

Chap 2 : Révisions SQL, solutions

Requêtes simples

1. Donner les nom et prénom de tous les clients de la base.

```
SELECT nom, prenom FROM client
```

2. Donner les nom et prénom de tous les clients de la base habitant à Paris.

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

N.B. Pour la chaîne de caractère "PARIS" les majuscules/minuscules comptent donc il faudrait savoir comment les villes sont rentrées dans la base.

3. Donner les identifiants de tous les comptes de type 'Livret A'.

```
SELECT idcompte FROM compte WHERE type="Livret A"
```

4. Donner les idop de type débit sur le compte dont l'idcompte vaut 1.

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

5. Donner les idproprietaire des personnes possédant un Livret A.

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

6. Donner les idproprietaire des personnes ne possédant aucun Livret A.

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

A l'écrit on peut aussi donner un nom à la requête de la question 4 et écrire ;

```
R4=SELECT idproprietaire FROM compte WHERE type="Livret A"  
SELECT idproprietaire FROM Compte EXCEPT R4
```

Utilisations de fonctions

7. Donner le solde (= somme de toutes les opérations effectuées) de chaque compte.

```
SELECT SUM(montants) AS Solde,idcompte FROM compte GROUP BY idcompte
```

Attention : toujours mettre dans le SELECT l'attribut sur lequel vous faites le GROUP BY, ici le `idcompte`, pour qu'on puisse savoir ici à quel idcompte correspond chaque somme calculée par regroupement `idcompte` par `idcompte`

8. Donner le solde de chaque compte, uniquement pour les comptes débiteurs.

Là cela devient subtil, car on ne peut savoir qu'un compte est débiteur que si on a déjà calculé la somme, et le WHERE s'effectue toujours avant (on dit en amont) du calcul du SUM. Aussi a-t-on besoin d'une sous-requête, qui est exactement la requête précédente.

```
R7=SELECT SUM(montants) AS Solde,idcompte FROM compte GROUP BY idcompte
```

```
SELECT Solde, idcompte FROM R7 WHERE Solde <0
```

N.B. Il existe une autre solution pour faire cela avec une seule requête, avec le HAVING (sélection en aval), mais cet opérateur n'est pas au programme.

```
SELECT idcompte, SUM(montant) AS Solde FROM compte GROUP BY idcompte  
HAVING Solde < 0
```

9. Donner le solde des comptes d'identifiant compris entre 1 et 10 qui sont débiteurs.

```
R=SELECT SUM(montants) AS Solde,idcompte FROM compte
WHERE idcompte >=1 AND idcompte <=10 GROUP BY idcompte
```

```
SELECT Solde, idcompte FROM R WHERE Solde <0
```

10. Calculer le nombre moyen de comptes possédé par un client.

On peut commencer par calculer le nombre de compte par client.

```
R=SELECT Count(idcompte) AS NbCpt, idproprietaire FROM compte GROUP BY idproprietaire
```

Puis à partir de ce tableau obtenu calculer la moyenne

```
SELECT AVG(NbCpt) FROM R
```

qui renvoie un nombre.

Mise en jeu d'une ou plusieurs jointures

11. 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
```

12. Donner la liste des idcompte qui ont pour propriétaire une personne dont le prénom est 'Marie'.

```
SELECT idcompte FROM client JOIN compte ON client.idclient = compte.idproprietaire
WHERE client.prenom="Marie"
```

13. Donner la liste des opérations (montant et informations) sur chaque compte de 'Sylvie Moulin'. Ici, le nom est dans client, les comptes dans compte, et les opérations dans opérations, donc besoin d'une jointure sur des trois tables.

```
SELECT idcompte, montant, informations
FROM client JOIN compte ON client.idclient = compte.idproprietaire
JOIN operations ON operations.idcompte = compte.idcompte
WHERE client.prenom = 'Sylvie' AND client.nom = 'Moulin'
```

14. Donner les nom et prénom des clients ayant fait au moins une opération de débit sur l'un de leurs comptes. On a besoin de la même jointure qu'à la question précédente, seule la condition de sélection et les attributs de projections changent.

```
SELECT client.nom, client.prenom
FROM client JOIN compte ON client.idclient = compte.idproprietaire
JOIN operations ON operations.idcompte = compte.idcompte
WHERE operation.montant<0
```

15. Donner pour chaque client identifié par ses nom et prenom le nombre de comptes qu'il possède. Indication : il s'agit ici de compter le nombre d'idcompte distinct associé à chaque client.

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

16. Donner pour chaque type de compte la somme totale disponible.

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

Tables imbriquées : sous-requêtes

En fait, on a déjà fait des sous-requêtes aux questions 8 à 10 ! Ici le résultat de la sous-requête est un nombre.

17. Donner l'idproprietaire possédant le compte dont l'idcompte a le plus grand numéro. Un sous-requête pour déterminer l'idcompte de plus grand numéro

```
SELECT idproprietaire
FROM compte
WHERE idcompte = (SELECT MAX(idcompte) FROM compte)
```

18. On a déjà calculé la moyenne demandée à la question 10, reprenons cette question

```
R=SELECT Count(idcompte) AS NbCpt, idproprietaire FROM compte GROUP BY idproprietaire
```

Puis à partir de ce tableau obtenu on a vu comment calculer la moyenne :

```
SELECT AVG(NbCpt) FROM R
```

Enfin la requête demandée :

```
SELECT idproprietaire FROM R WHERE NbCpt < (SELECT AVG(NbCpt) FROM R)
```