

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

ЗВІТ

з лабораторної роботи № 2

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

Виконав:

студент 3-го курсу, групи КП-82,
спеціальності 121 – Інженерія
програмного забезпечення
Мельничук Олексій Геннадійович

Перевірив:

к. т. н, старший викладач
Радченко Костянтин Олександрович

Київ – 2020

Мета роботи

Здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.

Завдання





Варіант 12







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-контроллер).

Хід роботи

За предметну галузь для проектування бази даних була обрана залізнична станція. Залізниця складається з таких сутностей як потяги, вагони, станції та рейси.

Data Output		Explain	Messages	Notifications			
	id [PK] bigint	trainid bigint	departureid bigint	arrivalid bigint	departuretime time without time zone	arrivaltime time without time zone	laststation bigint
1	1	4	1	5	14:50:00	16:00:00	6
2	2	1	3	4	12:40:00	20:30:00	5
3	3	4	6	7	15:00:00	15:50:00	4
4	4	5	3	1	03:20:00	04:15:00	1
5	5	3	8	2	21:30:00	22:00:00	2
6	6	3	2	3	06:25:00	12:20:00	8
7	7	7	5	2	13:15:00	16:55:00	3
8	8	30	35	52	00:43:56.994206	03:11:26.025277	54
9	9	88	42	21	01:13:55.004582	05:14:30.878362	63
10	10	97	86	53	21:07:24.288931	20:21:18.032466	71
11	11	88	50	28	05:08:43.105417	23:30:18.285752	60
12	12	27	50	12	20:49:08.247507	01:32:50.171943	48
13	13	53	4	37	03:00:11.183143	04:32:15.203773	56
14	14	33	73	54	00:05:29.270695	04:42:35.261185	5
15	15	4	4	62	00:24:46.412447	04:17:18.301628	51
16	16	14	14	96	22:05:54.851282	22:13:33.874549	81
17	17	29	26	17	22:46:54.304972	04:37:03.750786	36

Data Output	Explain	Messages	Notifications
	 id [PK] bigint 	name character varying (30) 	availability boolean 
1	1	Cape Earl	true
2	2	Bjorn Valley	true
3	3	New Vegas	true
4	4	Ghent Overlook	true
5	5	Houston	true
6	6	Edmonton	true
7	7	Port Nowhere	true
8	8	Bainbridge	true
9	9	82rufi	true
10	11	7-11 Breaks	false
11	12	Cowl Edge	true
12	13	264eb859e3	true
13	14	43f3f3	true
14	15	7c06d504c70	true
15	16	8599514a	true
16	17	aedff06b3b	false
17	18	09e3f93358d6	true
18	19	ff7dc2	true

Data Output		Explain	Messages	Notifications	
	 id [PK] bigint 	number integer 	type character varying (15) 	subtype character varying (15) 	trainid bigint 
1	2	45	Passenger	Econom	2
2	3	13	Passenger	Luxury	3
3	4	29	c91916e3b92f		3
4	5	4	Freight	Boxcar	5
5	6	9	Passenger	Econom	4
6	7	3	Freight	Boxcar	5
7	8	45	Freight	Hopper	3
8	9	7	Freight	Hopper	5
9	10	13	Freight	Tank	1
10	11	20	Passenger	Luxury	5
11	12	3	Passenger	Econom	2
12	13	44		8a2d75d3762c3	3
13	14	66	964adb0	2f5b09ca	3
14	15	54	491	cc4ab70	4
15	16	86	19e541e39057b	6de2a0d582	4
16	17	21	49380d879547	fb4a556	4
17	18	341	Big	Very Big	5

	 id [PK] bigint 	name character varying (30) 	type character varying 
1	1	Super Chef F	Freight
2	2	Sigma Atlantic	Passenger
3	3	Golden Arrow	Passenger
4	4	Amtrak A532	Passenger
5	5	Big Crimson	Freight

Рис. 0.1-4 Фото таблиц баз даних

Завдання 1

```
[4] Random values generation
table_routes
table_stations
table_trains
table_wagons
Choose correctly table index: 0
Chose table_routes
Those are the columns:
[bigint] id | [bigint] trainid | [bigint] departureid | [bigint] arrivalid | [time without time zone] departu
retime | [time without time zone] arrivaltime | [bigint] laststation |
Enter correctly amount of new rows: 10
Oops! An exception has occurred: ОШИБКА: повторяющееся значение ключа нарушает ограничение уникальности "tabl
e_routes_pkey"
DETAIL:  Ключ "(id)=(5)" уже существует.

Try again later.
```

Рис. 1.1 Обробка помики

```
[1] Adding a new row
table_routes
table_stations
table_trains
table_wagons
Choose correctly table index: 0
Chose table_routes
Those are the columns:
[bigint] id | [bigint] trainid | [bigint] departureid | [bigint] arrivalid | [time without time zone] departuretime | [time without time zone] arriva
ltime | [bigint] laststation |
Enter value of [trainid]: -4234
Enter correct value of [trainid]: train
Enter correct value of [trainid]: 5
Enter value of [departureid]: 3
Enter value of [arrivalid]: 7
Enter value of [departuretime]: not time
Enter correct value of [departuretime]: 25:00:00
Enter correct value of [departuretime]: 23:61:61
Enter correct value of [departuretime]: 23:00:00
Enter value of [arrivaltime]: 14:23:12
```

```

[2] Editing existing row
table_routes
table_stations
table_trains
table_wagons
Choose correctly table index: 1
Chose table_stations
Those are the columns:
[bigint] id | [character varying] name | [boolean] availability |
[1] | Cape Earl | True
[2] | Bjorn Valley | True
[3] | New Vegas | True
[4] | Ghent Overlook | True
[5] | Houston | True
[6] | Edmonton | True
[7] | Port Nowhere | True
[8] | Bainbridge | True
[9] | 82rufi | True
[11] | 7-11 Breaks | False
[12] | f7e77a4a1 | False
[13] | 264eb859e3 | True
[14] | 43f3f3 | True
Enter id of needed row or -1 to quit: 12
Leave input empty if you don't want to change anything.
Enter value of [name]: Cowl Edge
Enter value of [availability]: wfwfw
Enter correct value of [availability]: wfwfw
Enter correct value of [availability]: true
[('id', 12), ('name', "Cowl Edge"), ('availability', 'true')]
Success!. Press any key.

```

Рис. 1.2-3 Валідація даних при вводі

Завдання 2

	id [PK] bigint	trainid bigint	departureid bigint	arrivalid bigint	departuretime time without time zone	arrivaltime time without time zone	laststation bigint
2	2	1	3	4	12:40:00	20:30:00	5
3	3	4	6	7	15:00:00	15:50:00	4
4	4	5	3	1	03:20:00	04:15:00	1
5	5	3	8	2	21:30:00	22:00:00	2
6	6	3	2	3	06:25:00	12:20:00	8
7	7	7	5	2	13:15:00	16:55:00	3
8	8	30	35	52	00:43:56.994206	03:11:26.025277	54
9	9	88	42	21	01:13:55.004582	05:14:30.878362	63
10	10	97	86	53	21:07:24.288931	20:21:18.032466	71
11	11	88	50	28	05:08:43.105417	23:30:18.285752	60
12	12	27	50	12	20:49:08.247507	01:32:50.171943	48
13	13	53	4	37	03:00:11.183143	04:32:15.203773	56
14	14	33	73	54	00:05:29.270695	04:42:35.261185	5
15	15	4	4	62	00:24:46.412447	04:17:18.301628	51
16	16	14	14	96	22:05:54.851282	22:13:33.874549	81
17	17	29	26	17	22:46:54.304972	04:37:03.750786	36

	id [PK] bigint	name character varying (30)	availability boolean
22	23	7817	true
23	24	96995ca0ed75	false
24	25	66237ba7	true
25	26	ae5dc	false
26	27	544045a18	false
27	28	3dcd859d8ae1c	false
28	29	06f	true
29	30	001	false
30	31	3ef667e3	true
31	32	0056cbd9c1	true
32	33		false
33	34		true
34	35	e348	false
35	36	e37fda7f6	true
36	37	dd3b0b556	true
37	38	45f79bac2	true
38	39	a6d9ce1a59b	false

Рис. 2.1-2 Згенеровані дані

Завдання 3

```

[5] Formulating search request
table_routes
table_stations
table_trains
table_wagons
Choose correctly table index: 2
Chose table_trains
Those are the columns:
[bigint] id | [character varying] name | [character varying] type |
Proceed with formulating search request?
Print y or n: y
Print correct name of any column: coloomn
Print correct name of any column: name
Enter non-empty string: Crimson
Proceed with formulating search request?
Print y or n: y
Print correct name of any column: type
Enter non-empty string: Freight
Proceed with formulating search request?
Print y or n: n
SELECT * FROM table_trains WHERE name ILIKE '%Crimson%' AND type ILIKE '%Freight%'
0.0029273033142089844
[5] | Big Crimson | Freight
Press any button

```

```

[5] Formulating search request
table_routes
table_stations
table_trains
table_wagons
Choose correctly table index: 3
Chose table_wagons
Those are the columns:
[bigint] id | [integer] number | [character varying] type | [character varying] subtype | [bigint] trainid |
Proceed with formulating search request?
Print y or n: y
Print correct name of any column: number
Enter correctly the first integer: 1
Enter correctly the second integer: 50
Proceed with formulating search request?
Print y or n: y
Print correct name of any column: tpye
Print correct name of any column: type
Enter non-empty string: Passenger
Proceed with formulating search request?
Print y or n: n
SELECT * FROM table_wagons WHERE number BETWEEN 1 AND 50 AND type ILIKE '%Passenger%'
0.006258964538574219
[2] | 45 | Passenger | Econom | 2
[3] | 13 | Passenger | Luxury | 3
[6] | 9 | Passenger | Econom | 4
[11] | 20 | Passenger | Luxury | 5
[12] | 3 | Passenger | Econom | 2
Press any button

```



```

[5] Formulating search request
table_routes
table_stations
table_trains
table_wagons
Choose correctly table index: 0
Chose table routes
Those are the columns:
[bigint] id | [bigint] trainid | [bigint] departureid | [bigint] arrivalid | [time without time zone] departuretime | [time with
ut time zone] arrivaltime | [bigint] laststation |
Proceed with formulating search request?
Print y or n: y
Print correct name of any column: departuretime
Enter time1 in format HH:MM:SS : 00:00:00
Enter time2 in format HH:MM:SS : 12:00:00
Proceed with formulating search request?
Print y or n: n
SELECT * FROM table_routes WHERE departuretime BETWEEN '00:00:00'::time AND '12:00:00'::time
0.0012505054473876953
[4] | 5 | 3 | 1 | 03:20:00 | 04:15:00 | 1
[6] | 3 | 2 | 3 | 06:25:00 | 12:20:00 | 8
[8] | 30 | 35 | 52 | 00:43:56.994206 | 03:11:26.025277 | 54
[9] | 88 | 42 | 21 | 01:13:55.004582 | 05:14:30.878362 | 63
[11] | 88 | 50 | 28 | 05:08:43.105417 | 23:30:18.285752 | 60
[13] | 53 | 4 | 37 | 03:00:11.183143 | 04:32:15.203773 | 56
[14] | 33 | 73 | 54 | 00:05:29.270695 | 04:42:35.261185 | 5
[15] | 4 | 4 | 62 | 00:24:46.412447 | 04:17:18.301628 | 51
Press any button

```

Рис. 3.1-3 Пошук та виконання пошуку

Завдання 4

The screenshot shows a GitHub repository page for 'databases_kp' at the path 'code'. The page displays a commit history for the 'code' directory, listing files like '_pycache_', 'menu_ops.py', and 'work.py' uploaded 28 minutes ago. The repository is owned by 'auvy' and has 1 star and 0 forks.

Repository: [auvy / databases_kp](#)

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master databases_kp / lab2 / code /

Go to file Add file

auvy uploaded code of lab2 f83cc2c 28 minutes ago History

File	Uploaded
...	...
__pycache__	uploaded code of lab2 28 minutes ago
menu_ops.py	uploaded code of lab2 28 minutes ago
work.py	uploaded code of lab2 28 minutes ago

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

pgAdmin 4 x databases_kp/menu_ops.py at m x +

github.com/auvy/databases_kp/blob/master/lab2/code/menu_ops.py

Search or jump to... Pull requests Issues Marketplace Explore

auvy / databases_kp

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master databases_kp / lab2 / code / menu_ops.py / <> Jump to

Go to file

auvy uploaded code of lab2 Latest commit f83cc2e 30 minutes ago History

1 contributor

423 lines (354 sloc) 12.6 KB

Raw Blame

```
1 import time
2 import datetime
3
4 def isTimeFormat(input):
5     try:
6         time.strptime(input, '%H:%M:%S')
7         return True
8     except ValueError:
9         return False
10
11
12 def table_choice(cursor):
13     cursor.execute("SELECT table_schema, table_name FROM information_schema.tables WHERE (table_schema = 'public') ORDER BY table_schema, table_name;")
14     tables = cursor.fetchall()
15
16     integ = ''
```

pgAdmin 4 x databases_kp/menu_ops.py at m x +

github.com/auvy/databases_kp/blob/master/lab2/code/menu_ops.py

```
135     return columns
136
137 def get_column_names_only(columns):
138     newlist = [] * len(columns)
139     for entry in columns:
140         newlist.append(entry[0])
141     return newlist
142
143 def column_data_input(columns, cursor):
144     correctFormat = True;
145     data = ''
146     receivedVals = [] * len(columns)
147     for column in columns:
148         data = input('Enter value of [{0}]: '.format(column[0]))
149         correctFormat = format_check(str(column[1]), data)
150         while correctFormat == False:
151             data = input('Enter correct value of [{0}]: '.format(column[0]))
152             correctFormat = format_check(str(column[1]), data)
153         if str(column[1]) == 'time without time zone':
154             data = "\"" + data + "\"::time"
155         elif str(column[1]) == 'character varying':
156             data = "\"" + data + "\""
157         receivedVals.append(data)
158
159     return receivedVals
160
161
162 def print_all_values(tableName, cursor):
163     string = "SELECT * FROM " + tableName + " ORDER BY id"
164     cursor.execute(string)
165     entries = cursor.fetchall()
166     for entry in entries:
167         print(print_row(entry))
168     return entries
169
170 def print_row(row):
171     string = "[" + str(row[0]) + "]"
172     return string
```

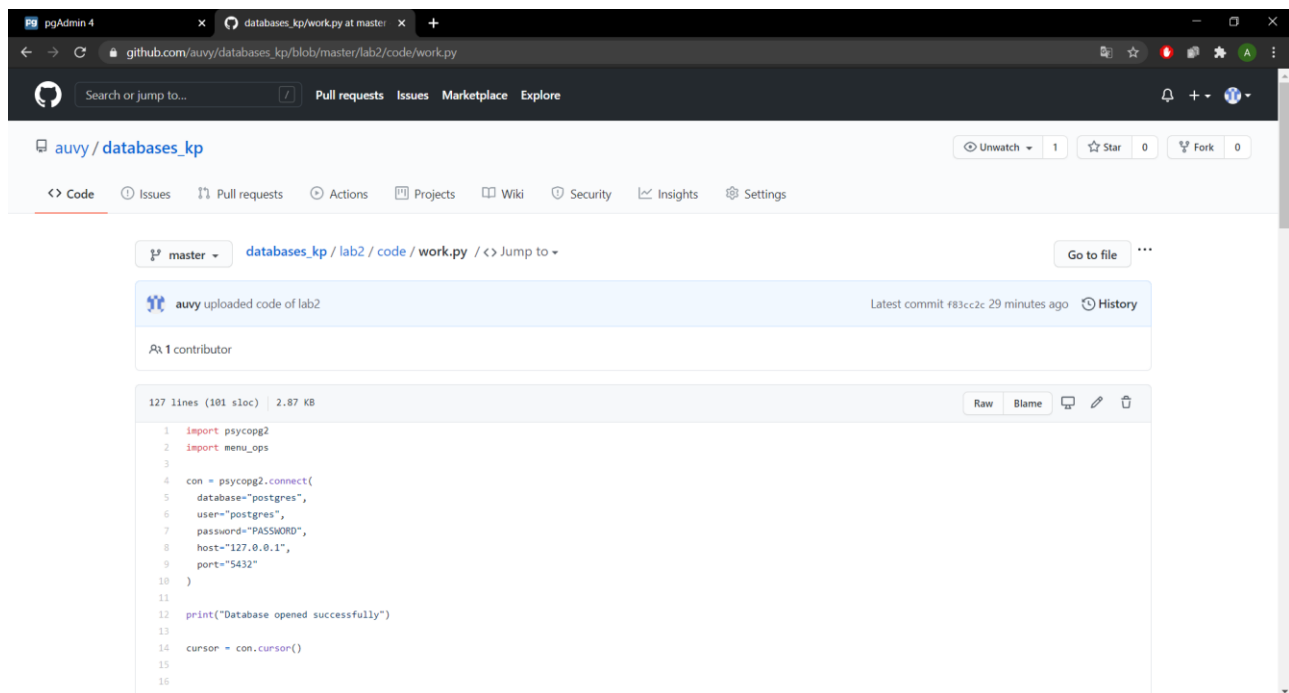


Рис. 4.1-5 Ілюстрації коду з репозиторію

Висновок

Виконавши дану лабораторну роботу були здобуті навички програмування мовою Python версії 3, проектування та програмування прикладних додатків на мові Python для роботи з базами даних PostgreSQL.