## **Короткий виклад вимог**

Ця база даних призначена для системи керування бібліотекою.

**Потреби зацікавлених сторін:**

**Бібліотекарі** повинні мати змогу каталогізувати книги та їхніх авторів, керувати фізичними примірниками книг, реєструвати читачів, а також відстежувати видачу та повернення книг.

**Читачі**  повинні мати змогу знаходити книги, брати їх в оренду, ставати в чергу на популярні книги та отримувати сповіщення (наприклад, про повернення простроченої книги або про те, що книга зі списку очікування стала доступною).

**Дані для зберігання:**

* + Інформація про авторів (ім'я).
  + Інформація про книги (назва, автор, жанр, мова, рік видання).
  + Дані про конкретні фізичні примірники кожної книги.
  + Дані про читачів (ім'я, прізвище, контактна інформація).
  + Історія видачі книг (хто, що, коли взяв і повернув).
  + Списки очікування на книги.
  + Записи про боржників.
  + Сповіщення для читачів.

**Бізнес-правила:**

* + Один автор може написати багато книг, але кожна книга має лише одного автора.
  + Одна книга може мати багато фізичних примірників.
  + Читач може взяти в оренду кілька примірників книг.
  + Якщо всі примірники книги зайняті, читач може стати в чергу (список очікування).
  + Система відстежує прострочені книги та позначає відповідних читачів як боржників.
  + Система надсилає сповіщення читачам у певних ситуаціях.

## **Сутності та зв'язки**

### **Список сутностей**

**AUTHOR**: Зберігає інформацію про авторів книг.

* + author\_id: Унікальний ідентифікатор автора (первинний ключ).
  + name: Ім'я автора.

**BOOK**: Зберігає інформацію про літературні твори.

* + book\_id: Унікальний ідентифікатор книги (первинний ключ).
  + title: Назва книги.
  + author\_id: Ідентифікатор автора (зовнішній ключ до AUTHOR).
  + genre: Жанр книги.
  + language: Мова книги.
  + year\_published: Рік видання.
  + pages: Кількість сторінок.

**BOOK\_COPY**: Представляє конкретний фізичний примірник книги.

* + book\_copy\_id: Унікальний ідентифікатор примірника (первинний ключ).
  + book\_id: Ідентифікатор книги, до якої належить цей примірник (зовнішній ключ до BOOK).
  + copy\_number: Інвентарний номер примірника.

**PATRON**: Зберігає інформацію про зареєстрованих користувачів бібліотеки.

* + patron\_id: Унікальний ідентифікатор читача (первинний ключ).
  + first\_name, last\_name: Ім'я та прізвище читача.
  + contact\_data: Контактна інформація (наприклад, email або телефон).

**CHECKOUT** : Запис про факт видачі примірника книги читачеві.

* + checkout\_id: Унікальний ідентифікатор видачі (первинний ключ).
  + start\_time, end\_time: Дата початку та кінця оренди.
  + book\_copy\_id: Ідентифікатор виданого примірника (зовнішній ключ до BOOK\_COPY).
  + patron\_id: Ідентифікатор читача, який взяв книгу (зовнішній ключ до PATRON).
  + is\_borrowing: Прапорець, що вказує, чи книга наразі на руках у читача.

**WAITLIST** : Запис про те, що читач чекає на певну книгу.

* + waitlist\_id: Унікальний ідентифікатор запису (первинний ключ).
  + patron\_id: Ідентифікатор читача в черзі (зовнішній ключ до PATRON).
  + book\_id: Ідентифікатор бажаної книги (зовнішній ключ до BOOK).

**NOTIFICATION** : Зберігає текст сповіщень для читачів.

* + notification\_id: Унікальний ідентифікатор сповіщення (первинний ключ).
  + patron\_id: Ідентифікатор отримувача (зовнішній ключ до PATRON).
  + contents: Текст сповіщення.

**DEBTORS**: Запис про прострочену книгу.

* + patron\_id, book\_id: Ідентифікатори читача та книги (складений первинний ключ).
  + due\_date: Кінцева дата повернення.

### **Пояснення зв'язків**

**Author ||--|{ Book (writes)**: **Один-до-багатьох**. Один автор може написати багато книг, але кожна книга в цій системі має лише одного автора.

**Book ||--o{ BookCopy (has)**: **Один-до-багатьох**. Одна книга (наприклад, "Кобзар") може мати нуль або більше фізичних примірників у бібліотеці.

**Patron ||--o{ Checkout та BookCopy ||--o{ Checkout ( borrowed\_in)**: **Багато-до-багатьох**. Цей зв'язок реалізовано через проміжну таблицю CHECKOUT. Один читач може здійснити багато видач, і один примірник книги може бути виданий багато разів (у різний час).

**Patron ||--o{ Waitlist та Book ||--o{ Waitlist (requested)** **Багато-до-багатьох**. Реалізовано через таблицю WAITLIST. Один читач може бути в списку очікування на багато книг, і на одну книгу може чекати багато читачів.

**Patron |o--o{ Notification (receives)**: **Один-до-багатьох**. Один читач може отримати нуль або більше сповіщень.

**Checkout ||--|| Debtors (overdue)**: **Один-до-одного** (або **Один-до-нуль-або-одного**). Кожен запис про борг пов'язаний з однією конкретною видачею книги.

**Debtors/Waitlist ||--|| Notification (receives)**: **Один-до-багатьох**. Один запис у списку очікування або списку боржників може згенерувати кілька сповіщень (наприклад, перше попередження про прострочення, друге тощо).

## **Припущення та обмеження**

**Один автор на книгу**: Схема передбачає, що у кожної книги є лише один автор. Для підтримки співавторства знадобиться проміжна таблиця Book\_Author.

**Логіка списку очікування**: Читачі стають у чергу на **книгу** (book\_id), а не на **конкретний примірник** (book\_copy\_id). Це стандартна практика, і ваша сутність WAITLIST реалізована правильно, хоча лінія на діаграмі помилково веде від BookCopy.

**Ідентифікація боргу**: Сутність DEBTORS використовує (patron\_id, book\_id) як ключ. Це може бути проблемою, якщо один читач позичить ту саму книгу (різні примірники або в різний час) і прострочить обидва повернення. Більш надійно було б пов'язувати борг безпосередньо з checkout\_id.

**Структура контактних даних**: Поле contact\_data у таблиці PATRON є простим рядком. Це ускладнює валідацію та використання окремих компонентів (наприклад, email та телефон). Краще було б мати окремі поля: email, phone\_number.

**Відсутність штрафів**: Система відстежує боржників, але не має функціоналу для розрахунку та зберігання штрафів за прострочення.