# **Створення БД засобами обраної СУБД (PostgreSQL)**

# **1. Завдання**

Потрібно створити програмний засіб для співробітників бібліотеки. Така система повинна забезпечувати зберігання відомостей про наявні в бібліотеці книги, про читачів бібліотеки та читальні зали.

Для кожної книги у БД повинні зберігатися такі відомості: назва книги, автор(и), видавництво, рік видання, кількість примірників цієї книги у кожному залі бібліотеки, а також шифр книги та дата закріплення книги за читачем.

Відомості про читачів бібліотеки повинні включати номер читацького квитка, прізвище читача, номер паспорта, дату народження, адресу, номер телефону, освіту, наявність наукового ступеня. Читачі закріплюються за певною залою і можуть записуватися та виписуватися з бібліотеки.

Бібліотека має кілька читальних залів, що характеризуються номером, назвою та місткістю, тобто кількістю людей, які можуть одночасно працювати у залі.

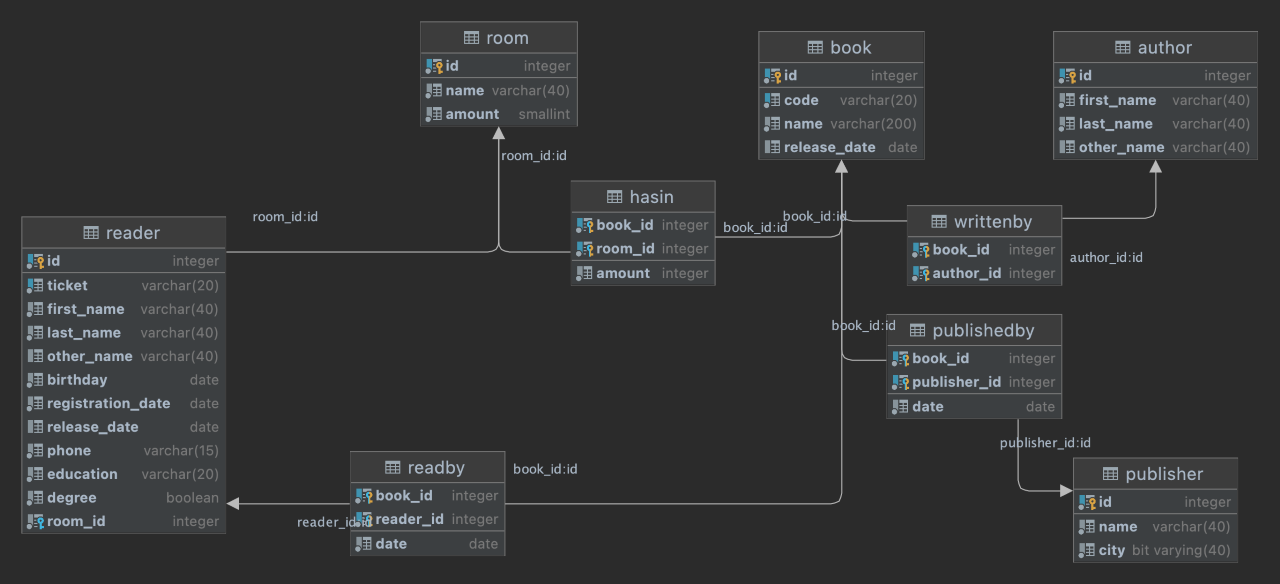
Бібліотека може отримувати нові книги та списувати старі. Шифр книги може змінитися внаслідок перекласифікації, а номер читацького квитка внаслідок перереєстрації.

Бібліотекарові можуть знадобитися такі відомості про поточний стан бібліотеки:

* Які книги закріплені за певним читачем?
* Як називається книга із заданим шифром?
* Який шифр у книги із заданою назвою?
* Коли книжку було закріплено за читачем?
* Хто з читачів узяв книгу понад місяць тому?
* За ким із читачів закріплено книги, кількість примірників яких у бібліотеці не перевищує 2?
* Яка кількість читачів користується бібліотекою?
* Скільки у бібліотеці читачів молодших 20 років?

**2. Логічне проектування бази даних**

За допомогою результатів проектування БД були визначені сутності та зв’язки між ними, була створена схема бази даних, яка зображена на малюнку нижче:



Малюнок 1. Логічна схема бази даних

Для кожної сутності була створена таблиця з первинним ключем – унікальним номером, що генерується автоматично при додаванні запису. Також для таблиць «Читач» та «Книга» були створені унікальні індекси для номера читацького квитка та шифру книги, щоб забезпечити їхню унікальність. Зв'язок 1-N (Читач - Читальна зала) був представлений за допомогою зовнішнього ключа в таблиці «Читач» (room\_id).

Всі інші зв'язки є N-M, тому були представлені як окремі таблиці зі складним первинним ключем — двома ідентифікаторами, кожен з яких є зовнішнім ключем до пов'язаних таблиць.

**3. Запити до бази даних**

**Які книги закріплені за певним читачем?**

**select** b.name, rb.date

**from** book b, reader r, readby rb

**where** rb.book\_id = b.id **and** rb.reader\_id = r.id **and** r.ticket = 'T12345'

Даний запит повертає перелік книг та дат, коли читач з квитком 'T12345' взяв ці книги в користування. У запиті використовується спрощена нотація — без явного використання операції INNER JOIN, що підтримується багатьма СУБД. Цей запит є еквівалентом наступного:

**select** b.name, rb.date

**from** book b

**inner join** readby rb **on** rb.book\_id = b.id

**inner join** reader r **on** r.id = rb.reader\_id

**where** r.ticket = 'T12345'

**Який шифр у книги із заданою назвою?**

**select** code, name **from** book **where** name **like** '%Поттер%'

У цьому запиті ми використовуємо оператор LIKE, тому що повна назва досить часто невідома та пошук здійснюється по частині назви

**Коли книжку було закріплено за читачем?**

**select** b.name, rb.date

**from** book b, reader r, readby rb

**where** rb.book\_id = b.id

**and** rb.reader\_id = r.id

**and** r.ticket = 'T12345' **and** b.code = 'A1B2'

Частково цей запит дублює перший, але у цьому ми беремо конкретну дату для конкретної книги (за кодом) та читачем (за квитком)

**Хто з читачів взяв книгу понад місяць тому?**

**select** r.first\_name, r.last\_name, rb.date

**from** reader r, readby rb

**where** rb.reader\_id = r.id

**and** rb.date <= current\_date - interval '1 month'

**За ким із читачів закріплено книги, кількість примірників яких у бібліотеці не перевищує 2?**

**select** max(b.name), sum(h.amount)

**from** Book b, Room r, hasin h

**where** h.book\_id = b.id

**and** h.room\_id = r.id **group by** b.code **having** sum(h.amount) <= 2

В даному запиті використовується операція групування по книгам (коду книги), за для того, щоб обчислити загальну кількість книг в усіх залах. Так як код книги унікальний, то в кожної групі будуть книги з однією та й самою назвою, тому функція *max* не має жодного впливу на вибір назви.

**Яка кількість читачів користується бібліотекою?**

**select** count(id) **from** reader **where** release\_date is null;

В цьому запиті ми обчислюємо кількість активних читачів, без урахування тих, що вже були виписані з бібліотеки.

**Скільки у бібліотеці читачів молодших за 20 років?**

**select** count(id) **from** reader

**where** release\_date is null **and** now() - birthday <= interval '20 years'

Запрос є розширенням попереднього, є додатковий крітерії – вік читача.

**4. Висновки**

У цієй роботі ми розробили логічну модель бази даних, створили БД засобами обраної СУБД (PostgreSQL) та зробили декілька запитів, що демонструють можливості використання створеної БД.