

iiExercices Requêtes SQL

Une entreprise qui fournit des cours en **Formation Continue** a implémenté ses sessions de formation à l'aide des deux tables. La première table se nomme **coursFC** et contient un numéro de cours (qui sert de clé primaire), un titre de cours, son niveau, une indication numérique de session. On pourra supposer qu'une session correspond à un semestre. Voici les instructions SQL qui ont permis de créer la table **coursFC** :

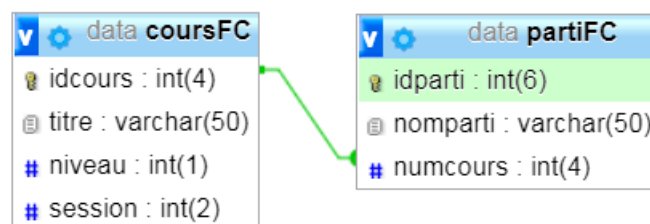
```
DROP TABLE IF EXISTS coursFC ;
CREATE TABLE coursFC (
    idcours INT(4) NOT NULL ,
    titre VARCHAR(50) NOT NULL ,
    niveau INT(1) DEFAULT 1 NOT NULL ,
    session INT(2) DEFAULT 1 NOT NULL ,
    PRIMARY KEY (idcours)
) ;
```

La deuxième table se nomme **partiFC** et recense les participants aux Formations Continues. Elle comporte un numéro de participant (clé primaire), le nom du participant et le numéro du cours qu'il doit suivre. Ce numéro doit correspondre, bien sûr, au champ idcours de la table **coursFC**. Voici les instructions SQL qui ont permis de créer la table **partiFC** :

```
DROP TABLE IF EXISTS partiFC ;

CREATE TABLE partiFC (
    idparti INT(6) NOT NULL ,
    nomparti VARCHAR(50) NOT NULL ,
    numcours INT(4) NOT NULL ,
    PRIMARY KEY (idparti)
) ;
```

Les tables sont présentes dans la base data que vous devrez importer sur le serveur mysql de votre machine virtuelle. Elles se nomment **coursFC** et **partiFC**.



Commandes tapées:

```
SELECT * FROM `coursFC`, `partiFC` WHERE `idcours` = `numcours`
```

JOINTURE (Utiliser deux tables en meme temps):

```
SELECT * FROM `coursFC` join `partiFC` ON `idcours` = `numcours`
```

```
SELECT * FROM `coursFC` left join `partiFC` ON `idcours` = `numcours`
```

```
SELECT count(*) FROM `coursFC` left join `partiFC` ON `numcours` = NULL
```

```
SELECT titre FROM `coursFC` left join `partiFC` ON `numcours` is NULL
```

```
SELECT nomparti, c1.titre, c1.niveau FROM partiFC, coursFC c1, coursFC c2 where c1.idcours = numcours and c1.titre = c2.titre and c1.niveau = 1 and c2.niveau = 2
```

```
SELECT * FROM `coursFC`, `partiFC` WHERE `idcours`= `partiFC`.`numcours` and `partiFC`.`nomparti` = 'Dupont'
```

```
SELECT nomparti, c1.titre FROM partiFC, coursFC c1, coursFC c2 WHERE c1.idcours=numcours AND c1.titre=c2.titre AND c1.niveau=1 AND c2.niveau =2;
```

```
SELECT count(c1.titre) as effectif FROM partiFC, coursFC c1, coursFC c2 WHERE c1.idcours=numcours AND c1.titre=c2.titre AND c1.niveau=1 AND c2.niveau
```

Questions

Écrire les requêtes SQL permettant de répondre aux questions suivantes

1. Combien y a-t-il de cours qui traitent de C ? **2**

```
select count(*) as nb from `coursFC` where `titre` like '% C%'
```

2. Combien y a-t-il de personnes qui suivent des cours de session 1 ?

```
SELECT count(*) as nb FROM `coursFC`,`partiFC` WHERE `idcours` = `numcours` and `session` = 1
```

3. Y a-t-il des cours qui ne sont suivis par personne (quelle que soit la session) ?

```
SELECT count(*) FROM `coursFC` left join `partiFC` ON `numcours` = NULL
```

4. Quelles sont les personnes ayant suivi un cours de niveau 1 à qui on pourrait proposer le même cours au niveau 2 ?

```
SELECT nomparti, c1.titre, c1.niveau FROM partiFC, coursFC c1, coursFC c2 where c1.idcours = numcours
```

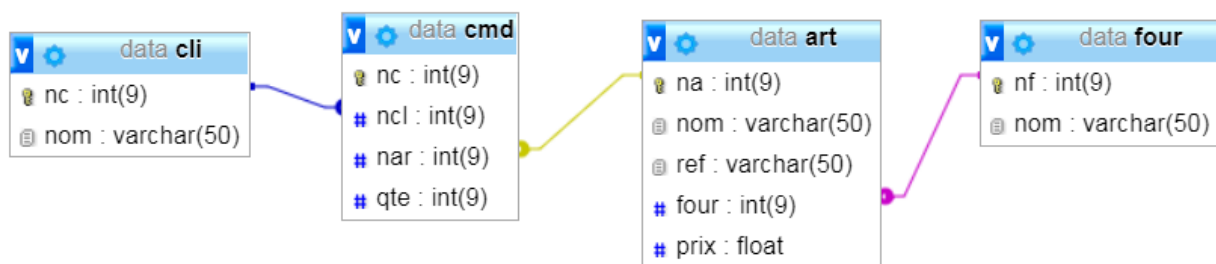
5. Donner le noms des cours et les effectifs correspondants pour la question précédente. On affichera par effectif décroissant.

```
SELECT count(c1.titre) as effectif, c1.titre FROM partiFC, coursFC c1, coursFC c2 WHERE
c1.idcours=numcours AND c1.titre=c2.titre AND c1.niveau=1 AND c2.niveau= 2group by
c1.titre order by effectif desc
```

Exercice 2 la commande d'articles en ligne.

Les 4 tables correspondent respectivement aux fournisseurs, clients, articles et commandes dans l'exemple de la commande d'articles en ligne. Elles sont disponibles dans la base data. Elles se nomment four, cli, art et cmd.

La table des fournisseurs contient un numéro de fournisseur et un nom de fournisseur ; la table des clients contient un numéro de client et un nom de client ; la table des articles contient un numéro d'article, un nom d'article, une référence d'article (texte), le numéro du fournisseur et le prix unitaire de l'article ; la table des commandes contient un numéro de commande, un numéro de client, un numéro d'article, et la quantité commandée.



```
select cli.nom, art.nom from cmd, art,cli where cmd.nc = 4 and cli.nc = cmd.ncl
and art.na = cmd.nar
```

Questions

1. Quel est le nom de la personne et le nom de l'article pour la commande numéro 4 ?

```
select cli.nom, art.nom from cmd, art,cli where cmd.nc = 4 and cli.nc = cmd.ncl
and art.na = cmd.nar
```

2. Comment afficher en clair pour chaque commande le numéro de commande, le nom de l'acheteur, le nom de l'article et celui du fournisseur ? On fournira les résultats triés par numéro de commande.

```
select cli.nom, art.nom, cmd.nc, four.nom from four, art,cmd, cli where
cli.nc = cmd.ncl and art.na = cmd.nar and four.nf = art.four order by
cmd.nc
```

3. Quels sont les clients ayant commandé au moins une fois ? On fournira les résultats triés par nombre de commandes puis par nom de client si même nombre de commandes.

4. Quels sont les clients qui n'ont rien commandé ? On fournira les résultats triés par nom de client.
5. Comment afficher la liste de tous les articles (même ceux qui n'ont jamais été commandés) avec le nombre de fois où chaque article a été commandé (ou 0 s'il ne l'a jamais été) ? On fournira les résultats triés par nombre décroissant d'articles commandés.
6. Quel est le nombre de commandes passées pour chaque fournisseur ? On fournira les résultats triés par nombre décroissant de commandes.

Attention, cette question est difficile car un fournisseur n'a eu aucune commande.