

Bases de données - 3 - Corrigé

1 Cinéma

1. Quels sont les prénoms des personnes ? Les prénoms sans doublon ? Qu'observez-vous ?

```
SELECT prenom FROM personne
SELECT DISTINCT prenom FROM personne
```

On observe que les prénoms sont rangés dans l'ordre alphabétique avec la seconde demande : ceci s'explique par le fait que pour supprimer les doublons de manière efficace en termes de complexité, le mieux est de commencer par trier ...

2. Quel est le nom des personnes dont le prénom est John ?

```
SELECT nom FROM personne WHERE prenom = 'John'
```

3. Quels sont les titres des films réalisés dans les années 90 ?

```
SELECT titre FROM film WHERE annee >= 1990 AND annee < 2000
```

4. Quelles sont les personnes qui sont des acteurs ?

```
SELECT DISTINCT nom, prenom FROM
personne JOIN jouer
ON personne.idp = jouer.ida
```

5. Quelles sont les personnes qui sont des réalisateurs ?

```
SELECT DISTINCT nom, prenom FROM
personne JOIN film
ON personne.idp = film.idr
```

6. Qui est à la fois acteur et réalisateur ?

```
SELECT DISTINCT nom, prenom FROM
personne JOIN jouer
ON personne.idp = jouer.ida
INTERSECT
SELECT DISTINCT nom, prenom FROM
personne JOIN film
ON personne.idp = film.idr
```

7. Dresser la liste de toutes les interprétations possibles, en précisant le nom et le prénom de l'acteur ainsi que le rôle et le titre du film. La liste sera triée par ordre alphabétique du nom.

```
SELECT nom, prenom, role, titre FROM
personne JOIN jouer ON idp = ida
JOIN film ON film.idf = jouer.idf
ORDER BY nom
```

8. Quels sont les titres des films où Kevin Spacey a joué un rôle ?

```
SELECT titre FROM
film JOIN jouer
ON film.idf = jouer.idf
JOIN personne
ON personne.idp = jouer.ida
WHERE nom = 'Spacey' AND prenom = 'Kevin'
```

9. Quels sont les drames que l'on a pu voir après le 1er janvier 2000 ?

```
SELECT DISTINCT titre FROM film
JOIN projection
ON film.idf = projection.idf
WHERE genre = 'Drame'
AND jour > '2000-01-01'
```

10. Dresser la liste des acteurs (nom et prénom) en précisant le nombre de films dans lesquels ils ont joué. Le résultat doit être trié par ordre alphabétique des noms.

```
SELECT nom, prenom, COUNT(*) AS nb_films FROM
personne JOIN jouer ON personne.idp = jouer.ida
GROUP BY nom, prenom
ORDER BY nom
```

11. Quels sont les acteurs ayant joué dans des drames ?

```
SELECT personne.nom, personne.prenom FROM
personne JOIN jouer
ON personne.idp = jouer.ida
JOIN film
ON jouer.idf = film.idf
WHERE genre = 'Policier'
```

12. Quels sont les titres des films où Kevin Spacey a joué un rôle et qui ont été projetés au cinéma UGC ?

```
SELECT titre FROM
film JOIN jouer
ON film.idf = jouer.idf
JOIN personne
ON personne.idp = jouer.ida
JOIN projection
ON projection.idf = film.idf
JOIN cinema
ON projection.idc = cinema.idc
WHERE personne.nom = 'Spacey'
AND personne.prenom = 'Kevin'
AND cinema.nom = 'UGC'
```

13. Combien de films différents ont été projetés à l'UGC ?

```
SELECT COUNT(*) FROM
(
SELECT DISTINCT projection.idf FROM
cinema JOIN projection
ON cinema.idc = projection.idc
WHERE cinema.nom = 'UGC'
GROUP BY projection.idf
)
```

14. Quels sont les acteurs ayant joué dans tous les films de Lars von Trier ?

On peut déjà voir comment calculer le nombre de films réalisés par Lars von Trier :

```
SELECT COUNT(*) FROM
film JOIN personne
ON film.idr = personne.idp
WHERE personne.nom = 'von_Trier'
AND personne.prenom = 'Lars'
```

On va ensuite faire une jointure entre la table des films réalisés par von Trier et les tables **personne** et **jouer** puis regrouper par personne et il restera à compter pour savoir dans combien de films de von Trier chaque acteur a joué. On teste alors l'égalité avec le nombre calculé précédemment.

```
SELECT personne.nom FROM
(
SELECT film.idf AS i FROM
personne JOIN film
ON personne.idp = film.idr
WHERE personne.nom = 'von_Trier'
)
JOIN jouer
ON jouer.idf = i
JOIN personne
ON personne.idp = jouer.ida
GROUP BY personne.idp
HAVING COUNT(*) =
(
SELECT COUNT(*) FROM
film JOIN personne
ON film.idr = personne.idp
WHERE personne.nom = 'von_Trier'
AND personne.prenom = 'Lars'
)
```

2 Colles

```
import sqlite3
con = sqlite3.connect('colles.s3db')
c = con.cursor()
```

On commence par traiter un élève à la main, par exemple celui dont l'identité est 595.
Il faut déjà chercher son nom par une requête SQL :

```
id_eleve = 595

c.execute(''SELECT nom FROM eleves WHERE ide=''+ str(id_eleve))

for a in c.fetchall():
    nom_eleve = a[0]
```

On va maintenant aller chercher ses notes de colle avec la date :

```
c.execute(''SELECT colles.date_AS_d, note FROM
colles JOIN eleves ON colles.ide=eleves.ide
WHERE eleves.ide=''+ str(id_eleve) + ''_ORDER_BY_d''')
```

On crée son fichier en écriture et on y écrit les informations obtenues par la requête :

```
fichier = open('notes_' + nom_eleve + '.txt', 'w')
fichier.write('Notes de colle de ' + nom_eleve + ':\n\n')

for (date, note) in c.fetchall():
    fichier.write(date + '\t' + str(note) + '\n')

fichier.close()
```

Il ne reste plus qu'à automatiser la chose par une fonction (*ide* désigne l'identité de l'élève et *nom* son nom que l'on obtiendra facilement) :

```
def feuille(ide, nom):

    fichier = open('notes_' + nom + '.txt', 'w')
    fichier.write('Notes de colle de ' + nom + ':\n\n')

    c.execute(''SELECT colles.date_AS_d, note FROM
colles JOIN eleves ON colles.ide=eleves.ide
WHERE eleves.ide=''+ str(ide) + ''_ORDER_BY_d''')
```

```
    for (date, note) in c.fetchall():
        fichier.write(date + '\t' + str(note) + '\n')

    fichier.close()
```

Il ne reste plus qu'à aller chercher tous les couples (*ide*, *nom*) par une requête SQL puis à appliquer la fonction à chacun de ces couples!

```
c.execute(''SELECT ide, nom FROM eleves''')

for a in c.fetchall():
    feuille(a[0], a[1])

con.commit()
con.close()
```