

# Documentation de la Base de Données - Système de Bibliothèque

Mohamed Karim BENKIRANE

7 avril 2025

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Architecture Globale</b>	<b>2</b>
<b>3</b>	<b>Structure Détaillée</b>	<b>2</b>
3.1	Tables Principales . . . . .	2
3.1.1	Table <code>book</code> . . . . .	2
3.1.2	Table <code>author</code> . . . . .	2
3.1.3	Table <code>book_author</code> . . . . .	3
3.1.4	Table <code>section</code> . . . . .	3
3.1.5	Table <code>domain</code> . . . . .	3
3.2	Tables de Gestion des Utilisateurs . . . . .	3
3.2.1	Table <code>user</code> . . . . .	3
3.2.2	Table <code>reservation</code> . . . . .	4
3.2.3	Table <code>book_review</code> . . . . .	4
<b>4</b>	<b>Choix de Conception</b>	<b>4</b>
4.1	Relations Many-to-Many . . . . .	4
4.2	Hierarchie Domaines-Sections-Livres . . . . .	5
<b>5</b>	<b>Contraintes et Intégrité</b>	<b>5</b>
<b>6</b>	<b>Considérations de Performance</b>	<b>5</b>
<b>7</b>	<b>Évolutions Futures Possibles</b>	<b>5</b>

# 1 Introduction

Ce document présente la conception de la base de données du système de bibliothèque. La base de données a été conçue pour gérer efficacement les livres, les auteurs, les sections, et les interactions des utilisateurs avec la bibliothèque.

## 2 Architecture Globale

La base de données est structurée en plusieurs tables principales qui gèrent :

- Les livres et leurs métadonnées
- Les auteurs et leurs relations avec les livres
- L'organisation des livres en sections et domaines
- Les interactions des utilisateurs (réservations, avis)

## 3 Structure Détaillée

### 3.1 Tables Principales

#### 3.1.1 Table book

- **Description** : Stocke les informations de base des livres
- **Attributs** :
  - `id` (int, PK) : Identifiant unique du livre
  - `section_id` (int, FK) : Référence à la section
  - `title` (varchar(255)) : Titre du livre
- **Relations** :
  - Many-to-Many avec `author` via `book_author`
  - Many-to-One avec `section`

#### 3.1.2 Table author

- **Description** : Gère les informations des auteurs
- **Attributs** :
  - `id` (int, PK) : Identifiant unique de l'auteur
  - `name` (varchar(255)) : Nom de l'auteur
- **Relations** :
  - Many-to-Many avec `book` via `book_author`

### 3.1.3 Table book\_author

- **Description** : Table de liaison entre livres et auteurs
- **Attributs** :
  - `book_id` (int, PK, FK) : Référence au livre
  - `author_id` (int, PK, FK) : Référence à l’auteur
- **Relations** :
  - Many-to-One avec `book`
  - Many-to-One avec `author`

### 3.1.4 Table section

- **Description** : Organise les livres en sections thématiques
- **Attributs** :
  - `id` (int, PK) : Identifiant unique de la section
  - `domain_id` (int, FK) : Référence au domaine
  - `name` (varchar(255)) : Nom de la section
- **Relations** :
  - Many-to-One avec `domain`
  - One-to-Many avec `book`

### 3.1.5 Table domain

- **Description** : Représente les grands domaines de classification
- **Attributs** :
  - `id` (int, PK) : Identifiant unique du domaine
  - `name` (varchar(255)) : Nom du domaine
- **Relations** :
  - One-to-Many avec `section`

## 3.2 Tables de Gestion des Utilisateurs

### 3.2.1 Table user

- **Description** : Gère les informations des utilisateurs
- **Attributs** :
  - `id` (int, PK) : Identifiant unique de l’utilisateur
  - `email` (varchar(180)) : Email de l’utilisateur

- `roles` (json) : Rôles de l'utilisateur
- `password` (varchar(255)) : Mot de passe hashé
- `is_verified` (tinyint(1)) : Statut de vérification

### 3.2.2 Table reservation

- **Description** : Gère les réservations de livres
- **Attributs** :
  - `id` (int, PK) : Identifiant unique de la réservation
  - `book_id` (int, FK) : Livre réservé
  - `user_id` (int, FK) : Utilisateur qui réserve
  - `reservation_date` (datetime) : Date de réservation
  - `status` (varchar(255)) : Statut de la réservation

### 3.2.3 Table book\_review

- **Description** : Gère les avis sur les livres
- **Attributs** :
  - `id` (int, PK) : Identifiant unique de l'avis
  - `book_id` (int, FK) : Livre concerné
  - `user_id` (int, FK) : Utilisateur qui donne l'avis
  - `rating` (int) : Note donnée
  - `comment` (text) : Commentaire
  - `created_at` (datetime) : Date de création

## 4 Choix de Conception

### 4.1 Relations Many-to-Many

La relation entre les livres et les auteurs est gérée via une table de liaison (`book_author`) pour permettre :

- Un livre peut avoir plusieurs auteurs
- Un auteur peut avoir écrit plusieurs livres
- Une meilleure normalisation des données
- Une gestion plus efficace des mises à jour

## 4.2 Hiérarchie Domaines-Sections-Livres

L'organisation hiérarchique des livres est structurée en trois niveaux :

- **Domaines** : Catégories principales
- **Sections** : Sous-catégories dans chaque domaine
- **Livres** : Organisés dans les sections appropriées

## 5 Contraintes et Intégrité

- Clés étrangères pour maintenir l'intégrité référentielle
- Contraintes NOT NULL sur les champs essentiels
- Index sur les clés étrangères pour optimiser les performances
- Encodage UTF-8 pour supporter les caractères spéciaux

## 6 Considérations de Performance

- Index sur les clés étrangères
- Index sur les champs fréquemment utilisés dans les recherches
- Optimisation des jointures via la structure relationnelle
- Gestion efficace des relations many-to-many

## 7 Évolutions Futures Possibles

- Ajout de métadonnées supplémentaires pour les livres
- Extension du système de notation et d'avis
- Gestion des emprunts et retours
- Système de recommandations
- Gestion des éditeurs et des collections