

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-91

Чорна Софія Олександрівна

*(прізвище, ім'я, по батькові)*

варіант №\_23\_

Перевірів

“\_\_\_\_\_” “\_\_\_\_\_” 20\_\_ р.

викладач

Петрашенко Андрій Васильович

*(прізвище, ім'я, по батькові)*

Київ 2020

*Мета роботи:*

Здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.

*Загальне завдання:*

1. Реалізувати функції внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі №1, 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).

*Репозиторій:*

<https://github.com/schernaya/databases/tree/master/lab2>

*Завдання №1:*

Enter a payment method: <i>Paypla</i>	Enter a payment method: <i>Paypal</i>
Enter a year: <i>2020</i>	Enter a year: <i>2020</i>
Enter a month: <i>1</i>	Enter a month: <i>23</i>
Enter a day: <i>12</i>	Enter a day: <i>22</i>
Enter a customer id: <i>2</i>	
Error! There is no such customer id!	It is not date. Try again:
	Enter a payment method: <i>mathod</i>
	Enter a quantity: <i>3</i>
Enter a name: <i>Name</i>	Enter a company: <i>company</i>
Enter a email: <i>email</i>	Query time: 0.004052639007568359
It is not email. Try again:	--- FORMS LIST ---
Enter a email:	There is no such forms.

Enter a form id: 633

Enter a phone id: 345

Error! There is no data with such ids.

Enter id: 1

There is no such customer.

680

Enter a company:

No such an option!

Nothing was entered. Try again:

Enter a price: price

It is not price.

Enter a price: |





Enter id: 23

Error! There is no data with such id.

## Завдання №2:

```
1 SELECT * FROM public.customer
2 ORDER BY id ASC
```

Data Output Explain Messages Notifications

	 id [PK] integer		name character varying (100)		email character varying (300)	
68		102	Nataliya Antonivna		user_68@hotmail.com	
69		103	Ruslan Anatolich		user_69@hotmail.com	
70		104	Tatiana Nikolaevna		user_70@yahoo.com	
71		105	Ruslan Anatolich		user_71@yahoo.com	
72		106	Ruslan Anatolich		user_72@ex.ua	
73		107	Ruslan Anatolich		user_73@yahoo.com	
74		108	Viktor Petrovich		user_74@hotmail.com	
75		109	Nataliya Antonivna		user_75@gmail.com	
76		110	Nataliya Antonivna		user_76@yahoo.com	
77		111	Ruslan Anatolich		user_77@hotmail.com	
78		112	Sophie		user_78@gmail.com	
79		113	Ruslan Anatolich		user_79@hotmail.com	
80		114	Tatiana Nikolaevna		user_80@yahoo.com	
81		115	Nataliya Antonivna		user_81@yahoo.com	
82		116	Ruslan Anatolich		user_82@hotmail.com	
83		117	Nataliya Antonivna		user_83@hotmail.com	
84		118	Nataliya Antonivna		user_84@hotmail.com	

```

1 SELECT * FROM public.form
2 ORDER BY id ASC

```

Data Output Explain Messages Notifications

	id [PK] integer	payment_method character varying (300)	ship_date date	customer_id integer
32	585	Yandex money	2020-04-12	48
33	586	WebMoney	2020-08-10	87
34	587	Privat24	2020-01-12	125
35	588	Yandex money	2020-04-06	114
36	589	Bitcoin Cash	2020-07-14	65
37	590	Bitcoin Cash	2020-05-26	69
38	592	WebMoney	2020-02-16	88
39	593	PayPal	2020-06-12	78
40	594	WebMoney	2020-07-23	86
41	595	QIWI	2019-11-13	43
42	597	Bitcoin Cash	2020-02-26	50
43	598	Bitcoin Cash	2020-07-02	109
44	599	WebMoney	2019-12-12	40
45	600	Yandex money	2020-04-28	85

```

1 SELECT * FROM public.form_phone
2 ORDER BY form_id ASC, phone_id ASC

```

Data Output Explain Messages Notifications

	form_id [PK] integer	phone_id [PK] integer	quantity integer
1	556	170	17
2	556	235	17
3	562	128	14
4	568	90	2
5	568	235	2
6	569	37	20
7	569	232	20
8	569	234	2
9	570	178	3
10	571	96	2
11	572	99	4
12	574	47	2

```
1 SELECT * FROM public.phone
2 ORDER BY id ASC
```

Data Output Explain Messages Notifications

	Id [PK] integer	model character varying (300)	company character varying (100)	price money
1	15	W1	Sony	490.45 ₴
2	16	H2	Sony	254.24 ₴
3	17	Y3	Samsung	314.55 ₴
4	18	E4	Huawei	290.77 ₴
5	19	G5	LG	116.29 ₴
6	20	N6	Xiaomi	398.18 ₴
7	21	T7	Sony	386.32 ₴
8	22	X8	IPhone	405.53 ₴
9	23	V9	LG	157.96 ₴
10	24	N10	Samsung	281.16 ₴

SQL-запити:

```
sql = "INSERT into customer (name, email) " \
      "SELECT " \
      "(case (random() * 5)::int " \
      "when 0 then 'Viktor Petrovich' " \
      "when 1 then 'Tatiana Nikolaevna' " \
      "when 2 then 'Ruslan Anatolich' " \
      "when 3 then 'Andrii Vasylovych' " \
      "when 4 then 'Nataliya Antonivna' " \
      "when 5 then 'Sophie Chorna' " \
      "when 6 then 'Irina Mikhailovna' " \
      "when 7 then 'Victoria Igorevna' " \
      "end) as name, " \
      "'user_' || seq || '@' || " \
      "(case (random() * 3)::int " \
      "when 0 then 'gmail.com' " \
      "when 1 then 'hotmail.com' " \
      "when 2 then 'yahoo.com' " \
      "when 3 then 'ex.ua' " \
      "end) as email " \
      "from GENERATE_SERIES(1, { }) as seq " \
      "RETURNING *".format(num)
```



```
sql = "insert into form(payment_method, ship_date, customer_id) " \
      "select " \
      "(case (random() * 7)::int " \
      "when 0 then 'PayPal' " \
      "when 1 then 'EasyPay' " \
      "when 2 then 'WebMoney' " \
      "when 3 then 'Privat24' " \
      "when 4 then 'Bitcoin Cash' " \
      "when 5 then 'Yandex money' " \
      "when 6 then 'Visa' " \
      "when 7 then 'QIWI' " \
      "end) as payment_method," \
      "((current_date - floor(random()* (365-0+ 1) + 0)*('1 day')::interval)::date)," \
      "fk_customer_id() " \
      "from GENERATE_SERIES(1, {}) ".format(num)
```

```
sql = "insert into phone(model, company, price) " \
      "select chr(trunc(65+random()*25)::int) || seq.a as company, " \
      "(case (random() * 5)::int " \
      "when 0 then 'Samsung' " \
      "when 1 then 'LG' " \
      "when 2 then 'iPhone' " \
      "when 3 then 'Sony' " \
      "when 4 then 'Xiaomi' " \
      "when 5 then 'Huawei' " \
      "end) as model, " \
      "(random()*(500-100+1)+100)::float8::numeric::money as price " \
      "from GENERATE_SERIES(1, {}) as seq(a)".format(num)
```

```
sql = "insert into form_phone (form_id, phone_id, quantity) " \
      "select fk_form_id(), fk_phone_id(), " \
      "floor(random() * (20 - 1) + 1) + 1" \
      "FROM generate_series(1, '{}')".format(num)
```

Були використані наступні функції:

#### `fk_customer_id()`

General	Definition	Code	Options	Parameters	Security	SQL
---------	------------	------	---------	------------	----------	-----

```
1 DECLARE
2     new_int int;
3 BEGIN
4     SELECT customer.id INTO new_int FROM customer
5     ORDER BY RANDOM() LIMIT 1;
6     RETURN new_int;
7 END;
```

### fk\_form\_id()

General   Definition   **Code**   Options   Parameters   Security

```
1 DECLARE
2     new_int int;
3 BEGIN
4     SELECT form.id INTO new_int FROM form
5     ORDER BY RANDOM() LIMIT 1;
6     RETURN new_int;
7 END;
```

### fk\_phone\_id()

General   Definition   **Code**   Options   Parameters   Security

```
1 DECLARE
2     new_int int;
3 BEGIN
4     SELECT phone.id INTO new_int FROM phone
5     ORDER BY RANDOM() LIMIT 1;
6     RETURN new_int;
7 END;
```

Enter number of random data: 100000

Query time: 1.2269222736358643

The data was successfully inserted!

*Завдання №3:*

Enter a search name: *Andr*  
Enter a search email: *user\_*  
Enter a date to:  
Enter a year: *2020*  
Enter a month: *01*  
Enter a day: *01*  
Enter a date from:  
Enter a year: *2020*  
Enter a month: *11*  
Enter a day: *11*  
Query time: 0.006637096405029297

--- CUSTOMERS LIST ---

1. id - 40 , name - Andrii Vasylovych , email - user_6@hotmail.com
2. id - 58 , name - Andrii Vasylovych , email - user_24@hotmail.com
3. id - 61 , name - Andrii Vasylovych , email - user_27@yahoo.com
4. id - 78 , name - Andrii Vasylovych , email - user_44@ex.ua
5. id - 83 , name - Andrii Vasylovych , email - user_49@gmail.com
6. id - 87 , name - Andrii Vasylovych , email - user_53@hotmail.com
7. id - 101 , name - Andrii Vasylovych , email - user_67@hotmail.com
8. id - 126 , name - Andrii Vasylovych , email - user_92@yahoo.com
9. id - 128 , name - Andrii Vasylovych , email - user_94@gmail.com
10. id - 130 , name - Andrii Vasylovych , email - user_96@yahoo.com
11. id - 134 , name - Andrii Vasylovych , email - user_100@gmail.com

Enter a payment method: *Bitc*  
Enter a quantity: *1*  
Enter a company: *Sams*  
Query time: 0.18550968170166016

--- FORMS LIST ---

1. id - 611 , payment method - Bitcoin Cash , ship date - 2020-05-30 , customer id - 61
2. id - 641 , payment method - Bitcoin Cash , ship date - 2020-05-15 , customer id - 70

Enter a price from: *100*  
Enter a price to: *1000*  
Enter a quantity: *2*  
Query time: 0.001695871353149414

--- PHONES LIST ---

1. id - 16 , model - H2 , company - Sony , price - 254.24 ₺
2. id - 47 , model - N11 , company - LG , price - 288.61 ₺
3. id - 56 , model - M20 , company - Sony , price - 423.88 ₺
4. id - 90 , model - H54 , company - Huawei , price - 153.36 ₺
5. id - 96 , model - E60 , company - iPhone , price - 322.17 ₺
6. id - 194 , model - E158 , company - Sony , price - 100.91 ₺
7. id - 222 , model - V186 , company - Sony , price - 457.42 ₺
8. id - 234 , model - Q198 , company - Huawei , price - 500.44 ₺
9. id - 235 , model - J199 , company - Xiaomi , price - 190.01 ₺
10. id - 236 , model - I200 , company - LG , price - 489.35 ₺



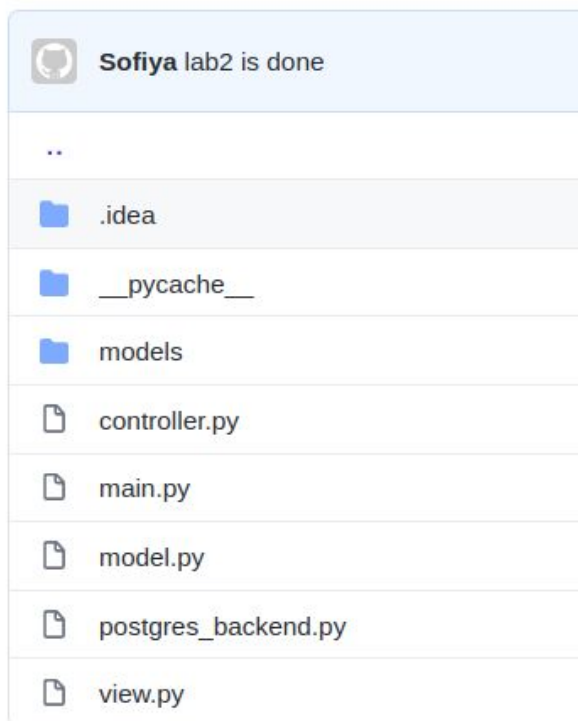
SQL-запити:


```
sql = "SELECT DISTINCT customer.id, name, email FROM customer " \
      "INNER JOIN form ON form.customer_id = customer.id " \
      "WHERE (name LIKE '%{}%') AND (email LIKE '%{}%') " \
      "AND (ship_date >= '{}') AND (ship_date <= '{}') " \
      "ORDER BY customer.id ASC". \

sql = "SELECT DISTINCT form.id, form.payment_method, " \
      "form.ship_date, form.customer_id FROM form " \
      "INNER JOIN form_phone ON form.id = form_phone.form_id " \
      "INNER JOIN phone ON phone.id = form_phone.phone_id " \
      "WHERE (payment_method LIKE '%{}%') AND (quantity = '{}') " \
      "AND (company LIKE '%{}%') " \
      "ORDER BY form.id".\

sql = "SELECT DISTINCT phone.id, phone.model, " \
      "phone.company, phone.price FROM phone " \
      "INNER JOIN form_phone ON phone.id = form_phone.phone_id " \
      "WHERE (price >= '{}') AND (price <= '{}') " \
      "AND (quantity = '{}') " \
      "ORDER BY phone.id".\
```

Завдання №4:




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

schernaya / databases

<> Code ⓘ Issues 🔗 Pull requests ⏮ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⚙ Settings


master databases / lab2 / controller.py / <> Jump to

 Sofiya lab2 is done

0 contributors

644 lines (520 sloc) 21.5 KB

```
1 import os
2 import datetime
3 import time
4
5 from model import Model
6 from view import View
7
8
9 class Controller(object):
10
11     def __init__(self):
12         self.view = View()
13         self.model = Model()
14
```

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

schernaya / databases

<> Code ⓘ Issues 🔗 Pull requests ⏮ Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⚙ Settings

master databases / lab2 / model.py / <> Jump to


 Sofiya lab1 is almost done

0 contributors

202 lines (165 sloc) 6.54 KB

```
1 import postgres_backend
2
3 from models.customer import Customer
4 from models.form import Form
5 from models.phone import Phone
6
7
8 class Model(object):
9
10     def __init__(self):
11         self._connection = postgres_backend.connect_to_db()
12
13     @property
14     def connection(self):
15         return self._connection
16
17     # read operations
18
19     def read_customer(self, id):
20         res = postgres_backend.select_customer(self.connection, id)
21         if res is None:
```

🔑 master ▾ databases / lab2 / view.py / <> Jump to ▾

 Sofiya lab2 is done

👤 0 contributors

268 lines (223 sloc) | 8.47 KB

```
1 class View(object):
2
3     # menu
4
5     @staticmethod
6     def display_menu():
7         return input('\nWhat would you like to work with?\n\n + '
8                     '\t(1) Customer\n + '
9                     '\t(2) Form\n + '
10                    '\t(3) Mobile phone\n + '
11                    '\t(0) Exit\n')
12
13     @staticmethod
```