

# Exercices SQL - Bases de Données

---



## Exercice 1 : Gestion d'une Bibliothèque

### Structure des Tables

#### Table auteurs

```
CREATE TABLE auteurs (  
    id_auteur INT PRIMARY KEY AUTO_INCREMENT,  
    nom VARCHAR(100) NOT NULL,  
    prenom VARCHAR(100) NOT NULL,  
    date_naissance DATE,  
    nationalite VARCHAR(50)  
);
```

#### Table livres

```
CREATE TABLE livres (  
    id_livre INT PRIMARY KEY AUTO_INCREMENT,  
    titre VARCHAR(200) NOT NULL,  
    id_auteur INT,  
    categorie VARCHAR(50),  
    isbn VARCHAR(13),  
    annee_publication INT,  
    disponible BOOLEAN DEFAULT TRUE,  
    FOREIGN KEY (id_auteur) REFERENCES auteurs(id_auteur)  
);
```

#### Table emprunts

```
CREATE TABLE emprunts (  
    id_emprunt INT PRIMARY KEY AUTO_INCREMENT,  
    id_livre INT,  
    nom_emprunteur VARCHAR(100) NOT NULL,  
    date_emprunt DATE NOT NULL,  
    date_retour_prevue DATE NOT NULL,  
    date_retour_effective DATE,  
    FOREIGN KEY (id_livre) REFERENCES livres(id_livre)  
);
```

## Requêtes Principales

### 1. Afficher tous les livres disponibles

```
SELECT
    l.id_livre,
    l.titre,
    l.categorie,
    CONCAT(a.prenom, ' ', a.nom) AS auteur,
    l.annee_publication
FROM livres l
JOIN auteurs a ON l.id_auteur = a.id_auteur
WHERE l.disponible = TRUE
ORDER BY l.titre;
```

### 2. Trouver les livres d'un auteur spécifique

```
SELECT
    l.titre,
    l.categorie,
    l.annee_publication,
    CASE
        WHEN l.disponible = TRUE THEN 'Disponible'
        ELSE 'Emprunté'
    END AS statut
FROM livres l
JOIN auteurs a ON l.id_auteur = a.id_auteur
WHERE a.nom = 'Hugo' AND a.prenom = 'Victor'
ORDER BY l.annee_publication;
```

### 3. Lister les livres empruntés avec les dates de retour

```
SELECT
    l.titre,
    CONCAT(a.prenom, ' ', a.nom) AS auteur,
    e.nom_emprunteur,
    e.date_emprunt,
    e.date_retour_prevue,
    e.date_retour_effective,
    CASE
        WHEN e.date_retour_effective IS NULL THEN 'En cours'
        ELSE 'Retourné'
    END AS statut_emprunt,
    CASE
        WHEN e.date_retour_effective IS NULL
            AND e.date_retour_prevue < CURDATE()
        THEN CONCAT(DATEDIFF(CURDATE(), e.date_retour_prevue), ' jours de retard')
        ELSE '-'
    END AS retard
FROM emprunts e
JOIN livres l ON e.id_livre = l.id_livre
JOIN auteurs a ON l.id_auteur = a.id_auteur
ORDER BY e.date_emprunt DESC;
```

### 4. Compter le nombre total de livres par catégorie

```
SELECT
    categorie,
    COUNT(*) AS nombre_total,
    SUM(CASE WHEN disponible = TRUE THEN 1 ELSE 0 END) AS nombre_disponibles,
    SUM(CASE WHEN disponible = FALSE THEN 1 ELSE 0 END) AS nombre_empruntes
FROM livres
GROUP BY categorie
ORDER BY nombre_total DESC;
```