

Sur Teams, télécharger les fichiers suivants :

- SQL01.sqlite
- donnees_etudiants.csv
- donnees_communes.csv

1. Découverte rapide de SQLiteStudio.

2. Dans la base de données **SQL01**, créer une table dont le schéma est :

```
donnees_etudiants(id:INTEGER,nom:TEXT,prenom:TEXT,dep:TEXT,commune:TEXT)
```

avec comme clé primaire l'attribut **id**.

```
CREATE TABLE donnees_etudiants(
id INTEGER,
nom TEXT,
prenom TEXT,
dep TEXT,
commune TEXT,
PRIMARY KEY (id)
);
```

Remarque

L'attribut **dep** contiendra le numéro de département qui est en type **TEXT** car il faut penser aux corse...

3. Ajouter votre numéro de département de résidence en attribut **dep**.

```
UPDATE donnees_etudiants SET dep='69' WHERE nom='PTIPEU'
```

4. Choisir une étudiante ou un étudiant puis :

- récupérer ses informations,

```
SELECT * FROM donnees_etudiants WHERE nom='PTIPEU'
```

- supprimer sa ligne,

```
DELETE FROM donnees_etudiants WHERE nom='PTIPEU'
```

- recréer sa ligne.

```
INSERT INTO donnees_etudiants VALUES ('41','PTIPEU','Justin','69')
```

Important !

Dans toute la suite, on considère un étudiant fictif dans la classe : Justin PTIPEU, qui habite dans le 1^{er} arrondissement de Lyon.

Remarque

On pourrait utiliser l'identifiant dans le cas où deux élèves auraient le même nom de famille.

5. Donner une clé primaire de la table **donnees_communes** ainsi qu'une clé étrangère vers la table **donnees_regions**.

Écrire alors la requête qui a permis la création de la table **donnees_communes**.

com est une clé primaire de cette table ; et **reg** est une clé étrangère vers la table **donnees_regions**.

```
CREATE TABLE donnees_communes(
reg TEXT,
region TEXT,
dep TEXT,
com TEXT,
commune TEXT,
pop_commune INTEGER,
PRIMARY KEY (com),
FOREIGN KEY (reg) REFERENCES donnees_regions(reg)
);
```

6. Importer ensuite le fichier **donnees_communes.csv** dans cette table.

7. Récupérer l'intitulé exact de votre ville de résidence et l'ajouter dans l'attribut **commune** qui vous est associé dans la table **donnees_etudiants**.

```
UPDATE donnees_etudiants SET commune='Lyon 1er Arrondissement' WHERE nom='PTIPEU'
```

8. Récupérer la population de la ville de votre choix.

```
SELECT pop_commune FROM donnees_communes WHERE nom='Lyon 1er Arrondissement'
```

9. Trouver les villes comptant 1743 habitants.

```
SELECT * FROM donnees_communes WHERE pop_commune='1743'
```

10. Trouver les villes du Rhône comptant plus de 20 000 habitants.

```
SELECT * FROM donnees_communes WHERE dep='69' AND pop_commune>='20000'
```

11. À l'aide d'une jointure, afficher les informations suivantes pour toutes et tous dans la classe : nom, prénom, commune, population de la commune.

```
SELECT nom, prenom, commune, pop_commune
FROM donnees_etudiants
JOIN donnees_communes
ON donnees_etudiants.commune=donnees_communes.commune
```

12. À l'aide d'une jointure, afficher les informations suivantes vous concernant : nom, prénom, commune, population de la commune.

```
SELECT nom, prenom, commune, pop_commune
FROM donnees_etudiants
JOIN donnees_communes
ON donnees_etudiants.commune=donnees_communes.commune
WHERE nom='PTIPEU'
```

Remarque

S'il y a un doute, on peut mettre **WHERE donnees_etudiants.nom='PTIPEU'**.

13. À l'aide de jointures, afficher les informations suivantes vous concernant : nom, prénom, commune, population de la commune, département, population du département, région et population de la région.

```
SELECT nom, prenom, donnees_etudiants.commune, pop_commune, departement, pop_departement, donnees_regions.region, pop_region
FROM donnees_etudiants
JOIN donnees_communes
ON donnees_etudiants.commune=donnees_communes.commune
JOIN donnees_departements
ON donnees_communes.dep=donnees_departements.dep
JOIN donnees_regions
ON donnees_departements.reg=donnees_regions.reg
WHERE nom = 'PTIPEU'
```