
REQUÊTES SQL

Exercice 0 On considère la base de données de la médiathèque introduite dans le cours/TP. Donne le code SQL de chacune des requêtes ci-dessous, sans utiliser de jointure.

1. Tous les titres de *livre*.
2. Tous les noms d'*usager*.
3. Tous les noms d'*usagers* 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.
7. Les noms d'*auteurs* triés par ordre alphabétique.
8. Les noms d'*usagers* ayant pour codes postaux 75012 et 75013.
9. Les noms et adresses des usagers n'habitant pas dans une rue (la chaîne « Rue » n'apparaît pas dans l'adresse).
10. Les années et titres des livres publiés lors d'une année bissextile (année divisible par 4, mais pas celle divisible par 100 sauf si elles sont divisibles par 400).

Exercice 1 On considère la base de données de la médiathèque introduite dans le cours/TP. Donner le code SQL de chacune des requêtes ci-dessous, en utilisant si besoin des jointures.

1. Le titre des livres empruntés.
2. Le titre des livres empruntés à rendre avant le 31/03/2020.
3. Le nom et prenom de l'auteur du livre « 1984 ».
4. Le nom et prenom des usagers ayant emprunté des livres, sans doublons.
5. Même requête que précédemment, avec les noms triés par ordre alphabétique.
6. Les titres des livres publiés strictement avant « Dune ».

Exercice 2 On considère la base de données de la médiathèque introduite dans le cours/TP. Formule simplement en français les requêtes SQL suivantes.

1. `SELECT * FROM livre WHERE titre LIKE '%Robot%' ;`
2. `SELECT nom, prenom FROM usager WHERE ville = 'Guingamp' ;`
3. `SELECT u.nom, u.prenom
FROM usager AS u
JOIN emprunt AS e ON u.code_barre = e.code_barre
WHERE retour < '2020-04-02' ;`

Exercice 3 Calculer tous les auteurs ayant collaboré sur un ouvrage et les renvoyer sous la forme (n_1, p_1, n_2, p_2, t) où les n_i sont les noms des auteurs, p_i leur prénoms et t le titre du livre sur lequel ils ont collaboré. Si trois auteurs ont collaboré sur le même livre, on souhaite avoir trois lignes de résultats ($auteur_1/auteur_2, auteur_2/auteur_3, auteur_1/auteur_3$) et non pas les trois sur la même ligne. Pour ne pas afficher deux fois le même couple, on demande en plus que $n_1 < n_2$.

Exercice 4 On considère les trois tables décrites à la figure ci-dessous. Pour chacune des requêtes SQL ci-dessous, calculer son résultat (à la main).

1. `SELECT * FROM x WHERE b > 3;`
2. `SELECT DISTINCT e FROM z WHERE e > 10 AND e < 50 ;`
3. `SELECT * FROM y WHERE c % 2 = 0 ORDER BY d ASC ;`
4. `SELECT x.a, x.b FROM x JOIN z ON z.a = x.a WHERE z.e < 9 ;`
5. `SELECT DISTINCT x.b, y.d FROM x JOIN z ON z.a = x.a ;`

```
CREATE TABLE x (a INT PRIMARY KEY, b INT, CHECK (b >= 0) );
```

```
CREATE TABLE y (c INT PRIMARY KEY, d INT, CHECK (d <= 30) );
```

```
CREATE TABLE z (a INT REFERENCES X(a), c INT REFERENCES Y(c), e INT, UNIQUE(a, c) );
```

x:		y:		z:		
a	b	c	d	a	c	e
1	1	9	9	1	11	30
2	2	10	10	2	14	9
3	2	11	9	5	15	1
4	2	12	20	7	17	3
5	1	13	30	1	10	50
6	9	14	9	2	9	8
7	1	15	1	2	15	15
		16	10	3	17	19
		17	10	4	16	12
				5	10	20
				2	11	30
				7	14	9
				7	9	12

Exercice 5 On considère les trois tables décrites à l'exercice précédent. Pour chacune des modifications ci-dessous, indiquer si elle réussit ou si elle échoue. Si elle réussit, indiquer comment la table est modifiée. Si elle échoue, expliquer pourquoi. Les questions sont indépendantes et ne tiennent pas compte des modifications effectuées par les autres questions.

1. `UPDATE x SET b = b + a ;`
2. `UPDATE x SET b = b - 2 ;`
3. `INSERT INTO z VALUES (1, 17, 1);`
4. `INSERT INTO z VALUES (1, 18, 1);`
5. `INSERT INTO z VALUES (1, 10, 1);`
6. `DELETE FROM y WHERE c >= 12 AND c <= 13;`
7. `DELETE FROM y WHERE c >= 12 AND c <= 14;`
8. `INSERT INTO y VALUES (40, 20);`
9. `INSERT INTO y VALUES (20, 40);`
10. `DELETE FROM z WHERE a % 2 = 0 OR c % 2 = 0 OR e % 2 = 0;`