
SELECT DISTINCT e1.POSTE
FROM EMP AS e1, EMP AS e2, EMP AS e3
WHERE e1. POSTE = e2. POSTE AND e2. POSTE = e3. POSTE
      AND e1.nodept <> e2.nodept AND e1.nodept <> e3.nodept AND e2.nodept <> e3.nodept;
```

12. Numéro des départements qui ont des ingénieurs et ils n'ont pas des secrétaires.

```
SELECT nodept
FROM EMP AS E1
WHERE POSTE ='Ingénieur' AND nodept NOT IN (SELECT nodept
                                           FROM EMP E2
                                           WHERE POSTE = 'Secrétarie' AND E1.nodept=E2.nodept);
```

13. Nom des départements qui participent dans les mêmes projets que département **numéro 2**.

```
SELECT DISTINCT D.nom
FROM EMP AS E, DEPT AS D, PARTICIPATION AS P
WHERE E.nodept <> '02' AND D.nodept= E.nodept AND E. MATRE =P. MATRE
      AND P.codep IN (SELECT codep
                     FROM EMP E, PARTICIPATION P
                     WHERE E. MATRE =P. MATRE AND E.nodept='02');
```

14. Numéro des départements qui ont plus de deux **ingénieurs** qui participent dans le projet **codeP=10**.

```
SELECT NODEPT, COUNT(*) as Nbr_Ing
FROM EMP, PARTICIPATION
WHERE (EMP.POSTE = 'Ingénieur' AND CODEP='10' AND EMP.MATRE=PARTICIPATION.MATRE)
GROUP BY NODEPT
HAVING COUNT(*) >= 2 ;
```