

Suite correction exercices TD

Correction exercice 3 « Employé » (TD5) :

c. Employé : description d'un employé travaillant sur un projet d'un laboratoire.

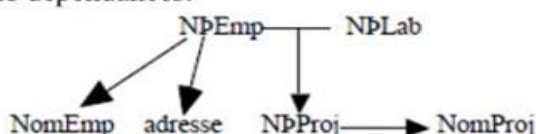
Employé (N°Emp, N°Lab, N°Proj, NomEmp, NomProj, adresse)

Avec les dépendances fonctionnelles suivantes :

- (N°Emp, N°Lab) → N°Proj, NomProj, NomEmp
- N°Emp → NomEmp
- N°Emp → adresse
- N°Proj → NomProj

a) Il y a des redondances pour les attributs NomEmp, adresse et NomProj.

b) Graphe minimum des dépendances:



c) L'identifiant est composé de (N°Emp + N°Lab).

d) La relation est en première forme normale, mais pas en seconde forme normale, car les attributs NomEmp et adresse dépendent d'un morceau d'identifiant (et non pas de l'identifiant entier).

e) La relation peut être décomposée en:

- Employé (N°Emp, NomEmp, adresse)
- Projet (N°Proj, NomProj)
- ELP (N°Emp, N°Lab, N°Proj)

Correction exercice 4 « Client » (TD5):

Soit la relation : Client (numcli, codepostal, nom, prenom, tel, ville)

➤ **1FN** : La relation « Client » est en 1FN puisqu'elle ne contient aucun attribut composé ou multivalué.

➤ **2FN** : La relation « Client » n'est pas en 2FN étant donné que la DF :

numcli, codepostal → ville n'est pas élémentaire (seul le code postal suffit pour déterminer la ville).

Pour éviter ce problème, on peut réduire la clé au numéro du client. Le schéma de la relation Client devient ainsi :

Client1 (numcli, nom, prenom, tel, ville, # codepostal)

➤ **3FN** : La relation « Client1 » n'est pas en 3FN étant donné que la DF numcli → ville n'est pas directe (numcli → codepostal et codepostal → ville).

Pour faire passer la relation « Client1 » en 3FN, il faut la décomposer en deux, ce qui donne les deux schémas suivants :

Client2 (numcli, nom, prenom, tel, # codepostal)

Adresse (codepostal, ville)

➤ **BCFN** : les deux relations « Client2 » et « Adresse » sont en BCFN (aucun attribut autre que la clé n'est déterminant)

Correction exercice 3 TD6 :

Exprimer en SQL les requêtes suivantes :

1. Date de naissance et l'adresse de Taha Lamharchi

```
SELECT DateNaissance, Adresse FROM Employe  
WHERE Nom='Lamharchi' AND Prenom ='Taha'
```
2. Nom et adresse des employés qui travaillent au département de recherche.

```
SELECT E.Nom, E.Adresse FROM Employe as E, Departement as D  
WHERE E.N_dep=D.N_dep AND NomD='recherche'
```
3. Nom et Prénom des employés dont le supérieur est Taha Lamharchi

```
SELECT Nom , Prenom FROM Employe  
WHERE superieur=(SELECT Matricule FROM Employe WHERE Nom='Lamharchi'  
AND Prenom ='Taha')
```
4. Nom des employés qui travaillent plus de 10heures sur un projet à Guelmim

```
SELECT E.Nom FROM Employe as E, Travaille as T , Projet P  
WHERE E.Matricule=T.Matricule AND T.N_proj=P.N_proj AND T.heures >=10 AND  
P.Lieu='Guelmim'
```
5. Nom et prénom des employés qui ne travaillent sur aucun projet.

```
SELECT Nom , Prenom FROM Employe  
WHERE Matricule NOT IN (SELECT Matricule FROM Travaille)
```
6. Nom des employés qui ne travaillent pas sur un projet à Guelmim

```
SELECT Nom FROM Employe  
WHERE Matricule NOT IN(SELECT T.Matricule FROM Travaille as T, Projet as P  
WHERE T.N_proj=P.N_proj AND P.Lieu='Guelmim')
```
7. Nom des projets sur lesquelles travaillent Taha Lamharchi et Dounia Mahmoud

```
SELECT T.N_proj FROM Travaille as T,Employe as E  
WHERE T.Matricule=E.Matricule AND E.Nom='Lamharchi' AND E.Prenom='Taha'  
INTERSECT  
SELECT T.N_proj FROM Travaille as T,Employe as E  
WHERE T.Matricule=E.Matricule AND E.Nom='Mahmoud' AND E.Prenom='Dounia'
```