

Exercice 1 :

- colonne, column, attribut
- entité, ligne, row
- domaine, type
- relation, table
- schéma (description d'une relation)
- base de données (ensemble des relations)

Exercice 2 :

1. — `Especes(id Integer, nom String)`
 - `Animaux(id Integer, nom String, age Integer, id_espece Integer)`
 - `Soins(id Integer, id_animal Integer, soin String)`
2. Dans un souci de lisibilité il est judicieux d'écrire un attribut par ligne.

```
1 CREATE TABLE Especies (  
2     id Integer PRIMARY KEY AUTOINCREMENT,  
3     nom String);  
4  
5 CREATE TABLE Animaux (  
6     id Integer PRIMARY KEY AUTOINCREMENT,  
7     nom String,  
8     age Integer,  
9     id_espece Integer,  
10    FOREIGN KEY (id_espece) REFERENCES Especies(id));  
11  
12 CREATE TABLE Soins (  
13     id Integer PRIMARY KEY AUTOINCREMENT,  
14     id_animal Integer,  
15     soin String,  
16    FOREIGN KEY (id_animal) REFERENCES Animaux(id));
```

Code 1 – Création des trois tables

Exercice 3 :

1. Dans le dossier compressé.
2. <https://fr.wikipedia.org/wiki/Soundex>.
3. Les codes :

```
1 SELECT * FROM Departements WHERE departement_code = 24;  
2  
3 SELECT * FROM Departements WHERE departement_nom_soundex = 'M200';  
4  
5 SELECT departement_nom FROM Departements WHERE departement_code <  
6     10;
```

```
7 SELECT departement_code, departement_nom FROM Departements WHERE
   departement_code > 20 AND departement_code < 30;
8
9 -- SQLite LIKE operator is case-insensitive. It means "A" LIKE "a"
   is true.
10 -- However, for Unicode characters that are not in the ASCII ranges
   , the LIKE operator is case sensitive e.g., "Ä" LIKE "ä" is false
11 SELECT departement_nom FROM Departements WHERE departement_nom LIKE
   '%haut%';
12
13 SELECT departement_nom FROM Departements WHERE departement_nom NOT
   LIKE '%-%' AND departement_nom NOT LIKE '% %';
```

Code 2 – Sélections des départements

Remarque

Il est possible de comparer des *String* comme des *Integer*. Le SQL est très permissif : `departement_code` est de type *String*, pourtant il accepte la comparaison avec un *Integer*.

Exercice 4 :

```
1 SELECT * FROM employees WHERE name = 'GARFIELD';
2
3 SELECT name FROM employees WHERE designation = 'MANAGER';
4
5 SELECT name FROM employees WHERE name LIKE 'H%';
6
7 SELECT name FROM employees WHERE hired_on > '1997-01-01';
8
9 SELECT name, salary FROM employees WHERE salary > 25000 AND salary <
   55000;
10
11 SELECT name, salary FROM employees WHERE salary > 25000 AND salary <
   55000 AND commission IS NOT NULL;
12 -- ou bien; NULL est différent de 0; commission = 0 ne renverrait rien
   ici
13 SELECT name, salary FROM employees WHERE salary > 25000 AND salary <
   55000 AND commission > 0;
14
15 INSERT INTO employees (name, designation, manager, hired_on, salary,
   dept) VALUES ('DURAN', 'TECH', 6, '1999-01-13', 35000, 4);
16
17 UPDATE employees SET salary = 60000 WHERE name = 'FILLMORE';
```

Code 3 – Sélections des employés