

Correction de l'exercice « la bibliothèque »

1. Le titre du livre et son ISBN apparaissent en double, dans la table Livre et dans la table Emprunt. Ce n'est pas nécessaire. Cela alourdit inutilement la base de données.

2. a. (Abonne_ID, Livre_ID, DateEmprunt)

b. Le SGBD n'acceptera pas de saisies erronées dans lesquelles la date de retour est avant la date d'emprunt.

c. On ne pourra pas ajouter un emprunt avec un identifiant d'abonné qui n'existe pas dans la table Abonne et on ne pourra pas supprimer un abonné de la table Abonne s'il est référencé dans la table Emprunt. Il faudra d'abord effacer tous ses emprunts (après avoir vérifié qu'il a bien rendu les livres).

d. On ne pourra plus saisir d'emprunt sans entrer la date de retour. Or, lors d'un nouvel emprunt, la date de retour n'est pas encore connue.

3. a. `SELECT Nom, Prenom FROM Abonne WHERE Ville = 'Bayonne'`

b. `SELECT Nom, Prenom FROM Abonne WHERE CodePostal LIKE '64%'`

c. `SELECT Abonne_ID FROM Emprunt WHERE
DateRetourReel > DateRetourPrevu`

d. `SELECT Nom FROM Abonne JOIN Emprunt
ON Abonne.Abonne_ID = Emprunt.Abonne_ID
WHERE DateRetourReel > DateRetourPrevu`

e. `SELECT Titre FROM Livre L JOIN Emprunt E
ON L.Livre_ID = E.Livre_ID
WHERE DateRetourReel IS NULL`

f. `SELECT Nom FROM Abonne
JOIN Emprunt ON Abonne.Abonne_ID = Emprunt.Abonne_ID
JOIN Livre ON Livre.Livre_ID = Emprunt.Livre_ID
WHERE Livre.Titre = 'Papillon'`

g. `SELECT * FROM Abonne
WHERE NOT EXISTS (SELECT * FROM Emprunt
WHERE Abonne.Abonne_ID = Emprunt.Abonne_ID)`

ou

`SELECT * FROM Abonne
WHERE Abonne_ID NOT IN (SELECT Abonne_ID FROM Emprunt)`

h. `SELECT Nom FROM Abonne WHERE Abonne_ID NOT IN
(SELECT Abonne_ID FROM Emprunt
WHERE DateRetourReel > DateRetourPrevu)`

ou

```
SELECT Nom FROM Abonne WHERE NOT EXISTS
  (SELECT * FROM Emprunt AS E
   WHERE Abonne.Abonne_ID = E.Abonne_ID
   AND E.DateRetourReel > E.DateRetourPrevu)
```

i. SELECT Nom FROM Abonne WHERE Abonne_ID NOT IN
(SELECT Abonne_ID FROM Emprunt
WHERE DateRetourReel <= DateRetourPrevu)

ou

```
SELECT Nom FROM Abonne WHERE NOT EXISTS
  (SELECT * FROM Emprunt
   WHERE Abonne.Abonne_ID = Emprunt.Abonne_ID
   AND Emprunt.DateRetourReel <= Emprunt.DateRetourPrevu)
```

j. SELECT Nom, Prenom FROM Abonne
JOIN Emprunt ON Abonne.Abonne_ID = Emprunt.Abonne_ID
WHERE Ville = 'Bayonne' AND DateRetourPrevu < DateRetourReel

k. SELECT * FROM Abonne
WHERE DateNaissance = (SELECT Min(DateNaissance) FROM Abonne)

ou

```
SELECT Abonne_ID FROM Abonne ORDER BY DateNaissance LIMIT 1
```

l. SELECT MAX(DateRetourReel-DateRetourPrevu) FROM Emprunt

Certains environnement SQL ne permettent pas d'effectuer de calculs à l'intérieur de la fonction MAX, alors on doit créer la table Temp avec la colonne retard.

```
SELECT MAX(retard) FROM (SELECT DateRetourReel-DateRetourPrevu AS retard FROM Emprunt) AS
Temp
```

m. SELECT Abonne_ID FROM Emprunt
WHERE DateRetourReel-DateRetourPrevu = (SELECT MAX(DateRetourReel-DateRetourPrevu)
FROM Emprunt)

n. SELECT AVG(DateRetourReel-DateRetourPrevu)
FROM Emprunt
WHERE DateRetourReel > DateRetourPrevu

o. SELECT * FROM Abonne WHERE nom BETWEEN 'A' AND 'T'

p. SELECT Count(*)
FROM Emprunt E

```
JOIN Livre L ON L.Livre_ID = E.Livre_ID
WHERE L.Titre = 'Papillon'
```

4. a. INSERT INTO Emprunt VALUES(803, 4252, 2020-11-02 , 2020-11-23, NULL)

b. UPDATE Abonne SET Adresse = '11 rue Lormand', Ville = 'Bayonne', CodePostal = '64100'
WHERE Nom = 'DUPONT' AND Prenom = 'Jean'

c. SELECT * FROM Emprunt E JOIN Abonne A ON E.Abonne_ID = A.Abonne_ID
WHERE Nom = 'ETCHEGARAY' AND Prenom = 'Jon' AND DateRetourReel IS NULL

d. DELETE FROM Emprunt
WHERE Abonne_ID IN (SELECT Abonne_ID
FROM Abonne
WHERE Nom = 'ETCHEGARAY' AND Prenom = 'Jon')

e. DELETE FROM Abonne WHERE Nom = 'ETCHEGARAY' AND Prenom = 'Jon'

5. SELECT * FROM Emprunt
WHERE (Abonne_ID NOT IN (SELECT Abonne_ID
FROM Abonne))
OR
(Livre_ID NOT IN (SELECT Livre_ID
FROM Livre))

Ou

```
SELECT *
FROM Emprunt
WHERE NOT EXISTS
(SELECT 1
FROM Abonne
WHERE Abonne.Abonne_ID = Emprunt.Abonne_ID)
OR NOT EXISTS
(SELECT 1
FROM Livre
WHERE Livre.Livre_ID = Emprunt.Livre_ID)
```