



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
студент II курсу
групи КП-01
Мазніченко Сергій Сергійович

Перевір
ив “_____” “_____”
” 2021_____р
виклад
ач Радченко Костянтин
Олександрович

Мета

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

Постановка задачі

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).

Копії екрану, що підтверджують вимоги 1-3 завдання

Вимоги до пункту №1

Обробка виняткових ситуацій (помилки) при введенні/вилученні даних:

```
Enter command (create, read, update, delete, generate, search): read
Enter table name (authors, books, abonements, readers): blablabla
Invalid table name
```

Валідація даних при введенні користувачем:

```
Enter command (create, read, update, delete, generate, search): read
Enter table name (authors, books, abonements, readers): books
Enter id
Enter integer: hello world
hello world is not integer
Invalid id
```

Вимоги до пункту №2

Згенеровані дані таблиць





Books:

	<div><div><div><div></div><div></div></div><div><div>id</div><div>[PK] integer</div></div></div><div></div></div>	<div><div><div><div></div><div></div></div><div><div>title</div><div>character varying (90)</div></div></div><div></div></div>
--	---	--

Abonements:

	<div><div><div><div></div><div></div></div><div><div>id</div><div>[PK] integer</div></div></div><div></div></div>	<div><div><div><div></div><div></div></div><div><div>price</div><div>real</div></div></div><div></div></div>
--	---	--

Readers:

 id [PK] integer	 username character varying (60)	 read_count integer	 abonnement_id integer
310014	66dfc942e44c0240da82c11f77054de5	3	2132
310015	50837e3ba8d73c1fd71471467152c235	7	77857
310016	7dcfdac78189982649bd04c14e7ca4ab	1	10952
310017	9b894382155bee2ce060f521e5bf967c	0	84852
310018	eda80ca41bd3c886b1002edb21349a62	8	16278
310019	b48b5ad74b1c926de21f32a3bd55e03a	5	38170
310020	c730b1c65d8efae82d1361cf6b4d368b	2	86079
310021	d31742a13860cd57e86cde952d4273e2	0	12167
310022	b28d7a66e05f338983c0757c5f7af649	8	23360
310023	9dd56fc0a08c700d35971c92b1f78c72	8	8580
310024	5aad4ddc67905a2f075ba28765609cb2	1	18326
310025	5c74340e7143d2d69e7d01286470f9d6	5	46339
310026	1244605e3f2927edd817bb34c5a8e7e8	6	40657

Authors:

 id [PK] integer	 name character varying (90)	 country character varying (60)
33	e2dfc567e90ced402f63556a0d36cb96	4bfb6293fc2307ca8259481e85fd8472
34	1c59a3eafa6c8a6ea59637ef4b778129	848590f75db865f3c2f566aabd470308
35	ad9eb28c58cc0a14b3eab62d83fccebd	53d263e3c17a2c2f56578bf36e722ef9
36	8ea32be75987512ffde383f1e0742cd4	bb958b13eea58e495023fc03a923eb27
37	ea270468a2241745d92c13aa9440ba05	34e116d253e2c377f56e8abdabe4a125
38	fb8ea58d6b3c1fa73efdc20e4d5861b4	d48a269e8e662344dceda39e48a6ce2c
39	2ac59d04245aae484b2b78e6899ff5b5	271357aa5ed4d806583955c7416ec4b7
40	5d0a1c53d8a531edf25d98f227ec86b4	21b32640197afe14e933b26c98508597
41	4b62e94cf460b3e7434aec7cb00416be	58f7bab87012926ad1aebc526b4de9d7
42	e266493741d7ee5862534206d1e539a1	c8fb410ab563b96873330ac8320fce8f
43	8bffd8f044d1ca6e87c595fc4d546c0c	f5ded8e73fc5764712d10c1871d73fc8
44	1e1da0ae117ac1dd19787bc15d083c2c	b2653c7a076803bcc20e40c8afa92e30
45	11d85ab6e85bfc55e0d832f403cbb559	84e278a267fe3a77cdf394ca7d4f989f

Вимоги до пункту №3

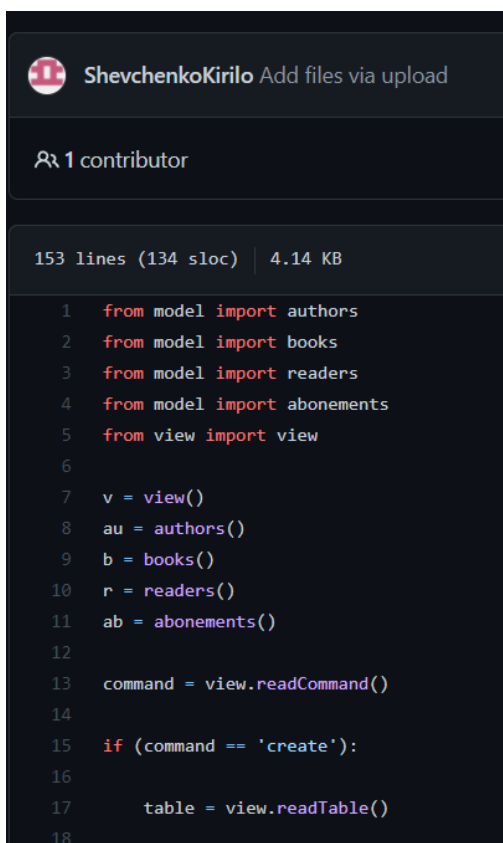
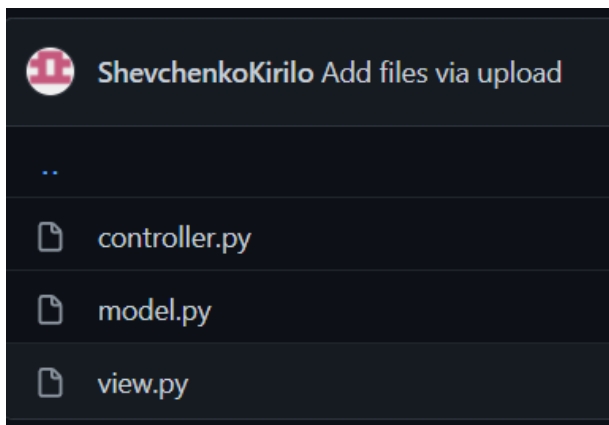
Уведення пошукового запиту та результати виконання запиту:


```
Enter command (create, read, update, delete, generate, search): search
Enter year, name
Enter integer: 1850
Enter string: Mark Twain
Result [('Mark Twain', 1915, 'e7026e7b8a7c69ecc8f009de137ea446'), ('Mark Twain', 1885, 'The Adventures of Huckleberry Finn'),
Search operation lasted 91.83550000000018 ms
```

```
n'), ('Mark Twain', 1876, 'The Adventures of Tom Sawyer'), ('Mark Twain', 1874, 'ab2d5f7b2e1350fac748600e68250bba')]
```

Вимоги до пункту №4

Код, завантажений на GitHub:




 ShevchenkoKirilo Add files via upload

1 contributor

587 lines (502 sloc) | 21.6 KB

```
1 import psycopg2
2 from timeit import default_timer as timer
3 from psycopg2 import Error
4
5
6 class authors:
7
8     def __init__(self):
9         self.id = 0
10        self.name = ""
11        self.country = ""
12
13    def create(self, name, country):
14        try:
15            connection = psycopg2.connect(user="postgres",
16                                           password="1",
17                                           host="127.0.0.1",
18                                           port="5432",
19                                           database="library_db")
20
21            cursor = connection.cursor()
22            insert_query = """ INSERT INTO authors (name, country) VALUES (%s, %s)"""
23            item_tuple = (name, country)
24            cursor.execute(insert_query, item_tuple)
25            connection.commit()
26            print("Author inserted")
```

 ShevchenkoKirilo Add files via upload

1 contributor

27 lines (22 sloc) | 746 Bytes

```
1 class view:
2
3     def __init__(self):
4         print('Program started')
5
6     def readCommand():
7         com = input("Enter command (create, read, update, delete, generate, search): ")
8         return com
9
10    def readTable():
11        tab = input("Enter table name (authors, books, abonements, readers): ")
12        if (tab == 'authors' or tab == 'books' or tab == 'abonements' or tab == 'readers'):
13            return tab
14        return ''
15
16    def getVal():
17        val = input("Enter string: ")
18        return val
```

Висновки

В результаті виконання лабораторної роботи я навчився програмуванню прикладних додатків баз даних PostgreSQL.