

Bases de Données relationnelles

TD 1

Informatique
Lycée Thiers

Exercice 1 : Base de donnée nobel.sqlite

Enoncé

Corrigé

Exercice 2 : Base de donnée atomic.sqlite

Enoncé

Corrigé

Exercice 3 : base de donnée movie.sqlite

Enoncé

Corrigé

Exercice 4 : densité de population et ressources hydriques

Enoncé

Corrigé

Exercice 1 : Enoncé

Exercice 1. Charger la base de donnée `nobel.sqlite` concernant les lauréats du prix nobel. Elle contient une seule table `nobel`

1. Quel est son schéma de relation ? Quelle requête a permis sa création ?
2. Afficher tous les lauréats de l'année 1990 et leurs disciplines.
3. Afficher les différentes disciplines du nobel.
4. 4.1 Déterminer pour chaque lauréat le nombre de prix nobel obtenus,
4.2 En déduire les noms des lauréats ayant reçu plusieurs prix nobel.

Exercice 1 : Corrigé

(1)

Schéma de relation :

```
nobel (année INT, sujet VARCHAR(15), laureat VARCHAR(50))
```

Obtenu par :

```
CREATE TABLE nobel(  
  annee INT  
  ,sujet VARCHAR(15)  
  ,laureat VARCHAR(50)  
)
```

(2)

```
SELECT laureat, sujet FROM nobel WHERE annee = 1990;
```

(3)

```
SELECT DISTINCT sujet FROM nobel
```

Exercice 1 : Corrigé

(4.1)

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

(4.2)

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

Exercice 2 : Enoncé

Exercice 2. Charger la base `periodic.sqlite` qui contient une table contenant les données de la table périodique des éléments.

1. Afficher numero atomique, symbole et nom des éléments de la table périodique.
2. Afficher les symboles des éléments dont la température de fusion est comprise entre 1000 et 2000 °K.
3. Quels est l'éléments ayant la température d'ébullition la plus élevée ?
4. Quels sont les 15 atomes les plus lourds ?

Exercice 2 : Corrigé

1)

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

2)

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

3)

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

4)

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

Exercice 3 : Énoncé

Exercice 3. Charger la base de donnée movie.sqlite. Elle contient 3 tables vérifiant les schémas suivants :

```
actor ( id INTEGER, name VARCHAR(35))
casting (movieid INTEGER, actorid INTEGER, ord INTEGER)
movie (id INTEGER, title VARCHAR(70), yr DECIMAL(4), score FLOAT,
       votes INTEGER, director INTEGER)
```

Elle porte sur tous les films de cinéma, leur casting, acteurs et réalisateurs jusqu'à l'année 2000. La table actor contient les acteurs mais aussi les réalisateurs

1. Afficher les (titres des) 10 films les mieux notés de l'année 1995.
2. Afficher le nombre de films contenus dans la base.
3. Afficher la liste des acteurs/réalisateurs dont les initiales sont T.G.
4. Combien de films ont obtenu une note supérieure à 8 ?
5. Combien d'acteurs/réalisateurs ont un nom commençant par la lettre A ?
6. Afficher les noms de tous les réalisateurs.
7. Afficher les noms des 10 réalisateurs dont la moyenne des scores de leurs films est la plus basse.

Exercice 3 : corrigé

(1)

```
SELECT title FROM movie ORDER BY score DESC LIMIT 10
```

(2)

```
SELECT COUNT(title) FROM movie;
```

(3)

```
SELECT name FROM actor WHERE name LIKE 'T% G%'
```

(4)

```
SELECT COUNT(title) FROM movie WHERE score > 8
```

(5)

```
SELECT COUNT(name) FROM actor WHERE name LIKE '% A%'
```

Exercice 3 : corrigé

(6)

```
SELECT name FROM actor WHERE id IN (SELECT director FROM movie)
```

(7)

```
SELECT name FROM actor WHERE id IN  
(SELECT director FROM  
(SELECT director, AVG(score) AS moy FROM movie GROUP BY director  
ORDER BY moy LIMIT 10))
```

Exercice 4 : Enoncé

Région	Eau Renouvelable	Superficie	Population
Monde	43 000 km ³	130 677 343 km ²	6 033 323 000
Afrique	11%	23%	13%
Amérique du Nord	15%	17%	8%
Amérique du Sud	26%	13%	6%
Asie	36%	24%	61%
Europe	8%	17%	12%
Océanie	5%	6%	1%

Exercice 4.

1. Créer une base de donnée appelée continents, puis une table t0 dans cette base ayant 4 attributs : nom VARCHAR(20), eau INT(2), superficie INT(2), population INT(2).
2. A partir des données pour les 6 continents dans le tableau ci-dessus charger la table t0.
3. Créer 3 tables teau, tsup, tpop ayant chacune pour attributs nom VARCHAR(20), et valeur FLOAT. Donner des requêtes permettant de les remplir avec les 6 continents et les valeurs absolues (sans %) de l'eau, de la superficie et de la population.
4. Donner une requête retournant le nom, classé dans l'ordre alphabétique, puis par densité décroissante, de ceux de densité inférieure à 30 hab/km².

Exercice 4 : Corrigé

(1)

```
CREATE TABLE t0 (nom VARCHAR(20), eau INT(2), superficie INT(2),  
population INT(2))
```

(2) Chargement des données :

```
INSERT INTO t0 VALUES('Afrique', 11, 23, 13);  
INSERT INTO t0 VALUES('AmerNord', 15, 17, 8);  
INSERT INTO t0 VALUES('AmerSud', 26, 13, 6);  
INSERT INTO t0 VALUES('Asie', 36, 24, 61);  
INSERT INTO t0 VALUES('Europe', 8, 17, 12);  
INSERT INTO t0 VALUES('Océanie', 5, 6, 1);
```

Exercice 4 : Corrigé

3)

```
CREATE TABLE teau(nom VARCHAR(20), valeur FLOAT);
INSERT INTO teau SELECT nom, 43000*eau/100.0 FROM t0;
CREATE TABLE tsup(nom VARCHAR(20), valeur FLOAT);
INSERT INTO tsup SELECT nom, 130677343*superficie/100.0 FROM t0;
CREATE TABLE tpop(nom VARCHAR(20), valeur FLOAT);
INSERT INTO tpop SELECT nom, 6033323000*population/100 FROM t0;
```

4)

```
SELECT nom FROM tdens WHERE valeur <= 30 ORDER BY nom
```

```
SELECT nom FROM tdens WHERE valeur <= 30 ORDER BY valeur DESC
```