



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
Тітов Єгор Павлович

Перевірив
“ ____ ” “ _____ ” 20__ р
асистент

Радченко Костянтин
Олександрович

Київ 2021

Мета: здобути вміння програмування прикладних додатків баз даних PostgreSQL.

Завдання

1. Реалізувати функції внесення, редагування та видалення 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).

Відповіді на вимоги до звітування







- Вимоги до пункту №1 деталізованого завдання:

Обробка виняткових ситуацій:	Enter command (create, delete, update, generate, search): create Enter table name: unExistedName Table with that name does not exist _
Валідація даних:	Enter table name: readers Enter integer value: f f is not integer _











- Вимоги до пункту №2 деталізованого завдання:

Копії екрану з фрагментами згенерованих даних таблиць









- authors:

	 id	 Integer	 fullname	 character varying (50)	 country	 character varying (50)
2231	103230		2f8f63981b4d5133a84bf4bedd0b1cd3	e8ec4370a259bbdf9d381f22b20b0ec8		
2232	103231		066c5ccbb522059d97cdc1abc7804f4	7d3f3ebd39e67643dd0c2af4ec83fcc9		
2233	103232		451290d3a994b945af4c68ed37a872d3	93a5543fc8ab0f7839d76bd079c07603		
2234	103233		2ad0974f6b0c2218b4750ac851f9e48a	2555d4a65208e198776b4116dbb6c8e1		
2235	103234		94178586bd05f7cf87447e4ad07979c5	6b11f7021af758101cd9898fd6a03d95		
2236	103235		e2c4612f6ea05836f71b59dce8dda28c	4efde9cde7c33ae403a8b0cf21f3cf76		
2237	103236		d09a695c37d98a9095f8837d72b54011	8829e58bd8c47361dbec5ff6de70d8f9		
2238	103237		73491a5b7a2d6302a3801c757143e056	efce901c7cd3a1520707372d5641fb25		
2239	103238		acd876eb6f13adaa3336834251f91b3c	991431b9b9a9b8ed90e427780851cd91		
2240	103239		b0b547279fcbc5b97cb703368b8cd7be	bc3b27f177a86a70dee48043f93ff042		
2241	103240		f242b71112ad002332073f8212038355	4169f64d08cf8313b93c855adada185f		
2242	103241		dd9e1251f69d84d1867311729a491797	b3b802debbde538a269d6728791003af		
2243	103242		1867f08fc4aef0b2c1efb4f6728da340	5afc3853227110d5dd8b90871338a70d		
2244	103243		d5e41aa60d06082df8e7d088014fc231	5b490dd34b99b90b97a15d4cd9e1f35b		
2245	103244		2a68837ea236116ffe2a42ba26d41929	44493fd66ae20b1671d68415c8fb966		
2246	103245		ed4ef720e255c785a6057015bc249322	b29b192ef022e9196d2c638d09b70953		
2247	103246		417a4c5385154ad497d6a1227a090c13	be7c3ef390621e5680dcbdfdae9201a		
2248	103247		6c06db7321aed03b0a245d9b7a7cd464	d161c7ab96367b79f7b07c57aad68cba		
2249	103248		1abae17177fdec6596f93aa350d39a33	dd84f764a41a1fc74ed348153ba8251b		
2250	103249		e7c3b09ba51e587aee7345f5a5c016a3	70c93d3563e3ec3e3aa8b67a9eefaf3a		

- books:

	 id	 Integer	 title	 character varying (50)	 rating	 real	 author_id	 integer	 abonement_id	 integer
1979	72001		6c8ea7161cda5d2eb626265e65b1e030		4	156608		6820		
1980	72002		cc48626a4a44784d203e36f5a1b0ca1d		9	102174		51135		
1981	72003		63209c8ab7e091d65e25a470b08b607e		6	153922		85201		
1982	72004		5a667a26a4c93b450d9a8b7078b93aa6		8	114588		1354		
1983	72005		1d50da5af01604a1287293ace8caa710		1	164151		27159		
1984	72006		830031881e0866088d58b30ac73401fe		2	121207		86035		
1985	72007		0ef2ef7ebacf9a754b682e1c21de1f05		4	161160		74692		
1986	72008		bfbdfc0ef89082d5bbf50eeaaecc62a5		4	198821		4609		
1987	72009		9c9dfe50713248271997d14dbab9dd9c		7	111976		82230		
1988	72010		a06eec72a908d6183c7bcc4d184e3f44		0	179582		76185		
1989	72011		74a3988dc3a2188b31a55a69e89cf6b7		0	114578		13958		
1990	72012		1cf6fc8ea3715fbfb18eb42feb199dff		6	138010		3206		
1991	72013		81cb6473f589a1a6163a36746f21559d		4	114917		12226		
1992	72014		4b90fa2dd3b8896bb8dd3c1e69dc4aab		7	120071		11037		
1993	72015		97be001522cac9a2e1f3095e1283fb1f		6	135429		49854		
1994	72016		3ea96793751cc96f8b532790d7dfbb96		6	183549		80065		
1995	72017		09856f3760e153d5037af85ee6844e7		5	150098		37841		
1996	72018		2ba03de22f755be66feb417cce3509b		2	142035		59926		

- readers:

	 id	 Integer	 username	 text	 number_of_read	 integer	 abonement_id	 integer
3415	3417		671d0b7fbde22984735a5488a4142a4f			1		88440
3416	3418		2733370e9a2f83cb370d34cb87648872			56		17653
3417	3419		88e3519f623524a52044cca927f5592a			3		33975
3418	3420		04ef89adc1035b82008e00615f38b434			64		77688
3419	3421		5037004fc9a71ba164441d22327828ae			74		50906
3420	3422		b0942ff373d2333876ff3cbfb02564da			81		11021
3421	3423		4966e39f29c385c30a699658f97d5999			47		33778
3422	3424		fb411581d0397ce265c4e98cc903d7a2			87		77384
3423	3425		636008ec898a78c383e3a47348a5cc50			32		68785
3424	3426		aa2a9677035b2ace8dd6bdbb0df86e3e			8		62315
3425	3427		69ae2d8d6948f94a36d1b787cbd5ad4b			30		98870
3426	3428		d0c76149c6d312ea12f6daedaae70221			47		86739
3427	3429		cc25b84d10a6b0d3f2206e8b3df75337			33		52387

- abonements:

	id [PK] integer	price real	expiring_time integer
1293	1293	321	83
1294	1294	950	140
1295	1295	138	88
1296	1296	936	3
1297	1297	405	174
1298	1298	623	222
1299	1299	489	103
1300	1300	815	201
1301	1301	79	274
1302	1302	999	110
1303	1303	309	190
1304	1304	886	27
1305	1305	377	207
1306	1306	455	79
1307	1307	921	261
1308	1308	105	279

- *Вимоги до пункту №3 деталізованого завдання:*

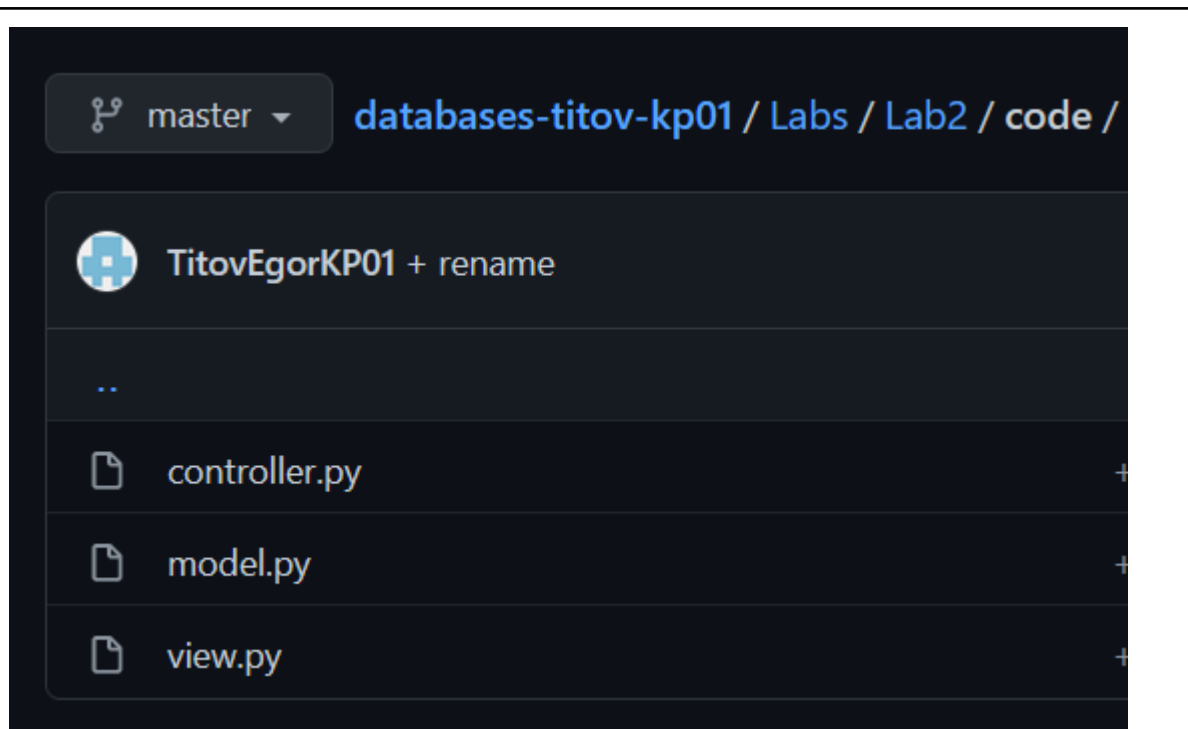
Ілюстрація введення пошукового запиту та результатів виконання запитів:

```
Enter command (create, read, update, delete, generate, search): search
Enter integer value: 1
Enter value: c
Enter value: author
Result [('author', 'bd3e1d54b872a90c60f59cf0615a4335', 8.0), ('some author', '8cbc9b66e7c48a8245241aae5b2d9d89', 2.0)]
Search operation take 92.18350000000086 ms
```

- *Вимоги до пункту №4 деталізованого завдання:*

Ілюстрації програмного коду з репозиторію Git:

- Організація коду згідно шаблону MVC:



- Controller:

141 lines (120 sloc) | 3.57 KB

```
1  from model import authors
2  from model import books
3  from model import readers
4  from model import abonements
5  from view import view
6
7  v = view()
8  au = authors()
9  b = books()
10 r = readers()
11 ab = abonements()
12
13 command = view.readCommand()
14
15 if (command == 'create'):
16
17     table = view.readTable()
18
19     if (table == 'authors'):
20         id = view.getInt()
21         fullname = view.getVal()
22         country = view.getVal()
23         au.create(id, fullname, country)
24     elif (table == 'books'):
25         id = view.getInt()
26         title = view.getVal()
27         rating = view.getInt()
28         author_id = view.getInt()
29         abonement_id = view.getInt()
30         b.create(id, title, rating, author_id, abonement_id)
31     elif (table == 'readers'):
32         id = view.getInt()
```

- Model:

611 lines (519 sloc) | 22.3 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.fullname = ""
11         self.country = ""
12
13
14     def create(self, id, fullname, country):
15
16         if (id < 1):
17             print('Invalid id entered')
18             return
19         try:
20             connection = psycopg2.connect(user="postgres",
21                                           password="1337",
22                                           host="127.0.0.1",
23                                           port="5432",
24                                           database="library_db")
25
26             cursor = connection.cursor()
27             insert_query = """ INSERT INTO authors (id, fullname, country) VALUES (%s, %s, %s)"""
28             item_tuple = (id, fullname, country)
29             cursor.execute(insert_query, item_tuple)
30             connection.commit()
31             print("Entity inserted")
32
33         except (Exception, Error) as error:
34             print("Error occured in PostgreSQL: ", error)
35     finally:
```

- View:

29 lines (24 sloc) | 787 Bytes

```
1  class view:
2
3      def __init__(self):
4          print('=====')
5          print('== Starting program ==')
6          print('=====')
7
8      def readCommand():
9          com = input("Enter command (create, read, update, delete, generate, search): ")
10         return com
11
12     def readTable():
13         tab = input("Enter table name: ")
14         if (tab == 'authors' or tab == 'books' or tab == 'abonements' or tab == 'readers'):
15             return tab
16         return ''
17
18     def getVal():
19         val = input("Enter value: ")
20         return val
21
22     def getInt():
23         val = input("Enter integer value: ")
24         try:
25             int(val)
26         except ValueError:
27             print(val + " is not integer")
28             return 0
29         return int(val)
```

Висновок

В результаті виконання даної лабораторної роботи я здобув вміння програмування прикладних додатків баз даних PostgreSQL.