**Expression en SQL des requêtes**

DUDON – MAVEYRAUD – TAIEB – SUTKOVIC

**Question 1 :**

SELECT DISTINCT o.id\_ouvrage, o.titre, o.auteur, o.categorie  
FROM Ouvrages o  
JOIN Emplacements e ON o.id\_ouvrage = e.id\_ouvrage  
JOIN Localisations l ON e.id\_localisation = l.id\_localisation  
JOIN Bibliotheques b ON l.id\_bibliotheque = b.id\_bibliotheque  
LEFT JOIN Prets p ON o.id\_ouvrage = p.id\_ouvrage AND p.date\_retour\_effective IS NULL  
LEFT JOIN Reservations r ON o.id\_ouvrage = r.id\_ouvrage AND r.etat\_reservation = 'active'  
WHERE p.id\_pret IS NULL  -- L'ouvrage n'est pas actuellement emprunté  
AND r.id\_reservation IS NULL  -- L'ouvrage n'est pas réservé  
AND b.id\_bibliotheque <> 6  -- ID de la bibliotheque de destination  
AND o.id\_ouvrage NOT IN (  
    SELECT id\_ouvrage FROM Transferts  
    WHERE etat\_transfert = 'En cours'  
);

*Explication :*

1. ***On sélectionne les ouvrages (Ouvrages) disponibles dans le réseau****en s'assurant qu'ils ont un emplacement (Emplacements).*
2. ***On exclut les ouvrages déjà empruntés (Prêts)****en vérifiant que date\_retour\_effective IS NULL (donc toujours en cours d'emprunt).*
3. ***On exclut les ouvrages réservés (Réservations)****en filtrant ceux ayant etat\_reservation = 'active'.*
4. ***On s'assure que l'ouvrage n'est pas déjà dans la bibliothèque cible****en comparant avec id\_bibliotheque\_destination.*
5. ***On exclut les ouvrages qui sont déjà en cours de transfert (Transferts)****.*

**Question 2 :**

*1.Ajouter la nouvelle bibliothèque :*

INSERT INTO Adresses (adresse, region)  
VALUES ('123 Rue des Livres', 'Île-de-France')  
RETURNING id\_adresse;

INSERT INTO Bibliothèques (nom, id\_adresse)  
VALUES ('Bibliothèque Centrale', <id\_adresse>);

*2. Ajouter les localisations internes*

INSERT INTO Localisations (id\_bibliotheque, nom\_localisation)  
VALUES  
(<id\_localisation>, <id\_bibliotheque>, ‘Etagère G45’),  
(<id\_localisation>, <id\_bibliotheque>, ‘Etagère G46’),  
(<id\_localisation>, <id\_bibliotheque>, ‘Etagère G47’);

*3. Affecter des ouvrages disponbles a cette bibliothèque*

INSERT INTO Emplacements (id\_localisation, id\_ouvrage)  
SELECT l.id\_localisation, o.id\_ouvrage  
FROM Ouvrages o  
JOIN Localisations l ON l.id\_bibliotheque = <id\_bibliotheque>  
WHERE o.id\_ouvrage NOT IN (  
    SELECT id\_ouvrage FROM Prêts WHERE date\_retour\_effective IS NULL  
);

*4. Programmer des transferts de livres d'autres bibliothèques*

INSERT INTO Transferts (id\_ouvrage, bibliotheque\_origine, bibliotheque\_destination, date\_emission, etat\_transfert)  
SELECT id\_ouvrage, id\_bibliotheque, <id\_bibliotheque>, CURRENT\_DATE, 'En cours'  
FROM Emplacements e  
JOIN Localisations l ON e.id\_localisation = l.id\_localisation  
WHERE l.id\_bibliotheque <> <id\_bibliotheque>  
LIMIT 50; -- Limite le nombre de transferts

1. *Organiser un événement d'ouverture*

INSERT INTO Événements (titre, description, date\_evenement, id\_bibliotheque)

VALUES ('Inauguration', 'Venez découvrir notre nouvelle bibliothèque !', '2025-03-01', <id\_bibliotheque>);

**Question 3 :**

SELECT  
    o.titre,  
    COUNT(t.id\_transfert) AS nombre\_transferts,  
    AVG(t.date\_reception - t.date\_emission) AS delai\_moyen\_transfert  
FROM Transferts t  
JOIN Ouvrages o ON t.id\_ouvrage = o.id\_ouvrage  
WHERE t.date\_reception IS NOT NULL  -- Assurer que le transfert est complété  
GROUP BY o.titre  
ORDER BY nombre\_transferts DESC  
LIMIT 10; -- Afficher les 10 ouvrages les plus transférés

*Cette requête :  
✅ Compte le nombre total de transferts pour chaque ouvrage  
✅ Calcule le délai moyen (en jours) entre l'émission et la réception  
✅ Trie les ouvrages du plus transféré au moins transféré  
✅ Limite l'affichage aux 10 ouvrages les plus concernés*

**Question 4 :**

SELECT

a.id\_abonne,

a.nom,

a.prenom,

COUNT(p.id\_pret) AS nombre\_infractions

FROM Abonnes a

JOIN Prets p ON a.id\_abonne = p.id\_abonne

WHERE p.date\_retour\_effective > p.date\_retour -- Retard constaté

GROUP BY a.id\_abonne, a.nom, a.prenom

ORDER BY nombre\_infractions DESC;

*Explication :*

*✅ Sélectionne les abonnés qui ont rendu un ouvrage en retard  
✅ Compte le nombre total de retards pour chaque abonné  
✅ Trie par ordre décroissant de fréquence d’infraction*

**Question 5 :**

WITH EventsInLibrary AS (

SELECT e.id\_evenement, e.titre, e.date\_evenement, e.id\_bibliotheque

FROM Evenements e

WHERE e.id\_bibliotheque = 5 -- ID de la bibliothèque cible

),

SimilarEvents AS (

SELECT DISTINCT pe.id\_abonne, e.id\_evenement, e.titre

FROM Participation\_evenements pe

JOIN Evenements e ON pe.id\_evenement = e.id\_evenement

WHERE e.titre IN (SELECT titre FROM EventsInLibrary)

)

SELECT eil.titre AS evenement\_bibliotheque, eil.date\_evenement,

s.id\_abonne, a.nom, a.prenom, s.id\_evenement AS evenement\_similaire

FROM EventsInLibrary eil

LEFT JOIN SimilarEvents s ON eil.titre = s.titre

LEFT JOIN Abonnes a ON s.id\_abonne = a.id\_abonne

ORDER BY eil.titre, a.nom;

*Explication :*

*✅ EventsInLibrary récupère les événements d’une bibliothèque spécifique.  
✅ SimilarEvents identifie les abonnés ayant participé à des événements similaires (même titre).  
✅ Requête finale associe ces abonnés aux événements programmés dans la bibliothèque cible.*

**Question 6 :**

-- Pour obtenir les ouvrages les plus populaires dans une région spécifique

SELECT

o.titre AS ouvrage,

c.nom AS collection,

COUNT(r.id\_reservation) AS nombre\_reservations,

COUNT(p.id\_pret) AS nombre\_prets

FROM

abonnes a

JOIN

reservations r ON r.id\_abonne = a.id\_abonne

JOIN

ouvrages o ON o.id\_ouvrage = r.id\_ouvrage

JOIN

collections c ON c.id\_collection = o.id\_collection

LEFT JOIN

prets p ON p.id\_ouvrage = o.id\_ouvrage AND p.id\_abonne = a.id\_abonne -- jointure avec la table prets

JOIN

adresses ad ON ad.id\_adresse = a.id\_adresse -- jointure avec la table adresses pour la région

WHERE

ad.region = 'Île-de-France' -- Région spécifiée entre guillemets

GROUP BY

o.titre, c.nom

ORDER BY

nombre\_reservations DESC, nombre\_prets DESC

LIMIT 10; -- Limiter à 10 ouvrages les plus populaires

**Question 7 :**

SELECT t.id\_transfert, t.bibliotheque\_origine, t.bibliotheque\_destination, t.date\_emission, t.date\_reception, t.etat\_transfert  
FROM transferts t  
WHERE t.etat\_transfert = 'En attente'  
ORDER BY t.date\_emission;

*On a décidé de montrer tous les transferts qui sont « En attente » et de les prioriser pour l’envoie. Ces transferts sont urgents à réaliser.*