

A l'aide d'un script que vous aurez téléchargé, créez une base de données contenant 2 tables :

- Une table des employés,
- Une table des services dans lesquels travaillent les employés

Ouvrez ensuite une fenêtre SQL puis réalisez les requêtes suivantes :

- Vous êtes recruté au sein du service numéro 50. Ecrivez la requête permettant de vous y insérer en choisissant des valeurs réalistes.
- A l'aide de phpMyAdmin ajoutez un champ « codeNat » à la table « EMPLOYE » sur 2 caractères alphanumériques. Valorisez à l'aide de 3 requêtes les valeurs de ce champ à « fr » pour tous les employés sauf pour Barbarin qui est belge (« be ») et Lechif qui est italien (« it »).

```
Update EMPLOYE set codeNat='fr';
Update EMPLOYE set codeNat='it' where nomEmploye = 'Lechif';
Update EMPLOYE set codeNat='be' where nomEmploye = 'Barbarin';
```

- A l'aide de phpMyAdmin, créez une table NATIONALITE avec 2 champs « codeNat » et libNat » et remplissez la table avec les nationalités nécessaires à l'aide de 3 requêtes.

```
Insert into NATIONALITE VALUES('fr','France');
Insert into NATIONALITE VALUES('it','Italie');
Insert into NATIONALITE VALUES('be','Belgique');
Update EMPLOYE set codeNat='it' where nomEmploye = 'Lechif';
Update EMPLOYE set codeNat='be' where nomEmploye = 'Barbarin';
```

- Un nouveau service a été créé : Le service ACTUARIAT. On ne sait pas encore où il sera localisé. Mettez à jour la base de données.

```
INSERT INTO SERVICE
(numService, nomService)
VALUES ( 60, 'Actuariat' )
```

- Mr Amin, comptable, a été embauché au service « Comptabilité » (code service=60) à Gif sur Yvette le 15/10/08 à un salaire annuel de 20000 euros. Il aura le matricule 25381 et a pour responsable le président. Ecrire les requêtes permettant de mettre à jour la base de données.

```
insert into service
VALUES (70,'Comptabilite' , 'Gif-sur-Yvette');
insert into employe
VALUES (25381,'Amin','Comptable','2008-10-08', 20000,NULL,60,16712);
```

- Suite aux bons résultats de l'entreprise, le président a décidé d'augmenter son salaire de 140% et le salaire de l'ensemble des employés de 2%. Ecrire les requêtes permettant de mettre à jour la base de données.

```
update employe
set salaire = salaire * 1.02
where fonction <> 'President';
```

```
update employe
set salaire = salaire * 2.4
where fonction = 'President';
```

- L'ingénieur « Dupont » a démissionné; il ne fait donc plus partie des effectifs. Supprimez le de la base de données.

```
delete from employe
where nomemploye = 'Dupont';
```

- Par 2 méthodes différentes (SQL1 et SQL2), affichez les noms et la date d'embauche de tous les employés travaillant à Metz.

```
Select nomEmploye , dateEmbauche from EMPLOYE , SERVICE
where numServEmploye = numService
and lieu = 'Metz';
```

```
Select nomEmploye , dateEmbauche from EMPLOYE
JOIN SERVICE
ON numServEmploye = numService
WHERE lieu = 'Metz';
```

- Affichez le produit cartésien des tables EMPLOYE et SERVICE avec 3 méthodes différentes.

```
Select * from EMPLOYE , SERVICE
```

```
Select * from EMPLOYE
JOIN SERVICE
```

```
Select * from EMPLOYE
CROSS JOIN SERVICE
```

- Par 3 méthodes différentes (SQL1, SQL2 et jointure naturelle) affichez les noms d'employés et leur nationalité en toutes lettres

```
Select nomEmploye , libNat from EMPLOYE E, NATIONALITE N
where E.codeNat = N.codeNat;
```

```
Select nomEmploye , libNat from EMPLOYE
NATURAL JOIN NATIONALITE;
```

```
Select nomEmploye , libNat from EMPLOYE E
JOIN NATIONALITE N on E.codeNat = N.codeNat;
```

```
Select nomEmploye , libNat from EMPLOYE E
INNER JOIN NATIONALITE N on E.codeNat = N.codeNat;
```

- Affichez les nationalités des employés travaillant à Metz par 2 méthodes (SQL1 et SQL2)

```
Select nomEmploye , libNat from EMPLOYE E, NATIONALITE N , SERVICE
where E.codeNat = N.codeNat
and numServEmploye = numService
and lieu = 'Metz';
```

```
Select nomEmploye , libNat from SERVICE
JOIN EMPLOYE
NATURAL JOIN NATIONALITE
ON numServEmploye = numService
where lieu = 'Metz';
```

```
Select nomEmploye , libNat from SERVICE
JOIN EMPLOYE
ON numServEmploye = numService
NATURAL JOIN NATIONALITE
where lieu = 'Metz';
```

```
Select nomEmploye , libNat from SERVICE
JOIN EMPLOYE
ON numServEmploye = numService
```

```
JOIN NATIONALITE
ON NATIONALITE.codeNat = EMPLOYE.codeNat
where lieu = 'Metz';
```

- A l'aide d'une jointure, affichez les noms d'employés et leur supérieur.

```
Select e.nomEmploye , s.nomEmploye from EMPLOYE e , EMPLOYE s
WHERE e.matSuperieur = s.mat
```

```
Select e.nomEmploye , s.nomEmploye from EMPLOYE e JOIN EMPLOYE s
ON e.matSuperieur = s.mat
```

- A l'aide d'une jointure, affichez les noms d'employés et leur supérieur mais affichez aussi ceux qui n'ont pas de supérieur

```
Select e.nomEmploye , s.nomEmploye AS chef from EMPLOYE e
LEFT OUTER JOIN EMPLOYE s
ON e.matSuperieur = s.mat
```

- A l'aide d'une jointure, affichez les noms d'employés et leur supérieur mais affichez aussi ceux qui n'encadrent personne

```
Select e.nomEmploye , s.nomEmploye AS chef from EMPLOYE e
RIGHT OUTER JOIN EMPLOYE s
ON e.matSuperieur = s.mat
```

- A l'aide d'une jointure, affichez les noms d'employés et leur supérieur mais affichez aussi ceux qui n'encadrent personne ainsi que ceux qui n'ont pas de supérieur

A chercher...