

## Devoir Maison « Reporting Consultants »

Affecté à la cellule « analyse » d'une importante société de service parisienne, vous devez produire des rapports réguliers sur les activités des consultants affectés à des projets. Un consultant peut être affecté à plusieurs projets (voire à aucun). Pour chaque projet, on lui affecte un nombre d'heures hebdomadaires (« nbHeuSem ») pendant une certaine période. Chaque projet a son budget (« budP », indiqué en K€) et un consultant qui le dirige. La réalisation d'un projet demande des compétences qui sont décrites, avec le niveau requis (« novice », « expert »...), dans la relation « CompProjet ». Les compétences déclarées par les consultants sont, quant à elles, décrites dans la relation « CompEquipe », toujours avec leur niveau respectif.

**Clients** (ncli, nomCli, emailCli, telCli, adrCli, vilCli)

**Consultants** (nc, emailCo, nomCo, bureauCo, datEmbCo, adrCo, vilCo)

**Projets** (np, ncli #, nc #, intitule, budP, datDebP, datFinP, descrP)

**Equipes** (nc #, np #, datDebEq, datFinEq, nbHeuSem)

**CompProjet** (np #, comp, niveau)

**CompEquipe** (nc #, comp, niveau)

*PS : durée approximative 2h (2h ÷ 10 questions à faire = ± 12min par question)*

### Questions algèbre et SQL

Répondre aux questions suivantes utilisant l'algèbre relationnelle et le langage SQL :

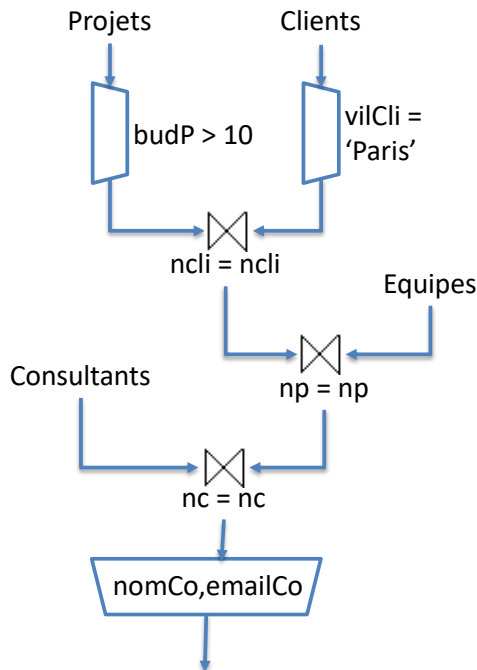
- 1) Afficher les noms et les emails des consultants travaillant sur des projets de budget supérieur 10K€ pour des clients situés à Paris. (2,0 pts)
- 2) Afficher les compétences nécessaires à la fois aux projets démarrant en 2018 et à ceux qui ont démarré en 2008. (2,0 pts)
- 3) Afficher le nom et l'email du consultant ayant dirigé le projet démarré en 2018 ayant le budget le plus élevé. (2,0 pts)
- 4) Afficher les intitulés et les budgets des projets ayant impliqué tous les consultants embauchés en 2018. (2,0 pts)
- 5) Afficher les compétences demandées par des projets démarrant en 2019 pour lesquelles nous n'avons aucun consultant ayant déclaré cette compétence (quel que soit son niveau). (2,0 pts)
- 6) Afficher les noms des consultants parisiens ayant toutes les compétences requises (avec les niveaux demandés) pour les projets démarrant en 2018. (3,0 pts)
- 7) Afficher les projets (intitulés et noms des clients associés) démarrant en 2018 de plus de 20k€ de budget dont les équipes ne comptent que des consultants experts (consultants ayant les compétences au niveau expert). (3,0 pts)

## Questions SQL uniquement

Répondre aux questions suivantes uniquement en langage SQL.

- 8) Afficher le nombre total d'heures hebdomadaires dépensées par projet démarrant en 2018. (1,0 pt)
- 9) Afficher les intitulés et les noms des clients des projets dont l'intitulé commence par « Management » impliquant plus de 10 consultants. (1,5 pts)
- 10) Afficher les noms et les numéros des clients qui débiteront au moins 3 projets en 2019 avec un budget supérieur au budget moyen des projets démarrés en 2018. (1,5 pts)

1) Afficher les noms et les emails des consultants travaillant sur des projets de budget supérieur 10K€ pour des clients situés à Paris.

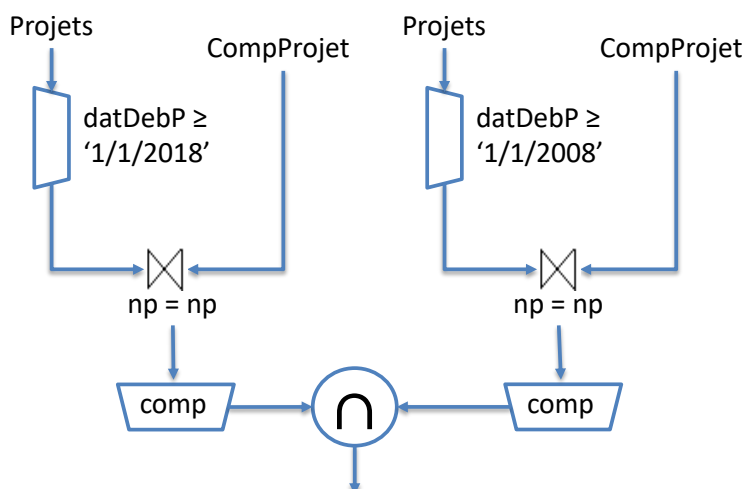


```

SELECT Co.nomCo, Co.emailCo
FROM Consultants as Co, Equipes as E,
Projets as P, Clients as Cl
WHERE Co.nc = E.nc
And E.np = P.np
And P.ncli = Cl.ncli
And P.budP > 10
And Cl.vilCli = 'Paris'
    
```

Dans cette question, attention à bien faire ses jointures, à les faire sur les bons attributs et à ne pas oublier les restrictions.

2) Afficher les compétences nécessaires à la fois aux projets démarrant en 2018 et à ceux qui ont démarré en 2008.



C'est la **compétence** qui doit être présente dans les deux ensembles.

Attention donc à faire l'**intersection** sur l'attribut « **comp** » (on cherche les compétences, et une compétence s'identifie grâce à « **comp** »).

On peut remplacer l'intersection par une jointure sur l'attribut **comp** (une jointure sur **np** ou une **jointure naturelle**, ça serait **faux**).

**2) Afficher les compétences nécessaires à la fois aux projets démarrant en 2018 et à ceux qui ont démarré en 2008.**

```

SELECT Cp18.comp
FROM CompProjet as Cp18, Projets as P18
WHERE Cp18.np = P18.np
And P18.datDebP >= '1/1/2018'
And P18.comp IN ( SELECT Cp08.comp
FROM CompProjet as Cp08, Projets as P08
WHERE Cp08.np = P08.np
And P08.datDebP >= '1/1/2008' )
    
```

Plusieurs solutions possibles :  
**IN, EXISTS** ou jointures.

**Ou encore**

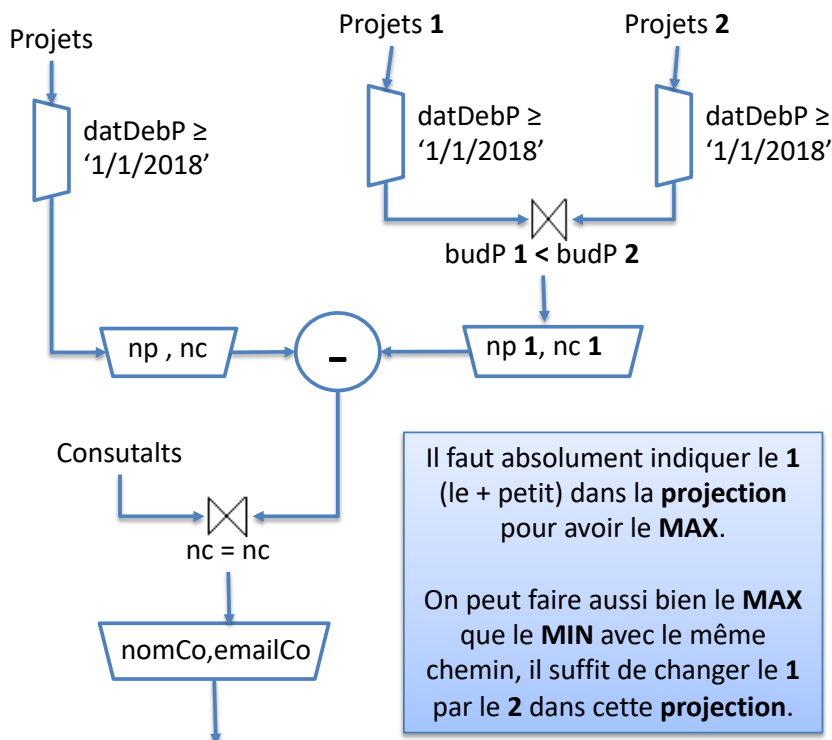
```

SELECT Cp18.comp
FROM CompProjet as Cp18, Projets as P18
WHERE Cp18.np = P18.np
And P18.datDebP >= '1/1/2018'
And EXISTS ( SELECT *
FROM CompProjet as Cp08, Projets as P08
WHERE Cp08.np = P08.np
And P08.datDebP >= '1/1/2008'
And Cp08.comp = Cp18.comp )
    
```

Si on utilise uniquement des jointures, il faut absolument avoir **2 fois** la table **Projets** et la table **CompProjet** dans le **FROM**, puisqu'on parle pas du même projet (un démarrant en 2018 et un autre démarrant en 2008).

Aussi possible avec **EXISTS**. Dans ce cas, on n'oublie pas la condition (« jointure ») reliant les 2 **SELECT** dans le 2<sup>ème</sup> **WHERE**

**3) Afficher le nom et l'email du consultant ayant dirigé le projet démarré en 2018 ayant le budget le plus élevé**



On va d'abord trouver le **projet** de 2018 qui compte le **budget le plus élevé**.  
 On a besoin des **restrictions** sur la date (**2018**) car de projets des autres années peuvent avoir un budget plus élevé alors qu'ils ne nous intéressent pas.

Il faut absolument indiquer le **1** (le + petit) dans la **projection** pour avoir le **MAX**.  
 On peut faire aussi bien le **MAX** que le **MIN** avec le même chemin, il suffit de changer le **1** par le **2** dans cette **projection**.

On trouve donc le projet (« **np** »), avec son **directeur** (indiqué par l'attribut « **nc** »).  
 Même si on veut le consultant qui dirige le projet, il faut **garder « np »**, car le même consultant peut diriger **plusieurs projets**, avec **différents budget**.

### 3) Afficher le nom et l'email du consultant ayant dirigé le projet démarré en 2018 ayant le budget le plus élevé

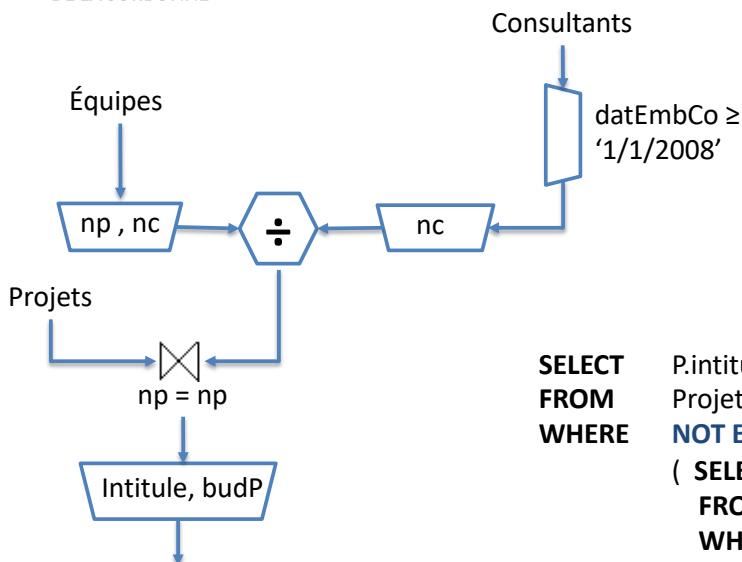
```

SELECT Co.nomCo, Co.emailCo
FROM Consultants as Co, Projets as P
WHERE Co.nc = P.nc
And P.datDebP >= '1/1/2018'
And P.budP = ( SELECT MAX ( Pmx.budP )
FROM Projets as Pmx
WHERE Pmx.datDebP >= '1/1/2008' )
  
```

Comme on ne connaît pas la valeur du **budget MAX en 2018**, il va falloir la retrouver. Le **2ème SELECT** sert à trouver ce budget **Max**.

Il faut garder la **restriction** sur la **date** aussi sur le **2ème SELECT**, car on peut avoir des projets avec un budget supérieur sur d'autres années.

### 4) Afficher les intitulés et les budgets des projets ayant impliqué tous les consultants embauchés en 2018.



$R(A, B) \div S(B)$   
 B → les consultants embauchés en 2018  
 A → les projets

```

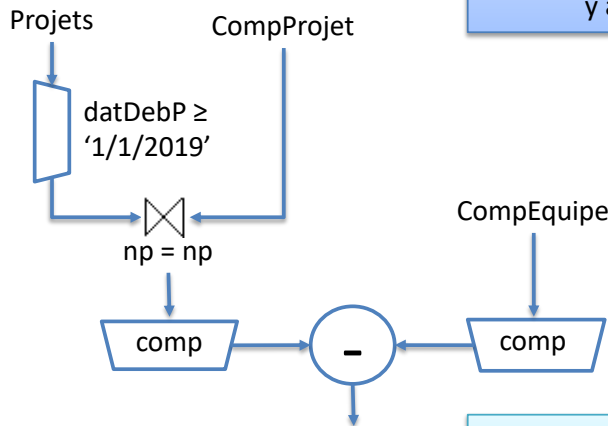
SELECT P.intitule, P.budP
FROM Projets as P
WHERE NOT EXISTS
( SELECT *
FROM Consultants as Co
WHERE Co.datEmbCo >= '1/1/2008'
AND NOT EXISTS
( SELECT *
FROM Equipes as E
WHERE E.nc = Co.nc
AND E.np = P.np
))
  
```

On retrouve notre **A** (les **projets**) dans le **1er SELECT**, notre **B** (les **consultants embauchés en 2018**) dans le **2ème SELECT**, et dans le **3ème SELECT**, les **équipes** qui permettent de relier les **projets** aux **consultants**.

5) Afficher les compétences demandées par des projets démarrant en 2019 pour lesquelles nous n'avons aucun consultant ayant déclaré cette compétence (*quel que soit son niveau*)

Les **compétences** requises par les **projets** sont dans la table **CompProjet**, alors que les **compétences** déclarées par les **consultants** sont dans **CompEquipe**.

Si une **compétence** est dans **CompEquipe**, c'est parce qu'il y a **au moins un consultant** la déclarant.



Comme on s'intéresse aux compétences, quel que soit leur niveau, on n'a besoin que de l'attribut **comp**.

5) Afficher les compétences demandées par des projets démarrant en 2019 pour lesquelles nous n'avons aucun consultant ayant déclaré cette compétence (*quel que soit son niveau*)

```
SELECT Cp.comp
FROM CompProjet as Cp, Projets as P
WHERE Cp.nc = P.nc
And P.datDebP >= '1/1/2019'
And Cp.comp NOT IN ( SELECT Ce.comp
FROM CompEquipe as Ce )
```

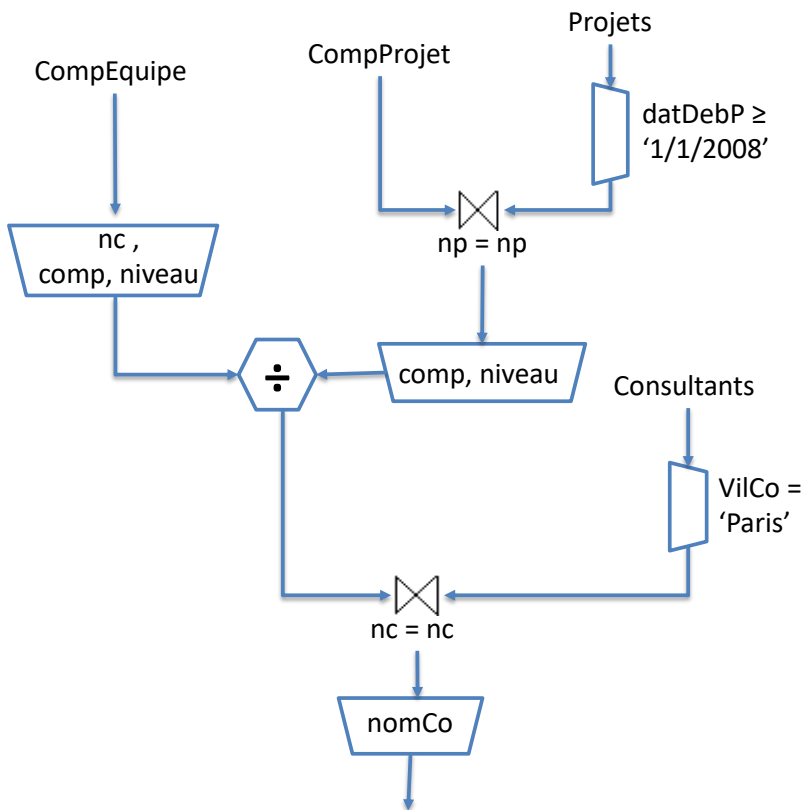
**Ou encore**

Le choix des tables à utiliser est essentiel. On aura besoin de la table **CompProjet** dans le **1<sup>er</sup> SELECT**, et de la table **CompEquipe** dans le **2<sup>ème</sup> SELECT**.

L'attribut pour le **NOT IN** doit être **comp**, puisque ces sont ces compétences qui ne peuvent pas être dans le **2<sup>ème</sup>** ensemble.

```
SELECT Cp.comp
FROM CompProjet as Cp, Projets as P
WHERE Cp.nc = P.nc
And P.datDebP >= '1/1/2019'
And NOT EXISTS ( SELECT *
FROM CompEquipe as Ce
WHERE Ce.comp = Cp.comp )
```

**6) Afficher les noms des consultants parisiens ayant toutes les compétences requises (avec les niveaux demandés) pour les projets démarrant en 2018**



$R(A, B) \div S(B)$   
 B → compétences + niveau des projets démarrant en 2018  
 A → les consultants parisiens

Attention aux attributs pour la division : le « B » ici est composé de deux attributs (comp, niveau), et le « A » par l'attribut « nc ». On aura donc, d'un côté « nc, comp, niveau » et de l'autre « comp, niveau ».

**6) Afficher les noms des consultants parisiens ayant toutes les compétences requises (avec les niveaux demandés) pour les projets démarrant en 2018**

```

SELECT Co.nomCo
FROM Consultants as Co
WHERE Co.vilCo = 'Paris'
And NOT EXISTS
( SELECT *
  FROM CompProjet as Cp, Projets as P
  WHERE Cp.np = P.np
  And P.datDebP >= '1/1/2008'
  And NOT EXISTS
  ( SELECT *
    FROM CompEquipe as Ce
    WHERE Ce.nc = Co.nc
    And Ce.comp = Cp.com
    And Ce.niveau = Cp.niveau ) )
    
```

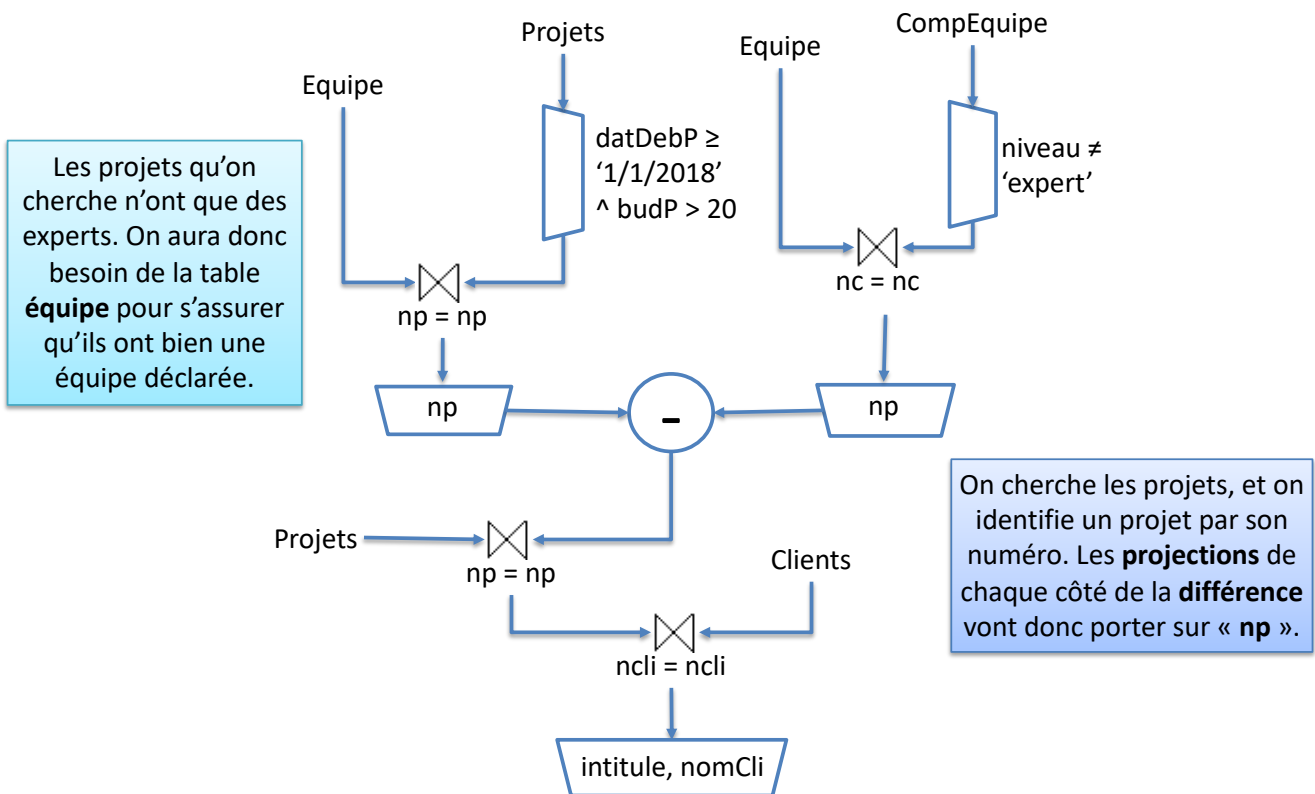
1<sup>er</sup> SELECT → A  
 consultants parisiens  
 2<sup>ème</sup> SELECT → B  
 Compétences et leurs niveaux des projets démarrant en 2018  
 3<sup>ème</sup> SELECT → on relie A et B  
 CompEquipe

Attention à bien placer les restrictions, elles doivent absolument être dans les bons SELECT :

1<sup>er</sup> SELECT → Restriction sur A  
 2<sup>ème</sup> SELECT → Restriction sur B  
 3<sup>ème</sup> SELECT → On relie A et B + restrictions sur ce qui relie A et B

Attention, on s'intéresse par les compétences et leur niveau (notre B), les « jointures » au 3<sup>ème</sup> SELECT doivent alors se faire sur ces attributs.

7) Afficher les projets (intitulés et noms des clients associés) démarrant en 2018 de plus de 20k€ de budget dont les équipes ne comptent que des consultants experts (consultants ayant les compétences au niveau expert).



7) Afficher les projets (intitulés et noms des clients associés) démarrant en 2018 de plus de 20k€ de budget dont les équipes ne comptent que des consultants experts (consultants ayant les compétences au niveau expert).

```

SELECT P.intitule, Cl.nomCli
FROM Projets as P, Clients as Cl, Equipes as E
WHERE P.ncli = Cl.ncli
And P.np = E.np
And P.datDebP >= '1/1/2018' And P.budP > 20
And P.np NOT IN ( SELECT E2.np
FROM Equipe as E2, CompEquipe as Ce
WHERE Ce.nc = E2.nc
And Ce.niveau <> 'expert' )

```

Ou encore

```

SELECT P.intitule, Cl.nomCli
FROM Projets as P, Clients as Cl, Equipes as E
WHERE P.ncli = Cl.ncli
And P.np = E.np
And P.datDebP >= '1/1/2018' And P.budP > 20
And NOT EXISTS ( SELECT *
FROM Equipe as E2, CompEquipe as Ce
WHERE Ce.nc = E2.nc
And Ce.niveau <> 'expert'
And E2.np = P.np )

```

Puisqu'on cherche les projets, on va opérer le « **NOT IN** » sur l'attribut « **np** », lequel on retrouve dans le 2<sup>ème</sup> **SELECT**.

Même chose pour le **NOT EXISTS** : la « jointure » au 2<sup>ème</sup> **SELECT** doit porter sur ce même attribut « **np** ».



8) Afficher le nombre total d'heures hebdomadaires dépensées par projet démarrant en 2018.

```
SELECT SUM (E.nbHeuSem), P.np
FROM Projets as P, Equipes as E
WHERE P.np = E.np
And P.datDebP >= '1/1/2018'
GROUP BY P.np
```

On veut connaître la somme des heures (indiqué par **nbHeuSem**), donc **SUM** (et pas **COUNT**).  
Par ailleurs, cette somme, on la fait **par projet**, donc **GROUP BY P.np**.  
L'attribut **np** étant présent dans les tables Equipe (qui contient nbHeuSem) et Projet, on a besoin de l'**alias**.

9) Afficher les intitulés et les noms des clients des projets dont l'intitulé commence par « Management » impliquant plus de 10 consultants.

```
SELECT P.np, P.intitule, Cl.nomCli
FROM Projets as P, Clients as Cl, Equipes
WHERE P.np = E.np
And P.ncli = Cl.ncli
GROUP BY P.np
HAVING COUNT(*) > 10
```

Pour savoir si un projet a plus de 10 consultants engagés, on devra regarder la table Equipe (où on affecte les équipes à un projet).  
Chaque consultant engagé dans un projet occupe une ligne, on va donc compter les lignes.  
On a donc besoin d'un HAVING...

10) Afficher les noms et les numéros des clients qui débiteront au moins 3 projets en 2019 avec un budget supérieur au budget moyen des projets démarrés en 2018.

```
SELECT Cl.ncli, Cl.nomCli
FROM Clients as Cl, Projets as P
WHERE Cl.ncli = P.ncli
And P.datDebP >= '1/1/2019'
And P.budP >= ( SELECT AVG (Pmoy.budP)
                FROM Projet as Pmoy
                WHERE Pmoy.datDebP > '1/1/2018' )

GROUP BY Cl.ncli
HAVING COUNT(*) > 3
```

On ne connaît pas le budget moyen des projets démarrés en 2018, il faut donc le trouver avec un SELECT.

On a besoin d'un HAVING (et par conséquent de son GROUP BY), car pour savoir si un client a plus de 3 projets avec un tel budget, il va falloir compter les lignes. Chaque ligne représente un projet ayant satisfait la condition mise dans le WHERE, de coup, si on a plus de 3 lignes c'est bon. 😊