



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 та засоби управління”
тема “Засоби оптимізації роботи СУБД PostgreSQL”

Виконав(ла)
студент(ка) III курсу
групи КП-82

Кривчук Денис Віталійович
(прізвище, ім'я, по батькові)

варіант №11

Зарахована
“ ____ ” “ ____ ” 20__ р.
викладачем

(прізвище, ім'я, по батькові)

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

Завдання роботи полягає у наступному:

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.

1) Було змінено метод виконання запитів до БД на користь бібліотеки SQLAlchemy, що доцільна при використанні реляційних моделей.

Відповідні класи :

```
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy import Column, String, Integer, Date, ForeignKey
from datetime import datetime
Base = declarative_base()

class Album(Base):
    __tablename__ = 'Albums'

    id = Column(Integer, primary_key=True)
    date = Column(Date)
    name = Column(String)
    description = Column(String)
    owner = Column(String)

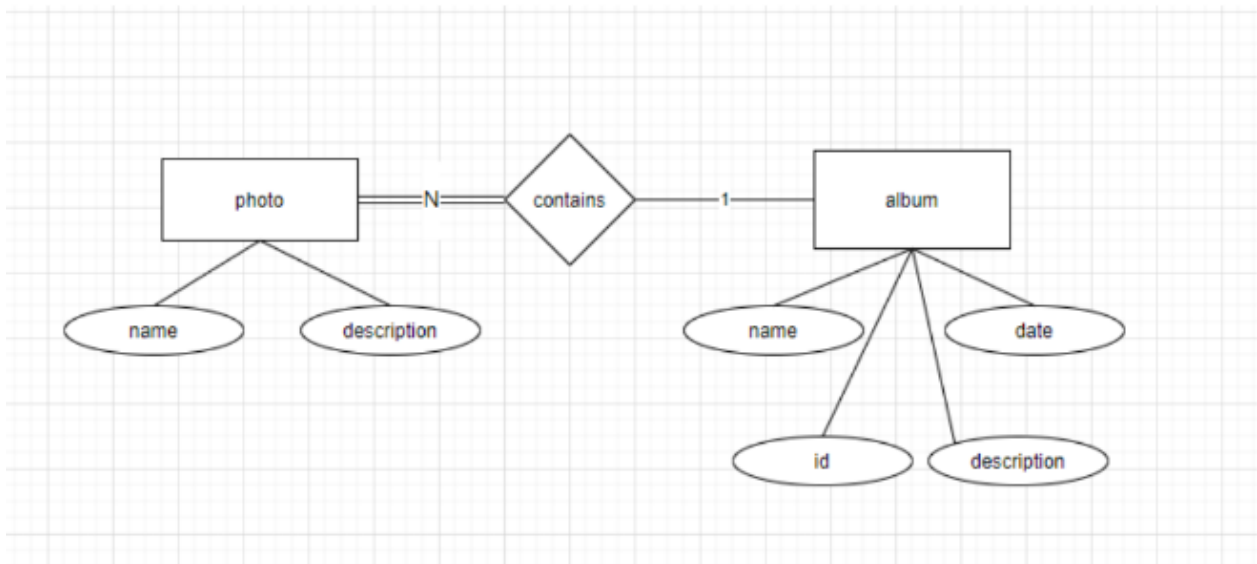
    def __init__(self, name, desc, owner):
        self.name = name
        self.description = desc
        self.owner = owner
        self.date = datetime.today()

class Photo(Base):
    __tablename__ = 'Photos'

    name = Column(String, primary_key=True)
    description = Column(String)
    albumid = Column(Integer, ForeignKey('Albums.id'))

    def __init__(self, name, desc, albId):
        self.name = name
        self.description = desc
        self.albumid = albId
```

таблиці

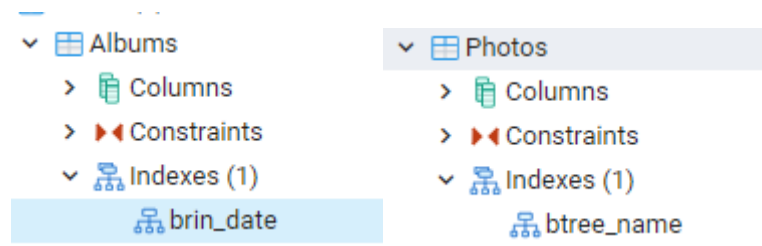


Запити

```

def updateAlbum(self, id, name, description, owner):
    self.session.query(Album).filter(Album.id == id).update({
        "name": name,
        "description": description,
        "owner": owner})
    self.session.commit()
  
```

2. Для бази даних було добавлено індексацію по певних параметрах, що пришвидшує пошук даних.



Результатом було пришвидшення виконання запитів в $\sim \frac{n}{\log n}$ раз

Виконання операцій на 1000 рядків даних

```

фільтрація Photo.name == smthreallynew
Duration: 0:00:00.104202
smthreallynew
агрегатні функції, count
Duration: 0:00:00.004997
1
groups by album
Duration: 0:00:00.003017
[(31,), (34,), (40,), (48,), (35,), (36,), (25,), (30,), (26,), (31,), (35,), (35,), (35,),
albums by time
Duration: 0:00:00.018988
newname
sometestalbum
NDIjZJVuL7
PIbXmagwVW
N7lb7SUOWn
bsdvin
ensbEwCLdF
Y0mPNwCQXE
xJynYEZcdp
5kvJD3FuFK
tuUMIxn86W

```

3) Були створені тригери для операцій видалення та перед оновленням даних, що добавляють логи про дані події.

Відповідні функції

```

1 BEGIN
2     IF OLD.date <= CURRENT_DATE THEN
3         INSERT INTO albums_log(albumid,operation,date,name)
4         VALUES(OLD.id,'delete', CURRENT_DATE, OLD.name);
5     END IF;
6
7     RETURN NEW;
8 END;
9

```

```
1 DECLARE
2     it RECORD;
3     updated varchar(50) := 'updated';
4     comp varchar(50);
5 BEGIN
6     IF OLD.date <= CURRENT_DATE THEN
7         FOR it IN
8             SELECT * FROM albums_log
9         LOOP
10             comp :=BTRIM(it.name, chr(10)) ;
11             IF NEW.name = comp THEN
12                 updated := 'updated with used name';
13                 EXIT;
14             END IF;
15         END LOOP;
16         INSERT INTO albums_log(albumid,operation,date,name)
17         VALUES(NEW.id,updated, CURRENT_DATE,NEW.name);
18     END IF;
19
20     RETURN NEW;
21 END;
```