

Requêtes SQL

Voici la traduction SQL des questions principales identifiées dans le cahier des charges pour valider la structure de la base de données.

Identifier les abonnés actuellement en retard (Question 4 du sujet).....	2
Trouver les ouvrages disponibles pour un transfert vers une bibliothèque donnée (Question 1 du sujet).....	3
Top des ouvrages les plus populaires par région (Question 6 du sujet).....	4
Liste des participants à un événement futur (Question 5 du sujet).....	5

Identifier les abonnés actuellement en retard

(Question 4 du sujet)

Objectif :

Lister les utilisateurs qui n'ont pas rendu leur livre alors que la date prévue est dépassée.

Requête SQL :

```
SELECT
    A.nom,
    A.prenom,
    O.titre,
    E.date_retour_prevue
FROM EMPRUNTE E
JOIN ABONNE A ON E.id_abonne = A.id_abonne
JOIN EXEMPLAIRE EX ON E.code_barre = EX.code_barre
JOIN OUVRAGE O ON EX.id_ouvrage = O.id_ouvrage
WHERE E.date_retour_effectif IS NULL
    AND E.date_retour_prevue < CURRENT_DATE;
```

Trouver les ouvrages disponibles pour un transfert vers une bibliothèque donnée (Question 1 du sujet)

Objectif :

Pour un ouvrage donné (ex: 'Dune'), trouver quels exemplaires sont disponibles dans la même région que la bibliothèque demandeuse (ex: id_biblio = 1), mais dans d'autres bibliothèques.

Requête SQL :

```
SELECT
    EX.code_barre,
    B_Source.nom AS bibliothèque_source,
    D.temps_transport
FROM EXEMPLAIRE EX
JOIN BIBLIOTHEQUE B_Source ON EX.id_biblio = B_Source.id_biblio
JOIN EST_DISTANT D ON D.id_biblio_A = B_Source.id_biblio
JOIN OUVRAGE O ON EX.id_ouvrage = O.id_ouvrage
WHERE O.titre = 'Dune'
    AND D.id_biblio_B = 1 -- ID de la bibliothèque qui a besoin du livre
    AND EX.etat != 'Abîmé'
    AND EX.code_barre NOT IN (
        -- On exclut ceux qui sont actuellement empruntés
        SELECT code_barre FROM EMPRUNTE WHERE date_retour_effectif IS
NULL
    )
    ORDER BY D.temps_transport ASC; -- On privilégié le plus proche
```

Top des ouvrages les plus populaires par région

(Question 6 du sujet)

Objectif :

Compter le nombre d'emprunts par titre pour déterminer les tops.

Requête SQL :

```
SELECT
    O.titre,
    COUNT(E.id_emprunt) AS nombre_total_prets
FROM EMPRUNTE E
JOIN EXEMPLAIRE EX ON E.code_barre = EX.code_barre
JOIN BIBLIOTHEQUE B ON EX.id_biblio = B.id_biblio
JOIN REGION R ON B.id_region = R.id_region
JOIN OUVRAGE O ON EX.id_ouvrage = O.id_ouvrage
WHERE R.nom_region = 'Île-de-France'
GROUP BY O.titre
ORDER BY nombre_total_prets DESC
LIMIT 10;
```

Liste des participants à un événement futur (Question 5 du sujet)

Objectif :

Voir qui est inscrit aux événements à venir.

Requête SQL :

SELECT

```
EV.nom AS nom_evenement,  
EV.date_event,  
A.nom AS nom_participant,  
A.prenom  
FROM EVENT EV  
JOIN PARTICIPE P ON EV.id_event = P.id_event  
JOIN ABONNE A ON P.id_abonne = A.id_abonne  
WHERE EV.date_event > CURRENT_TIMESTAMP  
ORDER BY EV.date_event;
```