



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 3

з дисципліни “Бази даних”

тема “Засоби оптимізації роботи СУБД PostgreSQL”

Виконав
студент II курсу
групи КП-01
Аванесов Тігран Каренович
Перевірів
“__” “_____” 20__р
викладач

Мета

Метою роботи є здобуття практичних навичок використання засобів оптимізації СУБД PostgreSQL.

Постановка задачі

1. Перетворити модуль “Модель” з шаблону MVC лабораторної роботи No2 у вигляд об’єктно-реляційної проекції (ORM).
2. Створити та проаналізувати різні типи індексів у PostgreSQL.
3. Розробити тригер бази даних PostgreSQL.

Варіант №18

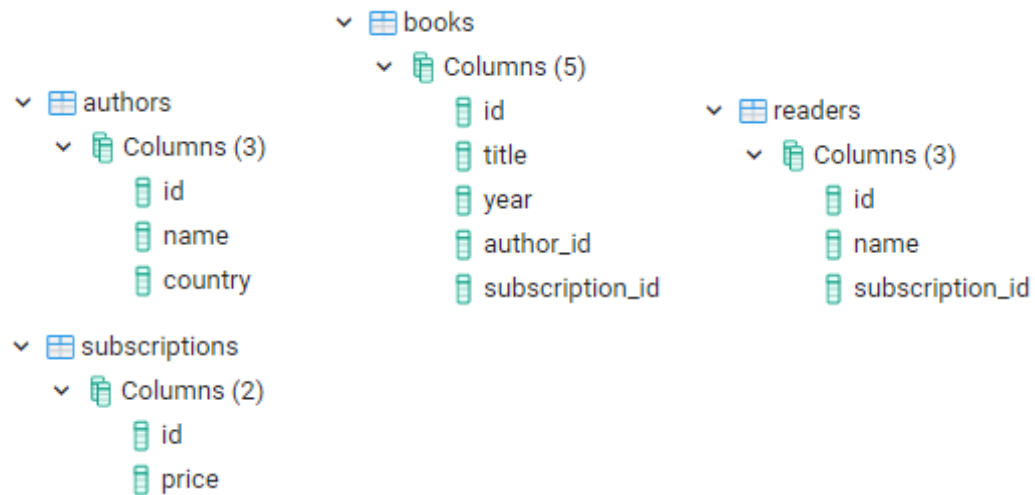
Індекси: BTree, GIN

Тригери: after update, insert

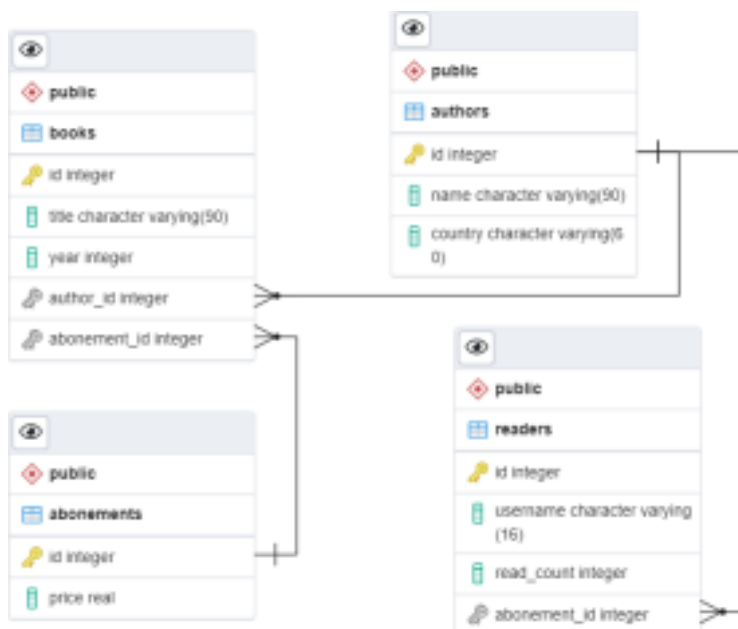
Результат роботи

Завдання №1

Таблиці бази даних:



Зв'язки між таблицями



Класи ORM для таблиць:

```

class Author(Base):
    __tablename__ = 'authors'
    id = Column(Integer, primary_key=True, unique=True, nullable=False)
    name = Column(String(90))
    country = Column(String(60))
    books = relationship('Book')
    __table_args__ = {'extend_existing': True}
  
```

```
class Book(Base):
    __tablename__ = 'books'
    id = Column(Integer, primary_key=True, unique=True, nullable=False)
    title = Column(String(90))
    year = Column(Integer)
    author_id = Column(Integer, ForeignKey('authors.id'))
    subscription_id = Column(Integer, ForeignKey('subscriptions.id'))
    __table_args__ = {'extend_existing': True}
```

```
class Reader(Base):
    __tablename__ = 'readers'
    id = Column(Integer, primary_key=True, unique=True, nullable=False)
    name = Column(String(60))
    subscription_id = Column(Integer, ForeignKey('subscriptions.id'))
    __table_args__ = {'extend_existing': True}
```

```
class Subscription(Base):
    __tablename__ = 'subscriptions'
    id = Column(Integer, primary_key=True, unique=True, nullable=False)
    price = Column(Float)
    __table_args__ = {'extend_existing': True}
```

Приклад запиту у ORM:

```
Enter command (create, update, delete): create
Enter table name (authors, books, subscriptions, readers): authors
Enter name, country
Enter string: Aleksandr Sergeevich Pushkin
Enter string: Russia
Author was inserted
```

1	1	Taras Shevchenko	Ukrain
2	2	John Ronald Reuel Tolkien	South Afrika
3	3	Sergey Sadv	Russia
4	4	Aleksandr Sergeevich Pushkin	Russia

Завдання №2

Команди створення індексів

BTree:

```
select_query = """CREATE INDEX ON authors USING BTREE(id);
```

GIN:

```
CREATE INDEX book_name ON books USING gin (to_tsvector('english', code));
```

Результати виконання команд:

```
"""SELECT * from readers WHERE id = 69420"""
```

```
Result [(69420, 'fd19311787af2cbd1e5e5cb2bcfc37ca', 11)]
```

Завдання №3

Команди що викликають тригер:

```
def create(self, title, year, author_id, subscription_id):
    try:
        session = Session()
        session.add(Book(title=title, year=year, author_id=author_id, subscription_id=subscription_id))
        session.commit()
        print("Book was inserted")

    except (Exception, Error) as error:
        print("Error occured in PostgreSQL: ", error)
```

```
def update(self, id, title, year, author_id, subscription_id):
    if (id < 1):
        print('Invalid id')
        return
    try:
        t = session.query(Book).get(id)
        t.title = title
        t.year = year
        t.author_id = author_id
        t.subscription_id = subscription_id
        session.add(t)
        session.commit()
        print("Book was updated")

    except (Exception, Error) as error:
        print("Error occured in PostgreSQL: ", error)
```

Текст тригера:

```

query = """DROP TABLE IF EXISTS book_logs;
CREATE TABLE book_logs(id integer NOT NULL, old_title text, new_title text, author_id integer);
CREATE OR REPLACE FUNCTION log_book() RETURNS trigger AS $BODY$
BEGIN
    IF NEW.title IS NULL THEN
        RAISE EXCEPTION 'Name cannot be null';
    END IF;
    IF NEW.author_id IS NULL THEN
        RAISE EXCEPTION 'Book cannot have null author_id';
    END IF;
    INSERT INTO book_logs VALUES(OLD.id, OLD.title, NEW.title, NEW.author_id);
    RETURN NEW;
END;
$BODY$ LANGUAGE plpgsql;
DROP TRIGGER IF EXISTS book_subj ON subjects;
CREATE TRIGGER book_subj AFTER UPDATE OR INSERT ON books
FOR EACH ROW EXECUTE PROCEDURE book_subj();"""

```

Зміни в таблиці після виконання команди:

```

Enter command (create, update, delete): create
Enter table name (authors, books, subscriptions, readers): books
Enter title, year, author_id, subscription_id
Enter string: New Book
Enter integer: 2021
Enter integer: 2
Enter integer: 1
Book was inserted

```

5	5	New Book	2021	2	1
---	---	----------	------	---	---

Висновки

В результаті виконання лабораторної роботи я здобув практичні навички використання засобів оптимізації СУБД PostgreSQL.