Exercices SGBD manipulation correction

Christophe Viroulaud

Terminale - NSI

BDD 05

Sommaire

1. Exercice 1

2. Exercice 2

3. Exercice 3

4. Exercice 4

Exercice 1

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

Sommaire

1. Exercice 1
2. Exercice 2
3. Exercice 3
4. Exercice 4

Exercice 2

- Especes(id Integer, nom String)
- Animaux(id Integer, nom String, age Integer, id_espece Integer)
- Soins(id Integer, id_animal Integer, soin String)

```
CREATE TABLE Especes (
        id Integer PRIMARY KEY AUTOINCREMENT,
        nom String);

CREATE TABLE Animaux (
        id Integer PRIMARY KEY AUTOINCREMENT,
        nom String,
        age Integer,
        id_espece Integer,
        FOREIGN KEY (id_espece) REFERENCES Especes(id));

CREATE TABLE Soins (
    id Integer PRIMARY KEY AUTOINCREMENT,
    id_animal Integer,
    soin String,
    FOREIGN KEY (id_animal) REFERENCES Animaux(id));
```

Code 1 – Création des 3 tables

```sql
INSERT INTO Especes (nom) VALUES
("chien"),
("chat"),
("poisson");
```

Code 2 – Insertion espèces

```sql
INSERT INTO Animaux (nom, age, id_espece) VALUES
("Minou", 15, 2),
("Tex", 8, 1),
("Rrrrr", 2, 1);
```

Code 3 – Insertion animaux

Remarque

Les identifiants des espèces peuvent varier.

```sql
INSERT INTO Soins (id_animal, soin) VALUES
(2, "patte cassée - plâtre"),
(1, "fièvre - antibiotiques");
```

Code 4 – Insertion soins

Sommaire

1. Exercice 1

2. Exercice 2

3. Exercice 3

4. Exercice 4

Exercice 3

Soundex : algorithme phonétique d'indexation
corriger les erreurs orthographiques
`https://fr.wikipedia.org/wiki/Soundex`

```
SELECT * FROM Departements WHERE
department_code = 24;

SELECT * FROM Departements WHERE
   departement_nom_soundex = 'M200';

SELECT departement_nom FROM Departements WHERE
   departement_code < 10;

SELECT departement_code, departement_nom FROM
   Departements WHERE
departement_code > 20 AND
departement_code < 30;
```

Remarque

- SQLite LIKE operator is case-insensitive. It means "A" LIKE "a" is true.
- However, for Unicode characters that are not in the ASCII ranges, the LIKE operator is case sensitive e.g., "Ä" LIKE "ä" is false

```sql
SELECT departement_nom FROM Departements WHERE
departement_nom LIKE '%haut%';

SELECT departement_nom FROM Departements WHERE
departement_nom NOT LIKE '%-%' AND
departement_nom NOT LIKE '% %';
```

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*.

Sommaire

1. Exercice 1
2. Exercice 2
3. Exercice 3
4. Exercice 4

Exercice 4

```sql
SELECT * FROM employees WHERE name = 'GARFIELD';

SELECT name FROM employees WHERE designation = '
   TECH';

SELECT name FROM employees WHERE name LIKE 'H%';

SELECT name FROM employees WHERE hired_on > '
   1997-01-01';

SELECT name, salary FROM employees WHERE
salary > 25000 AND
salary < 55000;
```

```sql
SELECT name, salary FROM employees WHERE
salary > 25000 AND
salary < 55000 AND
commission IS NOT NULL;

--ou bien
SELECT name, salary FROM employees WHERE
salary > 25000 AND
salary < 55000 AND
commission > 0;
```

Remarque

NULL est différent de 0 ; commission = 0 ne renverrait rien ici

```sql
INSERT INTO employees (name, designation, manager, hired_on, salary, dept) VALUES ('DURAN', 'TECH', 6, '1999-01-13', 35000, 4);

UPDATE employees SET salary = 60000 WHERE
name = 'FILLMORE';
```