

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 № 2

з дисципліни “Бази даних”

тема “Створення додатку бази даних, орієнтованого на взаємодію з СУБД
PostgreSQL”

Виконав(ла)
студент(ка) II курсу
групи КП-01

Перевірів
“____” “_____” 20__ р.
асистент

Грабовська Анастасія Дмитрівна
(прізвище, ім'я, по батькові)

Радченко Костянтин Олександрович
(прізвище, ім'я, по батькові)

Київ 2021

Мета роботи

Метою роботи є здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.

Завдання

Завдання роботи полягає у наступному:

1. Реалізувати функції внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі No1, засобами консольного інтерфейсу.
2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.
3. Забезпечити реалізацію пошуку за декількома атрибутами з двох та більше сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як шаблон функції LIKE оператора SELECT SQL, для логічного типу – значення True/False, для дат – у рамках діапазону дат.
4. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

Хід роботи:

Вимоги до пункту №1 деталізованого завдання:





- Enter table name: error
Invalid values
- Enter a number: a
a isn't a number

Вимоги до пункту №2 деталізованого завдання:






- Curators:

	id [PK] integer	name text	surname text	phone text
1843	2831	Klp	lwb	271768
1844	2832	Onf	Trs	468619
1845	2833	Vdb	Gxk	749955
1846	2834	Qan	Bkq	881726
1847	2835	Nun	Uhp	613042
1848	2837	Uyb	Mvv	471037
1849	2838	Rub	Clo	981016
1850	2840	Scm	Ted	440146
1851	2842	Eyt	Mfg	838315
1852	2843	Rsm	lbn	550245
1853	2844	Ajb	Coq	909059
1854	2845	Pbl	Kjl	333296

- Groups:

	 id [PK] integer 	code text 	curator_id integer 
8	8	KB-92	5
9	9	PP-50	10
10	10	BD-39	6
11	11	JI-22	5
12	12	TT-37	13
13	13	VD-51	12
14	14	SR-34	4
15	15	PL-44	4
16	16	YV-36	6
17	17	XY-84	13
18	18	AK-65	3
19	19	PD-42	27

- Students:

	 id [PK] integer 	surname text 	name text 	group_id integer 
16	24	Phn	Agn	23
17	27	Hnq	lyp	28
18	29	Gvv	Jrj	5
19	34	Bqf	Rtl	5
20	41	Nsj	Qsj	14
21	43	Iey	Ygf	23
22	47	Myy	Eyn	21
23	48	Bao	Xcr	13
24	49	Ysx	Giv	30

- Subjects:

	id [PK] integer	name text	curator_id integer
16	21	VL	13
17	23	UV	9
18	25	FY	13
19	27	HB	7
20	29	SX	5
21	32	GX	1
22	34	RP	38
23	35	XD	6
24	37	AQ	5

Вимоги до пункту №3 деталізованого завдання:

- Command: readStudentCurator

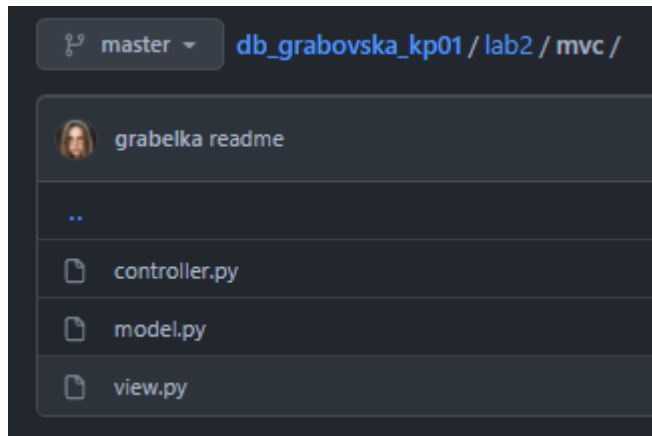
```
Enter command (create, delete, update, random, readCuratorGroup, readCuratorSubject, readStudentCurator or exit): readStudentCurator
Enter a number: 7
Enter a value: V
Result [('Ruslan', 'Hadyak', 'KP-02', 'Serhiy', 'Vyshnya'), ('Ruslan', 'Hadyak', 'KP-02', 'Cws', 'Vuo'), ('Ruslan', 'Hadyak', 'KP-02', 'Lrm', 'Vxi')]
Time for operation 0.011946400000056201
```

- Command: readCuratorSubject

```
Enter command (create, delete, update, random, readCuratorGroup, readCuratorSubject, readStudentCurator or exit): readCuratorGroup
Enter a number: 1
Enter a value: K
Result [('Vioria', 'Suschuk', 'KP-01')]
Time for operation -0.00522200000003391
```

Вимоги до пункту №4 деталізованого завдання:

- MVC struct:



- View:

```
1 class view:
2
3     def __init__(self):
4         print('-----Welcome-----')
5
6     def readCommand(self):
7         com = input("Enter command (create, delete, update, random, readCuratorGroup, readCuratorSubject, readStudentCurator or exit): ")
8         return com
9
10    def readTable(self):
11        tab = input("Enter table name: ")
12        if(tab == 'curators' or tab == 'groups' or tab == 'students' or tab == 'subjects'):
13            return tab
14        return ''
15
16    def getStr(self):
17        val = input("Enter a value: ")
18        return val
19
20    def getNum(self):
21        val = input("Enter a number: ")
22        try:
23            int(val)
24        except ValueError:
25            print(val + " is invalid")
26            return 0
27        return int(val)
```

- Model:

```
import timeit
import psycopg2
from psycopg2 import Error

class curators:

    def __init__(self):
        self.id = 0
        self.name = ""
        self.surname = ""
        self.phone = ""

    def readGroup(self, c, code):
        start = timeit.timeit()
        if (c < 1):
            print('Error with input!')
            return
        try:
            connection = psycopg2.connect(user="postgres",
                                           password="1",
                                           host="127.0.0.1",
                                           port="5432",
                                           database="university")

            cursor = connection.cursor()
            selectr_query = """SELECT curators.name, curators.surname, groups.code
            item_tuple = (c, c, code + "%")
            cursor.execute(selectr_query, item_tuple)
            connection.commit()
            print("Result", cursor.fetchall())
        except (Exception, Error) as error:
            print("Error with PostgreSQL", error)
        finally:
            if connection:
                cursor.close()
                connection.close()
            end = timeit.timeit()
            print("Time for operation " + str(end - start))
```

- Controller:

```
from model import curators
from model import groups
from model import students
from model import subjects
from view import view

v = view()
c = curators()
g = groups()
st = students()
s = subjects()

cont = True
while(cont):
    com = v.readCommand()
    if (com == 'exit'):
        cont = False
    elif (com == 'readCuratorGroup'):
        id = v.getNum()
        code = v.getStr()
        c.readGroup(id, code)
    elif (com == 'readCuratorSubject'):
        id = v.getNum()
        surname = v.getStr()
        c.readSubject(id, surname)
    elif (com == 'readStudentCurator'):
        id = v.getNum()
        surname = v.getStr()
        c.readStudent(id, surname)
    elif (com == 'create' or com == 'delete' or com == 'update' or com == 'random'):
        tab = v.readTable()
```

Висновки

Під час виконання роботи я здобула вміння та практичні навички створення та програмування прикладних додатків баз даних PostgreSQL.