

## Travaux Pratiques SQL

On travaille dans le « bac à sable » SQL <https://tchou.github.io/sqlsb/>. Il y a un menu **Choisir une base de données** en haut à gauche, avec quatre bases de données déjà créées. On écrit une requête SQL dans la fenêtre en haut à droite. On l'exécute en cliquant sur le bouton **Exécuter** (le triangle) ou avec **Maj-Entrée** sur le clavier. Le résultat s'affiche dans la fenêtre en bas à droite. Dans cette fenêtre, si on clique sur une requête, elle est recopiée dans la fenêtre de saisie et on peut alors la modifier.

Dans les questions ci-dessous, les mots en **police fixe** donnent une indication sur les colonnes et les tables à utiliser dans la requête.

### 1 Base Médiathèque

Sélectionner la base de données **Médiathèque** dans le menu en haut à gauche. Voici les schémas des tables de cette base :

```
livre(titre (chaîne), editeur (chaîne), annee (entier), isbn (chaîne))
auteur(a_id (entier), nom (chaîne), prenom (chaîne))
auteur_de(a_id (entier), isbn (chaîne))
usager(nom (chaîne), prenom (chaîne), adresse (chaîne), cp (chaîne),
        ville (chaîne), email (chaîne), code_barre (chaîne))
emprunt(code_barre (chaîne), isbn (chaîne), retour (chaîne))
```

**Question 1** (requêtes simples) Donner le code SQL de chacune des requêtes ci-dessous.

1. Tous les titres de livre.
2. Tous les noms d'usager.
3. Tous les noms d'usager en retirant les doublons.
4. Les titres des livres publiés avant 1980.
5. Les titres des livres dont le titre contient la lettre « A ».
6. Les isbn des livres à rendre pour le 01/01/2020. (Les dates sont écrites au format '2020-06-29'.)
7. Les noms d'auteurs triés par ordre alphabétique.
8. Les noms d'usagers vivant dans le 12<sup>e</sup> ou 13<sup>e</sup> arrondissement de Paris (codes postaux 75012 et 75013).
9. Les noms et adresses des usagers n'habitant pas dans une rue (*i.e.*, la chaîne « Rue » ne doit pas apparaître dans l'adresse).

---

**Correction :**

1. `SELECT titre FROM livre;`
2. `SELECT nom FROM usager;`
3. `SELECT DISTINCT nom FROM usager;`

4. SELECT titre FROM livre WHERE annee <= 1980;
  5. SELECT titre FROM livre WHERE titre LIKE '%A%';
  6. SELECT isbn FROM emprunt WHERE retour = '2020-01-01';
  7. SELECT nom FROM auteur ORDER BY nom ASC;
  8. SELECT nom FROM usager WHERE cp = '75012' OR cp = '75013';
  9. SELECT nom, adresse FROM usager  
WHERE NOT (adresse LIKE '%Rue%');
- 

**Question 2** (requêtes avancées avec jointures et imbrication) Donner le code SQL de chacune des requêtes ci-dessous.

1. Le titre des livres empruntés.
  2. Le titre des livres empruntés à rendre avant le 31/03/2020.
  3. Le nom et prénom de l'auteur du livre '1984'.
  4. Le nom et le prénom des usagers ayant emprunté des livres, sans doublons (*i.e.* si un usager a emprunté plusieurs livres, il ne doit apparaître qu'une fois dans le résultat).
  5. Même requête que précédemment, avec les noms triés par ordre alphabétique.
  6. Les titre des livres publiés strictement avant 'Dune'.
  7. Les noms et prénoms des auteurs des livres trouvés à la question précédente.
  8. Comme la question précédente, en retirant les doublons.
  9. Le nombre de résultats trouvés à la question précédente.
- 

**Correction :**

1. SELECT titre FROM livre AS l  
JOIN emprunt AS e ON l.isbn = e.isbn;
2. SELECT titre FROM livre AS l  
JOIN emprunt AS e ON l.isbn = e.isbn  
WHERE e.retour <= '2020-03-31';
3. SELECT a.nom, a.prenom FROM auteur AS a  
JOIN auteur\_de AS ad ON ad.a\_id = a.a\_id  
JOIN livre AS l ON l.isbn = ad.isbn  
WHERE l.titre = '1984';
4. SELECT DISTINCT u.nom, u.prenom FROM emprunt AS e  
JOIN usager AS u  
ON e.code\_barre = u.code\_barre;
5. SELECT DISTINCT u.nom, u.prenom FROM emprunt AS e  
JOIN usager AS u  
ON e.code\_barre = u.code\_barre  
ORDER BY u.nom;

```

6. SELECT titre FROM livre
   WHERE annee < (SELECT annee FROM livre
                  WHERE titre = 'Dune');

7. SELECT a.nom,a.prenom FROM livre AS l
   JOIN auteur_de AS ad ON l.isbn = ad.isbn
   JOIN auteur AS a ON a.a_id = ad.a_id
   WHERE annee < (SELECT annee FROM livre
                  WHERE titre = 'Dune');

8. SELECT DISTINCT a.nom,a.prenom FROM livre AS l
   JOIN auteur_de AS ad ON l.isbn = ad.isbn
   JOIN auteur AS a ON a.a_id = ad.a_id
   WHERE annee < (SELECT annee FROM livre
                  WHERE titre = 'Dune');

9. SELECT COUNT (*) FROM
   (SELECT DISTINCT a.nom,a.prenom FROM livre AS l
     JOIN auteur_de AS ad ON l.isbn = ad.isbn
     JOIN auteur AS a ON a.a_id = ad.a_id
     WHERE annee < (SELECT annee FROM livre
                    WHERE titre = 'Dune')
   ) AS tmp;

```

---

## 2 Base Films

Sélectionner la base de données Films dans le menu en haut à gauche. Voici les schémas des tables de cette base :

```

personne(pid (entier), nom (chaîne), prenom (chaîne))
film(fid (entier), titre (chaîne), annee (entier),
     duree (entier), titre_orig (chaîne))
joue(fid (entier), pid (entier), role (chaîne))
genre(fid (entier), genre (chaîne))
pays(fid (entier), pays (chaîne))

```

**Question 3** (requêtes simples) Donner le code SQL de chacune des requêtes ci-dessous.

1. Les titres des films qui contiennent le mot **aventure**.
2. Les titres des films dont la durée est inférieure à 90 minutes.
3. La durée du film le plus court.
4. Tous les genres de films distincts, triés par ordre alphabétique.
5. Le nombre de genres de films distincts.
6. L'identifiant (**pid**) de l'acteur Kevin Bacon.
7. Le nombre de films dans lesquels joue Kevin Bacon.

8. La moyenne des durées des films entre les années 1960 et 1980.

---

**Correction :**

1. `SELECT titre FROM film WHERE titre LIKE '%aventure%';`
2. `SELECT titre FROM film WHERE duree<90;`
3. `SELECT MIN(duree) FROM film;`
4. `SELECT DISTINCT genre FROM genre ORDER BY genre ASC;`
5. `SELECT COUNT(DISTINCT genre) FROM genre;`
6. `SELECT * FROM personne WHERE prenom = 'Kevin' AND nom = 'Bacon';`
7. `select * from joue where pid = 2833;`  
Ici, on réutilise directement le résultat de la requête précédente, mais on pourrait faire une requête imbriquée.
8. `SELECT AVG (duree)  
FROM Film  
WHERE annee >= 1960 AND annee <= 1980;`

---

**Question 4** (requêtes avancées avec jointures et imbrication) Donner le code SQL de chacune des requêtes ci-dessous.

1. Les titres des films français.
2. Le titre du film le plus court.
3. Les noms des personnes qui ont joué dans un film sorti en 2000, sans doublons.

---

**Correction :**

1. `SELECT f.titre  
FROM film AS f  
JOIN pays AS p ON p.fid = f.fid  
WHERE p.pays = 'FRANCE';`
  2. `SELECT titre FROM film WHERE duree = (SELECT min(duree) FROM film);`
  3. `SELECT DISTINCT p.nom FROM personne AS p  
JOIN joue AS j ON j.pid = p.pid  
JOIN film AS f ON j.fid = f.fid  
WHERE f.annee = 2000;`
-