

# Travaux pratiques sur base des étudiants

## Formation-SQL

### 1 Introduction

### 2 Exercices

#### Exercice 1:

Donner les titres des cours ayant au moins un étudiant inscrit et dont le nombre d'heures de TD est au moins 18.

```
SELECT titre
FROM UE
WHERE NumUE in (SELECT NumUE FROM Inscrit)
AND HTD >= 18;
```

#### Exercice 2:

Donner les noms des étudiants qui ont 'Albert A.' comme enseignant.

```
SELECT DISTINCT NomEt
FROM Etudiant , Inscrit , Enseigne , Enseignant
WHERE NomEns = 'Albert_A.'
AND Etudiant.NumEt = Inscrit.NumEt
AND Inscrit.NumUE = Enseigne.NumUE
AND Enseignant.NumEns = Enseigne.NumEns;
```

#### Exercice 3:

Donner les noms des enseignants qui enseignent dans la même UE que 'Albert A.' (sauf Albert A. lui-même).

```
SELECT DISTINCT E2.NomEns
FROM Enseignant E1, Enseignant E2, Enseigne C1, Enseigne C2
WHERE E1.NumEns = C1.NumEns
AND E2.NumEns = C2.NumEns
AND C1.NumUE = C2.NumUE
AND E2.NomEns <> 'Albert_A.'
AND E1.NomEns = 'Albert_A.';
```

#### Exercice 4:

Donner le titre des UEs dont le nombre d'heures total par groupe est au moins 46.

```
SELECT titre
FROM UE
WHERE hcours + htd + htp >= 46;
```

### Exercice 5:

Donner le nombre d'UE n'ayant pas de TP (on appellera NB\_UES l'attribut donnant ce résultat).

```
SELECT COUNT(*) AS NB_UES
FROM UE
WHERE HTP = 0;
```

### Exercice 6:

Donner le nombre d'étudiants qui suivent le cours d'Analyse (on appellera NB\_ETUDIANTS l'attribut donnant ce nombre).

```
SELECT COUNT(*) AS NB_ETUDIANTS
FROM UE NATURAL JOIN Inscrit
WHERE Titre = 'Analyse';
```

### Exercice 7:

Donner la moyenne du nombre d'heures de cours, de TD et de TP par UE. On appellera MOY\_COURS la moyenne des heures, MOY\_TD celle des TD et MOY\_TP celle des TP.

```
SELECT AVG(HCOURS) AS MOY_COURS, AVG(HTD) AS MOY_TD, AVG(HTP) AS MOY_TP
FROM UE;
```

### Exercice 8:

Donner le numéro de l'UE ayant le plus d'heures par groupe.

```
SELECT NumUE
FROM UE
WHERE HCours+HTD+HTP >= ALL (SELECT HCours+HTD+HTP FROM UE);
```

### Exercice 9:

Calculer le nombre total de groupes ayant 'Carine C.' comme chargée de TD. On nommera NB\_GROUPES l'attribut contenant le résultat.

```
SELECT SUM(NID) AS NB_GROUPES
FROM Enseigne NATURAL JOIN Enseignant
WHERE NomEns = 'Carine_C.';
```

### Exercice 10:

Calculer le nombre de groupes de TP par enseignant. On donnera le nom de l'enseignant et on appellera NB\_TP son nombre de groupes de TP.

```
SELECT NomEns, SUM(NTP) AS NB_TP
FROM (Enseignant NATURAL JOIN Enseigne)
GROUP BY NumEns, NomEns;
```

### Exercice 11:

Donner les numéros des enseignants qui effectuent plus de 17 heures de cours magistraux.

```
SELECT NumEns
FROM (Enseigne NATURAL JOIN UE)
```

```
GROUP BY NumEns
HAVING SUM(NCours*HCours) > 17;
```

#### Exercice 12:

Donner les numéros des enseignants qui effectuent plus de 17 heures de cours magistraux et qui participent au cours de 'Bases de données'.

```
SELECT NumEns
FROM (Enseigne NATURAL JOIN UE)
GROUP BY NumEns
HAVING SUM(NCours*HCours) > 17
AND NumEns IN (SELECT NumEns FROM (UE NATURAL JOIN Enseigne) WHERE Titre = 'Bases_
de_donnees' ) ;
```

#### Exercice 13:

Donner le(s) titre(s) de la (des) UE(s) ayant le plus d'inscrits.

```
SELECT Titre
FROM (UE NATURAL JOIN Inscrit)
GROUP BY NumUE, Titre
HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM Inscrit GROUP BY NumUE);
```

#### Exercice 14:

Donner pour chaque étudiant le nombre total d'heures qu'il suit. On donnera dans le résultat le numéro de l'étudiant ainsi qu'un attribut HEURES qui indiquera son nombre d'heures.

```
SELECT NumEt, SUM(HCOURS+HID+HTP) AS HEURES
FROM (Inscrit NATURAL JOIN UE)
GROUP BY NumEt;
```

#### Exercice 15:

Donner pour chaque enseignant le nombre total d'heures qu'il effectue. On donnera dans le résultat le nom de l'enseignant ainsi qu'un attribut HEURES qui indiquera son nombre d'heures.

```
SELECT NomEns, SUM(NCours*HCours+NID*HID+NTP*HTP) AS HEURES
FROM Enseignant , Enseigne , UE
WHERE Enseignant.NumEns = Enseigne.NumEns
AND Enseigne.NumUE = UE.NumUE
GROUP By Enseignant.NumEns,NomEns;
```

#### Exercice 16:

Donner pour chaque UE le nombre total d'heures tous groupes confondus. On exprimera le résultat en fonction du titre de l'UE et d'un attribut TOTAL\_HEURES indiquant le nombre total d'heures. On triera le résultat par nombre d'heures croissant.

```
SELECT titre , SUM(NCours*HCours+NID*HID+NTP*HTP) AS TOTAL_HEURES
FROM Enseigne , UE
WHERE Enseigne.NumUE = UE.NumUE
GROUP BY titre , UE.NumUE
ORDER BY SUM(NCours*HCours+NID*HID+NTP*HTP) ;
```

#### Exercice 17:

Donner les couples enseignant/étudiant tel que l'on soit sûr que l'enseignant ne rencontre pas l'étudiant lors d'un cours, d'un TD ou d'un TP. On indiquera les numéros dans les couples.

```
SELECT NumEns, NumEt
FROM Enseignant, Etudiant
WHERE (NumEns, NumEt) NOT IN (SELECT NumEns, NumEt FROM Enseigne NATURAL JOIN
    Inscrit);
```

### Exercice 18:

Donner la liste des couples d'étudiants qui suivent au moins deux cours ensemble. On indiquera le numéro du premier étudiant par l'attribut ETU\_A et celui de second par l'attribut ETU\_B. On ne souhaite pas avoir des couples avec le même numéro pour ETU\_A et ETU\_B, mais on souhaite obtenir une relation symétrique (i.e. si on a (3333,4444) dans le résultat, on doit avoir (4444,3333) également).

```
SELECT A.NumEt AS ETU_A, B.NumEt AS ETU_B
FROM Inscrit A, Inscrit B
WHERE A.NumUE = B.NumUE
    AND A.NumEt <> B.NumEt
GROUP BY A.NumEt, B.NumEt
HAVING COUNT(*) >= 2;
```

### Exercice 19:

Donner le nom du ou des étudiants qui suit le plus grand nombre d'heures de cours.

```
SELECT NomEt
FROM Etudiant, Inscrit, UE
WHERE Etudiant.NumEt = Inscrit.NumEt
    AND Inscrit.NumUE = UE.NumUE
GROUP BY Etudiant.NumEt, NomEt
HAVING SUM(HCours+HID+HTP) >= ALL
    (SELECT SUM(HCours+HID+HTP)
    FROM Inscrit, UE
    WHERE Inscrit.NumUE = UE.NumUE
    GROUP BY NumEt);
```

### Exercice 20:

Donner le nombre total d'heures de cours/TD/TP dispensées à l'université. On nommera TOTAL\_HEURES ce nombre.

```
SELECT SUM(NCours*HCours+NID*HID+NTP*HTP) AS TOTAL_HEURES
FROM Enseigne NATURAL JOIN UE;
```