

Requêtes SQL

Les exercices 1 à 4 font références à la base de données CDI réalisée en classe. Les mots en **police fixe** donne une indication sur les tables et attributs à utiliser dans la requête.

Exercice 1. Requêtes simples.

Donner le code SQL de chacune des requêtes suivantes.

- Afficher tous les **noms** des **auteurs**.
- Afficher le **titre** de tous les **livres**.
- Afficher les **noms** des **classes** du lycée sans doublon.
- Afficher les **titres** des **livres** et les **annees** d'édition classé selon l'**annee**.
- Quelles sont les **livres** dont le **titre** contient le mot *Astérix* ?

Exercice 2. Expressions et fonctions.

Donner le code SQL de chacune des requêtes suivantes.

- Afficher Les **noms** et **prénoms** des **élèves** de la **classe** 1G-1.
- Afficher les **titres** des **livres** publiés après 2000.
- Afficher les **isbn** des **livres** dont la date retour est déjà passée.
- Combien d'**auteurs** sont présents dans la base de données ?
- Quelle est l'**annee** d'édition du ou des **livre(s)** le(s) plus ancien(s) ?

Exercice 3. Jointure.

Donner le code SQL de chacune des requêtes suivantes en utilisant la clause JOIN.

- Afficher les **titres** des **livres** empruntés.
- Afficher, sans doublon, le **nom** et **prénom** des **élèves** qui ont emprunté au moins un **livre**.
- Afficher, sans doublon, le **nom** et **prénom** des **élèves** qui ont emprunté au moins un **livre** avec une date retour dépassée.
- Qui est l'auteur du **livre** 1984 ?
- Quels sont les **editeurs** à avoir édité un **livre** contenant *Astérix* dans le **titre** ?
- Afficher les **titres** des **livres** écrit par *Pierre Boulle*.
- Combien de **livres** ont été écrit par *Isaac Asimov* ?
- Afficher les **noms** des **editeurs** ayant édité un **livre** écrit par *Barjavel*.

Exercice 4.

Formuler en français les requêtes SQL suivante.

- `SELECT * FROM livres WHERE titre LIKE '%Robot%';`
- `SELECT nom, prenom FROM eleves WHERE classe = 'TG-2';`
- `SELECT el.nom, el.prenom
FROM eleves AS el
JOIN emprunt AS em ON el.num_etu = em.num_etu
WHERE date_ret < '2020-12-25';`
- `SELECT l.titre
FROM livres AS l
WHERE l.isbn IN (SELECT isbn FROM livres WHERE annee > 1990);`

Réécrire la requête en utilisant une seule clause **SELECT**.

Pour les exercices 6 et 7, on considère les tables décrite ci-dessous.

```
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,
                  PRIMARY KEY(a,c));
```

x:

a	b
1	1
2	2
3	2
4	2
5	1
6	9
7	1

y:

c	d
9	9
10	10
11	9
12	20
13	30
14	9
15	1
16	10
17	10

z:

a	c	e
1	11	30
2	14	9
5	15	1
7	17	3
1	10	50
2	9	8
2	15	15
3	17	19
4	16	12
5	10	20
2	11	30
7	14	9
7	9	12

Exercice 5.

Pour chacune des requêtes suivantes, calculer son résultat à la main.

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

Exercice 6.

Pour chacune des modifications suivantes, indiquer si elle réussit ou elle échoue. Si elle réussit indiquer comment la table est modifier. Si elle échoue, expliquer pourquoi.

- `UPDATE x SET b = b + a;`
- `UPDATE x SET b = b - 2;`
- `INSERT INTO z VALUES (1, 17, 1);`
- `INSERT INTO z VALUES (1, 18, 1);`
- `INSERT INTO z VALUES (1, 10, 1);`
- `DELETE FROM y WHERE c >= 12 AND c <= 13;`

g. `DELETE FROM y WHERE c >= 12 AND c <= 14;`

h. `INSERT INTO y VALUES (40, 20);`

i. `INSERT INTO y VALUES (20, 40);`

j. `DELETE FROM z WHERE a % 2 = 0 OR c % 2 = 0 OR e % 2 = 0;`

Ce document est mis à disposition selon les termes de la licence Creative Commons “Attribution - Pas d’utilisation commerciale - Partage dans les mêmes conditions 3.0 non transposé”.



Auteur : Pascal Seckinger