

## Corrigé de la feuille d'exercices

### Bases de données

#### Exercice 1

1)

```
SELECT numero_atomique, symbole, nom FROM periodic
```

2)

```
SELECT symbole FROM periodic
WHERE Tfusion >=1000 AND Tfusion <= 2000
```

3)

```
SELECT nom, MIN(conductivite_thermique) FROM periodic
```

4)

```
SELECT nom FROM periodic ORDER BY Tebullition DESC LIMIT 1
```

5)

```
SELECT nom FROM periodic ORDER BY masse_atomique DESC LIMIT 15
```

#### Exercice 2

1)

```
SELECT sujet FROM nobel GROUP BY sujet
```

2)

```
SELECT sujet, COUNT(laureat) FROM nobel GROUP BY sujet
```

3)

```
SELECT COUNT(laureat) FROM nobel
WHERE annee > 1900 AND annee <= 2000
AND sujet = 'Physique'
```

4)

```
SELECT COUNT(*) AS NbreNobel, laureat FROM nobel
GROUP BY laureat;
```

5)

```
SELECT laureat, NbreNobel FROM
(SELECT COUNT(*) AS NbreNobel, laureat FROM nobel
GROUP BY laureat)
WHERE NbreNobel > 1;
```

6)

```
SELECT nobel.laureat, sujet, annee FROM
(SELECT laureat AS supernobel FROM
(SELECT COUNT(*) AS NbreNobel, laureat FROM nobel
GROUP BY laureat)
WHERE NbreNobel > 1)
JOIN nobel on supernobel = nobel.laureat
ORDER BY nobel.laureat;
```

#### Exercice 3

##### I. Requêtes simples

a)

```
SELECT SUM(population_2010) FROM communes
```

b) Remarquer l'utilisation de 100.0 pour avoir un résultat non entier :

```
SELECT 100.0 * (SUM(population_2010)-SUM(population_1999))
/ SUM(population_1999) FROM communes;
```

c)

```
SELECT nom FROM communes ORDER BY population_2010 DESC LIMIT 10
```

d)

```
SELECT nom, population_2010/surface AS densite FROM communes
ORDER BY densite DESC LIMIT 10
```

e) En population absolue :

```
SELECT nom, population_2010-population_1999 AS accroissement
FROM communes
ORDER BY accroissement DESC LIMIT 10
```

En population relative : (remarquer 1.0 pour avoir un résultat non entier) :

```
SELECT nom, 1.0*(population_2010-population_1999)/population_1999
AS accroissement FROM communes
ORDER BY accroissement DESC LIMIT 10
```

f)

```
SELECT num_departement, SUM(population_2010) AS population
FROM communes
GROUP BY num_departement ORDER BY population DESC
```

## II - Jointures à deux tables

a)

```
SELECT departements.nom, COUNT(communes.nom)
FROM departements JOIN communes
ON communes.num\_departement = departements.num_departement
GROUP BY communes.num_departement;
```

b)

```
SELECT departements.nom
FROM departements JOIN regions
ON departements.num_region = regions.num_region
AND (regions.nom = "Basse Normandie"
OR regions.nom = "Haute Normandie")
```

c)

```
SELECT departements.nom, SUM(communes.population_2010) AS pop
FROM departements JOIN communes
ON communes.num_departement = departements.num_departement
GROUP BY departements.nom
ORDER BY pop DESC;
```

d)

```
SELECT departements.nom,
SUM(communes.population_2010)/SUM(communes.surface) AS densite
FROM departements JOIN communes
ON communes.num_departement=departements.num_departement
GROUP BY communes.num_departement
ORDER BY densite DESC;
```

e) Avec ORDER BY ... HAVING ... :

```
SELECT departements.nom, SUM(population_2010) AS pop
FROM departements JOIN communes
ON departements.num_departement=communes.num_departement
GROUP BY departements.nom HAVING pop >= 10000000;
```

Avec une requête composée :

```
SELECT nom, pop
FROM (SELECT departements.nom, SUM(population_2010)
AS pop FROM departements)
JOIN communes
ON departements.num_departement= communes.num_departement
GROUP BY departements)
WHERE pop >= 10000000
```

## III - Intersection

a)

```
SELECT nom FROM communes WHERE num_departement = 31
INTERSECT
SELECT nom FROM communes WHERE num_departement = 24
```

b)

```
SELECT nom FROM communes WHERE num_departement = 13
INTERSECT
SELECT nom FROM communes WHERE NOT num_departement = 13;
```

#### IV - Jointures à 3 tables

1)

```
SELECT communes.nom, departements.nom, regions.nom, communes.zmax
FROM communes JOIN departements JOIN regions
ON communes.num_departement=departements.num_departement
AND departements.num_region=regions.num_region
ORDER BY communes.zmax DESC LIMIT 10;
```

2)

```
SELECT regions.nom, COUNT(communes.nom)
FROM regions JOIN departements JOIN communes
ON communes.num_departement=departements.num_departement
AND departements.num_region=regions.num_region
GROUP BY regions.nom;
```

3)

```
SELECT regions.nom, SUM(communes.population_2010) AS pop
FROM regions JOIN departements JOIN communes
ON communes.num_departement=departements.num_departement
AND departements.num_region=regions.num_region
GROUP BY regions.nom
ORDER BY pop DESC LIMIT 10;
```