

Devoir 3 Informatique Commune : SQL, solutions

1. Donner les **nom** et **prenom** de tous les clients de la base habitant à Paris.

```
SELECT nom, prenom
FROM client
WHERE ville = "PARIS"
```

en supposant que **PARIS** soit rentré en majuscule dans la base.

2. Donner les **idop** des opérations de montant négatif (débit) sur le compte dont l'**idcompte** vaut 1.

```
SELECT idop
FROM operations
WHERE montant < 0
AND idcompte = 1
```

3. Donner les **idproprietaire** des personnes ne possédant aucun Livret A (mais ayant un compte dans la banque). On peut utiliser la commande **EXCEPT** pour faire la différence entre tous les propriétaires de compte et ceux possédant un livret A

```
SELECT idproprietaire
FROM compte
EXCEPT
SELECT idproprietaire
FROM compte
WHERE type = "Livret A"
```

ou encore la commande **NOT IN** dont le contenu constitue une sous-requête

```
SELECT idproprietaire
FROM compte
WHERE idproprietaire NOT IN (SELECT idproprietaire
                              FROM compte
                              WHERE type = "Livret A")
```

4. Donner le solde (= somme de toutes les opérations effectuées) de chaque compte : on fera apparaître un tableau à deux colonnes avec les attributs **idcompte** et **solde** :

```
SELECT idcompte, SUM(montant) AS solde
FROM operations
GROUP BY idcompte
```

5. Donner, pour chaque personne définie par son couple **nom**, **prenom**, la liste de ses comptes (identifiés par leur **idcompte**).

```
SELECT nom, prenom, idcompte
FROM client JOIN compte ON client.idclient = compte.idproprietaire
```

ou de manière équivalente :

```
SELECT nom, prenom, idcompte
FROM client, compte
WHERE client.idclient = compte.idproprietaire
```

6. Donner la liste des **idcompte** qui ont pour propriétaire une personne dont le nom est "Dumbledore".

```
SELECT idcompte
FROM client JOIN compte ON client.idclient = compte.idproprietaire
WHERE client.nom = "Dumbledore"
```

ou de manière équivalente :

```
SELECT idcompte
FROM client, compte
WHERE client.idclient = compte.idproprietaire
AND client.nom = "Dumbledore"
```

7. Donner pour chaque client identifié par ses **nom** et **prenom** le nombre de comptes qu'il possède.

```
SELECT cl.nom, cl.prenom, COUNT(idcompte)
FROM client cl
JOIN compte co ON cl.idclient = co.idproprietaire
GROUP BY co.idproprietaire
```

ou de manière équivalente

```
SELECT client.nom, client.prenom, COUNT(idcompte)
FROM client, compte
WHERE client.idcompte = compte.idproprietaire
GROUP BY idproprietaire
```

Noter que le `GROUP BY idproprietaire` est préférable à `GROUP BY nom, prenom` que l'énoncé semblait suggérer car `nom, prenom` n'est pas une clef.

8. Donner la liste des opérations (`montant` et `informations`) sur chaque compte de "Harry Potter", figuré par son `idcompte`

```
SELECT compte.idcompte, montant, informations
FROM client JOIN compte ON client.idclient = compte.idproprietaire
      JOIN operations ON operations.idcompte = compte.idcompte
WHERE client.prenom = "Harry" AND client.nom = "Potter"
```

9. Donner pour chaque type de compte la somme totale d'argent stockée dans la banque.

```
SELECT type, SUM(montant)
FROM compte JOIN operations ON compte.idcompte = operations.idcompte
GROUP BY type
```

ou par produit cartésien

```
SELECT type, SUM(montant)
FROM compte, operations
WHERE compte.idcompte = operations.idcompte
GROUP BY type
```

10. Donner l'`idcompte` du compte dont la moyenne des opérations de débit est la plus importante.

```
select idcompte avg(montant)
from operations
where montant < 0
group by idcompte
having avg(montant) = (
  select min(moyenne)
  from (
    select avg(montant) as moyenne
    from operations
    where montant < 0
    group by idcompte
  )
)
```