NSI · TERMINALE / COURS

Le langage SQL

Astrolab Cours · astrolabcours.fr

01 Le problème

La base de la bibliothèque contient des milliers de livres. On veut la liste des romans parus après 2000, le nombre d'emprunts par lecteur, ou le nom de ceux qui ont emprunté un livre précis. Il faut un langage pour poser ces questions.

01 L'idée : la commande au restaurant

SQL, c'est passer commande à un serveur très précis : tu dis **quoi** tu veux (les colonnes), **où** le chercher (la table) et **à quelles conditions** (les filtres). Tu ne dis pas comment aller en cuisine : le SGBD s'en charge.

01 Consulter : SELECT

```sql
SELECT titre, annee
FROM Livre
WHERE auteur = 'Hugo' AND annee > 1850
ORDER BY annee DESC;
```

- `SELECT` : les colonnes voulues (`*` pour toutes) ;
- `FROM` : la table ;
- `WHERE` : la condition sur les lignes ;
- `ORDER BY` : le tri (`ASC` croissant, `DESC` décroissant).

`SELECT DISTINCT auteur FROM Livre` donne chaque auteur une seule fois.

Compter et calculer

```sql
SELECT COUNT(*) FROM Emprunt WHERE id_lecteur = 12;
SELECT AVG(annee), MIN(annee), MAX(annee) FROM Livre;
```

`COUNT`, `SUM`, `AVG`, `MIN` et `MAX` résument une colonne en une seule valeur.

Croiser deux tables : la jointure

Pour avoir le **titre** des livres empruntés par Léa, il faut relier `Emprunt` à `Livre` et à `Lecteur` :

© Astrolab Cours · astrolabcours.fr/nsi/terminale/langage-sql/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

```sql
SELECT Livre.titre
FROM Emprunt
JOIN Livre ON Emprunt.id_livre = Livre.id_livre
JOIN Lecteur ON Emprunt.id_lecteur = Lecteur.id_lecteur
WHERE Lecteur.nom = 'Léa';
```

La condition `ON` relie la clé étrangère à la clé primaire correspondante.

Modifier les données

```sql
INSERT INTO Lecteur VALUES (57, 'Sami', '3 rue Haute');
UPDATE Lecteur SET adresse = '8 rue Basse' WHERE id_lecteur = 57;
DELETE FROM Emprunt WHERE date < '2020-01-01';
```

Exemple guidé

Tables : `Livre(id, titre, annee)`, `Lecteur(id, nom)`, `Emprunt(id_livre, id_lecteur, date)`. Qui a emprunté « Germinal » ?

```sql
SELECT Lecteur.nom
FROM Emprunt
JOIN Livre
  ON Emprunt.id_livre = Livre.id
JOIN Lecteur
  ON Emprunt.id_lecteur = Lecteur.id
WHERE Livre.titre = 'Germinal';
```

1. On part de la table `Emprunt`, qui relie livres et lecteurs.
2. Deux **jointures** retrouvent le titre et le nom grâce aux clés étrangères.
3. `WHERE` filtre sur le titre.

Nombre d'emprunts par lecteur : `SELECT id_lecteur, COUNT(*) FROM Emprunt GROUP BY id_lecteur;`

Retour au problème

SQL permet de poser ces questions en une requête : `SELECT titre FROM Livre WHERE annee > 2000;` pour les livres récents, `COUNT` pour compter, `JOIN` pour croiser les tables.

Les pièges

- `UPDATE` ou `DELETE` sans `WHERE` : toute la table est touchée.
- Les textes s'écrivent entre apostrophes : `'Hugo'`.
- Dans une jointure, préciser la table quand deux colonnes ont le même nom : `Livre.id_livre`.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/langage-sql/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits