TD 4 :
Sur les bases de données

PC* - Lycée Thiers

Exercice 1 : D'après X-ENS 2016
- Enoncé
- Correction

Exercice 2 : D'après Centrale-Supélec 2016
- Enoncé
- Correction

Exercice 3 : D'après Mines-Ponts 2016
- Enoncé
- Correction

Exercice 4 : Base de donnée `movie`
- Enoncé
- Correction

Exercice 5 : Dessins en `python`
- Enoncé
- Corrigé

D'après X-ENS 2016 MP-PC

Exercice 1. Une représentation simplifiée, réduite à deux tables de la base de donnée d'un réseau social est donnée dans la figure suivante :

INDIVIDUS

id	nom	prenom
1	Potter	Harry
2	Granger	Hermione
⋮	⋮	⋮

LIENS

id1	id2
1	2
2	1
⋮	⋮

La table `INDIVIDUS` répertorie les individus et contient les colonnes

- `id` (clé primaire), un entier identifiant chaque individu ;
- `nom`, une chaine de caractère donnant le nom de famille de l'individu ;
- `prenom`, une chaine de caractères donnant le prénom de l'individu.

La table `LIENS` répertorie les liens d'amitiés entre individus et contient les colonnes

- `id1`, entier identifiant le premier individu du lien d'amitié ;
- `id2`, entier identifiant le second individu du lien d'amitié ;

Par exemple Harry Potter et Hermione Granger sont amis dans le réseau social. On supposera par ailleurs que pour tout couple `(x,y)` présent dans la table `LIENS`, le couple `(y,x)` y est aussi présent.

1. Écrire une requête SQL qui renvoie les identifiants des amis de l'individu d'identifiant `x`.
2. Écrire une requête SQL qui renvoie les (noms,prénoms) des amis de l'individu d'identifiant `x`.
3. Écrire une requête SQL qui renvoie les identifiants des individus qui sont amis avec au moins un ami de l'individu d'identifiant `x`.
4. Écrire une requête SQL qui renvoie les (noms,prénoms) des individus qui sont amis avec au

D'après X-ENS 2016 MP-PC - Corrigé

1.

```
SELECT id2 FROM LIENS WHERE id1 = x
```

2. par une requête utilisant IN :

```
SELECT nom, prenom FROM INDIVIDUS
        WHERE id IN (SELECT id2 FROM LIENS WHERE id1 = x)
```

ou à l'aide d'une jointure :

```
SELECT nom, prenom FROM INDIVIDUS JOIN LIENS
        ON id = id2 AND id1 = x
```

3.

```
SELECT DISTINCT id1 FROM LIENS
    WHERE id2 IN (SELECT id1 FROM LIENS WHERE id2 = x)
```

4.

```
SELECT nom, prenoms FROM INDIVIDUS JOIN
    (SELECT DISTINCT id1 FROM LIENS
        WHERE id2 IN (SELECT id1 FROM LIENS WHERE id2 = x))
ON id = id1
```

D'après Centrale-Supélec 2016

Exercice 2. Nous modélisons (de manière très simplifée) les plans de vol gérés par Eurocontrol sous la forme d'une base de données comportant deux tables : la table `vol` qui répertorie les plans de vol déposés par les compagnies aériennes ; elle contient les colonnes

- `id_vol` : numéro du vol (chaine de caractères) ;
- `depart` : code de l'aéroport de départ (chaine de caractère) ;
- `arrivee` : code de l'aéroport d'arrivée (chaine de caractère) ;
- `jour` : jour du vol (de type `date`, au format `aaaa-mm-jj` ;
- `heure` : heure de décollage souhaitée (de type `time`, au format `hh:mm`) ;
- `niveau` : niveau de vol souhaité (entier).

id_vol	depart	arrivee	jour	heure	niveau
AF1204	CDG	FCO	2016-05-02	07 :35	300
AF1205	FCO	CDG	2016-05-02	10 :25	300
AF1504	CDG	FCO	2016-05-02	10 :05	310
AF1505	FCO	CDG	2016-05-02	13 :00	310

Figure 1 extrait de la table `vol` : vols de la compagnie Air France entre les aéroports Charles-de-Gaulle (Paris) et Léonard-de-Vinci à Fiumicino

D'après Centrale-Supélec 2016

la table `aeroport` qui répertorie les aéroports européens ; elle contient les colonnes

- `id_aero` : code de l'aéroport (chaine de caractères) ;
- `ville` : principale ville desservie (chaine de caractères) ;
- `pays` : pays dans lequel se situe l'aéroport (chaine de caractères).

id_aero	ville	pays
CDG	Paris	France
ORY	Paris	France
MRS	Marseille	France
FCO	Rome	Italie

Figure 2 extrait de la table `aeroport`

D'après Centrale-Supélec 2016

1. Écrire une requête SQL qui fournit le nombre de vols qui doivent décoller dans la journée du 2 mai 2016 avant midi.
2. Écrire une requête SQL qui fournit la liste des numéros de vols au départ d'un aéroport le 2 mai 2016 et desservant Paris.
3. Que fait la requête suivante ?

```
SELECT id_vol
FROM vol
    JOIN aeroport AS d ON d.id_aero = depart
    JOIN aeroport AS a ON a.id_aero = arrivee
WHERE
    d.pays = 'France' AND
    a.pays = 'France' AND
    jour = '2016-05-02'
```

4. Certains vols peuvent engendrer des conflits potentiels : c'est par exemple le cas lorsque deux avions suivent un même trajet, en sens inverse, le même jour et à un même niveau. Écrire une requête SQL qui fournit la liste des couples $(\mathrm{Id}_1, \mathrm{Id}_2)$ des identifiants des vols dans cette situation.

D'après Centrale-Supélec 2016

1.

```
SELECT COUNT(*) FROM vols
    WHERE jour = '2016-05-02' AND heure < '12:00'
```

2.

```
SELECT id_vol FROM vols
    WHERE jour = '2016-05-02' AND
    arrivee IN (SELECT id_aero FROM aeroport
        WHERE ville = 'Paris')
```

3. Retourne les numéros des vols restreints au seul territoire français ayant eu lieu le 05 mai 2016.

D'après Centrale-Supélec 2016

4.

```
SELECT vol.id_vol, T2.id_vol
    FROM vol
        JOIN
    SELECT id_vol, depart, arrivee, jour, niveau FROM vol AS T2
        ON
    vol.depart = T2.arrivee AND
    vol.arrivee = T2.depart AND
    vol.jour = T2.jour AND
    vol.niveau = T2.niveau
```

D'après Mines-Ponts 2016

Exercice 3. Pour suivre la propagation des épidémies, de nombreuses données sont recueillies par les institutions internationales comme l'O.M.S. Par exemple, pour le paludisme, on dispose de deux tables :

– la table `palu` recense le nombre de nouveaux cas confirmés et le nombre de décès liés au paludisme ; certaines lignes de cette table sont données en exemple (on précise que `iso` est un identifiant unique pour chaque pays) :

nom	iso	annee	cas	deces
Bresil	BR	2009	309 316	85
Bresil	BR	2010	334 667	76
Kenya	KE	2010	898 531	26 017
Mali	ML	2011	307 035	2 128
Ouganda	UG	2010	1 581 160	8431

– la table `demographie` recense la population totale de chaque pays ; certaines lignes de cette table sont données en exemple :

pays	periode	pop
BR	2009	193 020 000
BR	2010	194 946 000
KE	2010	40 909 000

D'après Mines-Ponts 2016

1. Au vu des données présentées dans la table `palu`, parmi les attributs `nom`, `iso` et `annee`, quels attributs peuvent servir de clé primaire ? Un couple d'attributs pourrait-il servir de clé primaire ? (on considère qu'une clé primaire peut posséder plusieurs attributs). Si oui, en préciser un.
2. Écrire une requête en langage SQL qui récupère depuis la table `palu` toutes les données de l'année 2010 qui correspondent à des pays où le nombre de décès dus au paludisme est supérieur ou égal à 1 000.

On appelle taux d'incidence d'une épidémie le rapport du nombre de nouveaux cas pendant une période donnée sur la taille de la population-cible pendant la même période. Il s'exprime généralement en " nombre de nouveaux cas pour 100 000 personnes par année ". Il s'agit d'un des critères les plus importants pour évaluer la fréquence et la vitesse d'apparition d'une épidémie.

3. Écrire une requête en langage SQL qui détermine le taux d'incidence du paludisme en 2011 pour les différents pays de la table `palu`.
4. Écrire une requête en langage SQL permettant de déterminer le nom du pays ayant eu le deuxième plus grand nombre de nouveaux cas de paludisme en 2010 (on pourra supposer qu'il n'y a pas de pays ex-æquo

D'après Mines-Ponts 2016 - Corrigé

1. Aucun ; aucune ne prend que des valeurs uniques. On peut choisir un couple d'attributs, `(nom,annee)` ou `(iso,annee)` comme clé primaire.

2.

```
SELECT * FROM palu WHERE annee = 2010 AND deces >= 1000
```

3.

```
SELECT nom, 100 000 * cas / pop AS txIncidence
    FROM palu JOIN demographie ON iso = pays AND annee = periode
    WHERE periode = 2011
```

D'après Mines-Ponts 2016 - Corrigé

4. Plusieurs approches sont possibles :

```
SELECT nom FROM palu WHERE annee = 2010
    ORDER BY cas DESC LIMIT 2,1
```

```
SELECT nom FROM palu WHERE annee = 2010
    ORDER BY cas DESC LIMIT 2
EXCEPT
SELECT nom FROM palu WHERE annee = 2010
    ORDER BY cas DESC LIMIT 1
```

```
SELECT nom FROM palu WHERE annee = 2010
AND cas = (SELECT MAX(cas) FROM palu WHERE annee = 2010
    AND cas < (SELECT MAX(cas) FROM palu WHERE annee = 2010))
```

Base de donnée movie

Exercice 4. 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 cinema, leur casting, acteurs et réalisateurs jusqu'à l'année 2000. La table `actor` contient les acteurs mais aussi les réalisateurs

1. Obtenir titre et score des films ayant obtenu au moins 10000 votes classés par score décroissant
2. Obtenir les titres de tous les films réalisés par Alfred Hitchcock.
3. Obtenir les noms de tous les acteurs et actrices ayant tourné dans un film d'Alfred Hitchcock.
4. Obtenir nom et notes moyennes de tous les réalisateurs classés par score moyen de leur film décroissant.
5. Obtenir le titre du film ayant le meilleur score, puis le casting de ce film.

Base de donnée movie

1.

```
SELECT title, score from movie WHERE votes >= 10000
    ORDER BY score DESC
```

2.

```
SELECT title FROM movie JOIN actor
    ON actor.id = director and actor.name = "Alfred Hitchcock"
```

3.

```
SELECT actor.name FROM actor JOIN casting
JOIN
(SELECT movie.id FROM movie JOIN actor
    ON actor.id = director and actor.name = "Alfred Hitchcock")
    AS hitch
ON actor.id = casting.actorid AND casting.movieid = hitch.id
```

Base de donnée movie

4.

```
SELECT name, moy FROM actor
JOIN (SELECT director, AVG(score) As moy FROM movie GROUP BY
director)
ON id = director ORDER BY moy DESC
```

5.

```
SELECT title, MAX(score) FROM movie
```

6.

```
SELECT name FROM actor JOIN casting JOIN movie
ON actor.id=actorid AND movieid = movie.id
AND movieid = (SELECT id FROM (SELECT id, MAX(score) FROM movie))
```

Exercice 5 : Enoncé

Exercice 5. *Dessins en `python`.*

Dans cette partie on travaillera sous `python` avec le module `sqlite3`. Les graphiques utiliseront la fonction `scatter()` du module `matplotlib.pyplot` :

Si `LX` et `LY` sont deux listes/tableaux de nombres de même longueur N, l'instruction : `plt.scatter(LX,LY)` représentera tous les points du plan d'abscisses respectives dans `LX` et d'ordonnées respectives dans `LY` par de petits ronds pleins (des disques).

Si `Aires` et `Couleurs` sont deux tableaux de nombres (de longueurs N), alors avec les options :

```
plt.scatter(LX, LY, s = Aires, c= Couleurs)
```

les disques seront chacun dotés d'une aire et d'une couleur.

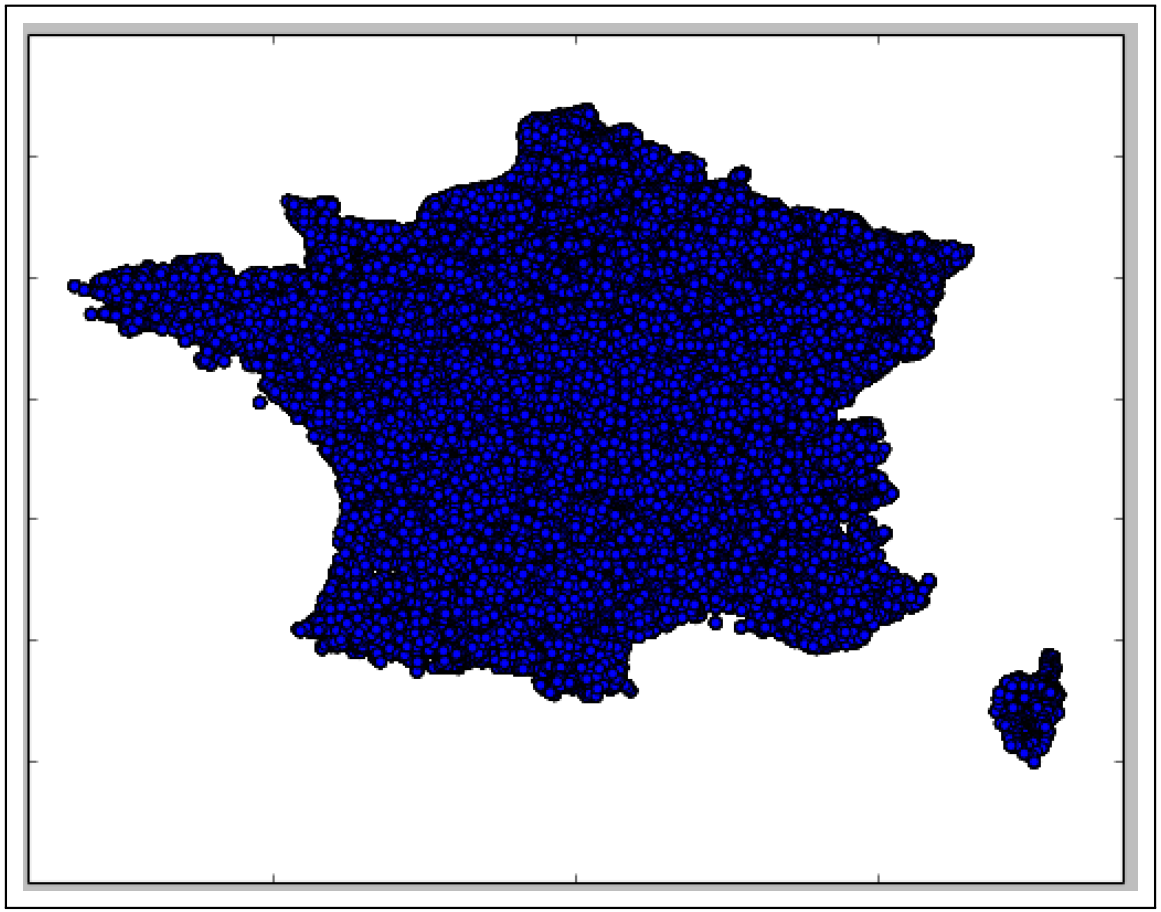
1. Tracer la carte des communes de métropole (département $\leqslant 95$).
2. Tracer la carte des 100 communes de métropole les plus peuplées. La taille de chacune sera proportionnelle à sa population
3. Tracer la carte des communes de métropoles, colorées selon leur altitude.

Exercice 5 : Corrigé

1)

```
import sqlite3
import numpy as np

bdd = sqlite3.connect("geographie.sqlite")
result = bdd.execute("SELECT longitude, latitude FROM communes WHERE
num_departement <= 95")

Lx, Ly = [], []

for ligne in result:
    Lx.append(ligne[0])
    Ly.append(ligne[1])

import matplotlib.pyplot as plt
plt.figure(1)
plt.scatter(Lx,Ly)
plt.axis('equal')
plt.show()
```

Exercice 5 : Corrigé

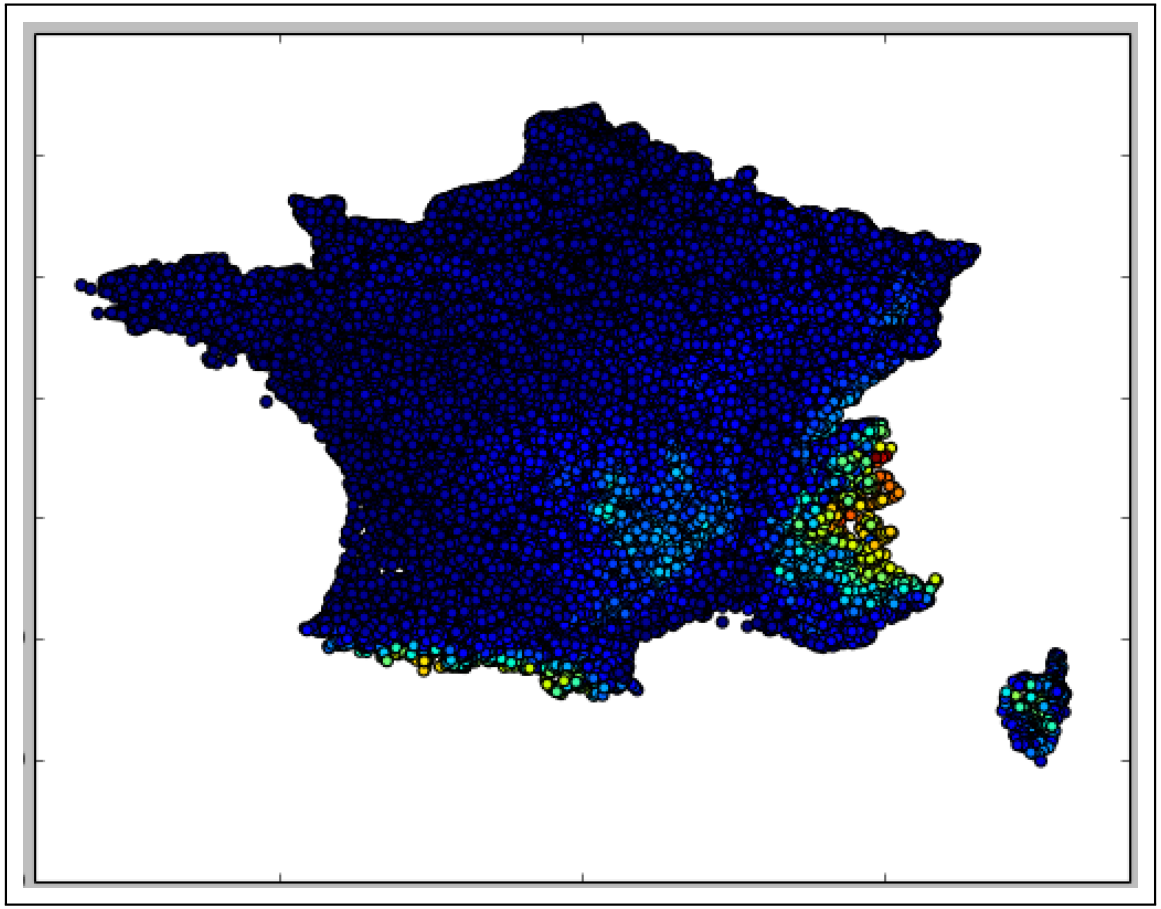


Exercice 5 : Corrigé

2)

```
result = bdd.execute("SELECT longitude, latitude, zmax FROM communes WHERE
num_departement <= 95")

Lx, Ly, Lz = [], [], []

for ligne in result:
    Lx.append(ligne[0])
    Ly.append(ligne[1])
    Lz.append(ligne[2])

plt.figure(2)
plt.clf()
plt.scatter(Lx,Ly,c=Lz )
plt.axis('equal')
plt.show()
```

Exercice 5 : Corrigé



Exercice 5 : Corrigé

3)

```
result = bdd.execute("SELECT longitude, latitude, population_2010 AS pop FROM
communes WHERE num_departement <= 95 ORDER BY pop DESC LIMIT 5000")

Lx, Ly, Lz = [], [], []

for ligne in result:
    Lx.append(ligne[0])
    Ly.append(ligne[1])
    Lz.append(ligne[2]/2000)

plt.figure(3)
plt.clf()
plt.scatter(Lx,Ly, s=Lz)
plt.axis('equal')
plt.show()
```

Exercice 5 : Corrigé