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 2-го курсу
групи КП-93
Фещенко Єгор Олександрович

Перевірив
“” “вересня” 2020р.
викладач
Петрашенко Андрій Васильович

Київ 2020

Загальне завдання роботи полягає у наступному:

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).

URL репозиторія (код):
https://github.com/YehorFeshchenko/study/tree/master/Database%20Labs/lab2

URL репозиторія (звіт):

https://github.com/YehorFeshchenko/study/tree/master/Database%20Labs/lab2_doc

1) Валідація даних та обробка виключних ситуацій Приклад валідації деяких даних при додаванні нового запису до таблиць User та Group:

```
Choose action:
1)Add
2)Update
3)Delete
4)Generate
5)Find/Filter
6)Get
7)Set links
8)Delete links
Exit
Input number
1
Input first_name:
John
Input last_name:
Brown
Input age:
dd
Not a number!
Input age:
18
```

```
Input number
1
Input name:
school
Input number_of_members:
sdas
Not a number!
Input number_of_members:
10
Input date_of_creation(yyyy-mm-dd):
101000-2121-121212
Incorrect date format
Input date_of_creation(yyyy-mm-dd):
2020-03-03
```

Приклад помилок, які виникають у one-to-many при порушенні foreign key у таблиці Profile:

```
Input number
1
Input nickname:
Knight
Input date of registration(yyyy-mm-dd):
2020-01-01
Input country:
USA
Input user_id:
1000000
No user on this id
```

Приклад помилки перехопленою із БД:

```
Input number
7
Input Main Entity, then second
Input id:
200000
Input id:
200000
ОШИБКА:  INSERT или UPDATE в таблице "user_groups" нарушает ограничение внешнего ключа "fk_user_id"
DETAIL:  Ключ (user_id)=(200000) отсутствует в таблице "users".
```

Приклад обробки виключної ситуації при перегляді данних із таблиці User:

```
Choose action:
1)Add
2)Update
3)Delete
4)Generate
5)Find/Filter
6)Get
7)Set links
8)Delete links
Exit
Input number
6
Input id:
200000
'NoneType' object is not subscriptable
No user on this id
```

Приклад обробки помилок при вилученні даних у таблиці User:

```
Choose action:
1)Add
2)Update
3)Delete
4)Generate
5)Find/Filter
6)Get
7)Set links
8)Delete links
Exit
Input number
3
Input id:
200000
```

2) Приклад згенерованих даних у таблицях Groups та Profiles

Query Editor | Query History

```
1  Select * from groups
2
```

Data Output | Explain | Messages | Notifications

	group_id [PK] integer	name text	number of members integer	date of creation date
12	12	a2764f71f7...	439	2010-09-12
13	13	3524f97bb...	514	2001-01-20
14	14	f23c59a37...	50	2005-03-13
15	15	a92fd9648...	59	2018-04-22
16	16	e90dafb0a...	373	2003-12-11
17	17	631939102...	665	2001-05-09
18	18	e57e3796f...	98	2014-01-11
19	19	cb8769e6a...	687	2002-09-08
20	20	2f925803f6...	67	2017-07-05
21	21	877ddd85a...	356	2012-12-25
22	22	4af11dee1...	706	2001-04-19
23	23	37390e0bf...	609	2013-06-13
24	24	c4fa80077...	528	2004-06-28
25	25	39b36f43bf...	924	2018-12-10
26	26	cfb0c37a3...	171	2002-11-28
27	27	d52a9a358...	584	2015-12-27
28	28	ae4b1be42...	264	2008-11-19
29	29	25cee44af...	419	2006-08-17
30	30	a8fde8476...	816	2014-08-03
31	31	fba4781c9...	132	2019-02-07
32	32	16bb1c5ae...	691	2007-05-13

Query Editor | Query History

```
1  Select * from profiles
2
```

Data Output | Explain | Messages | Notifications

	profile_id [PK] integer	nickname text	date of registration date	country text	user_id integer
1	115216	e1631a328c3...	2005-10-27	bd181d7f2...	3
2	115217	c4203a37a98...	2003-08-15	ad25e1379...	3
3	115218	e6d14661fb5...	2007-03-04	8dabf3b20...	2
4	115219	085a570bf88...	2012-07-30	56fbff43e5...	3
5	115220	8e5917fa6aa...	2008-12-07	bfeaddfde5...	7
6	115221	8a8700b3b2f...	2010-06-19	9c5c6d0f53...	1
7	115222	55f304ffbd69...	2018-04-29	99bc8f580e...	2
8	115223	164ca7b6c52...	2005-09-25	32591d41c...	7
9	115224	53c8efadea3b...	2007-08-21	d4a9c0f24...	4
10	115225	f6ed2755f98e...	2012-08-11	ed833a547...	2
11	115226	0347ce09b1d...	2014-02-24	86dd500f6...	6
12	115227	190a94ecfa9a...	2020-04-25	fbe614931...	7
13	115228	8c56f5d25f58...	2003-08-28	805d54f0fb...	1
14	115229	54b7b7a863b...	2018-09-21	715e2c635...	1
15	115230	3105de59d51...	2006-03-05	3904a42f7...	4
16	115231	78e6f2c49f25...	2001-11-30	31447510c...	2
17	115232	445afb84e7bf...	2011-08-07	37bd97edf...	3
18	115233	6e920bc0de1...	2009-09-03	5ae01b4dd...	5
19	115234	da0fed58808...	2020-07-11	fb2fad29ce...	3
20	115235	29729344509...	2010-09-02	890bf4716...	1
21	115236	903af98d596...	2001-07-13	89b8fcaab1...	7

Запити, які генерують рандомізовані дані:

```
request = 'INSERT INTO groups(name, "number of members", "date of creation") ' \
          'SELECT MD5(random()::text), trunc(random()*%s)::int, timestamp \'1-1-1\' ' \
          '+ random()*(timestamp \'2020-10-10\' - timestamp \'1-1-1\') FROM generate_series(1 , %s)'
data = (number, number)
```

```
request = 'INSERT INTO profiles(nickname, "date of registration", country, user_id) ' \
          'SELECT MD5(random()::text), timestamp \'1-1-1\' + random()*(timestamp \'2020-10-10\' - ' \
          'timestamp \'1-1-1\'), MD5(random()::text), ' \
          'trunc(random()*(SELECT MAX(user_id) FROM "users")::int '\
          'FROM generate_series(1 , %s)'
data = (number,)
```

```
request = 'INSERT INTO users("first name", "last name", age) SELECT MD5(random()::text), MD5(random()::text),' \
          ' trunc(random()*%s)::int FROM generate_series(1 , %s)'
data = (number, number)
```

```
request = 'INSERT INTO posts(topic, "date of publishing", owner, profile_id) ' \
          'SELECT MD5(random()::text), timestamp \'1-1-1\' + random()*(timestamp \'2020-10-10\' - ' \
          'timestamp \'1-1-1\'), MD5(random()::text), ' \
          'trunc(random()*(SELECT MAX(profile_id) FROM "profiles")::int ' \
          'FROM generate_series(1 , %s)'
data = (number,)
```

3)Приклад пошукового запиту у User

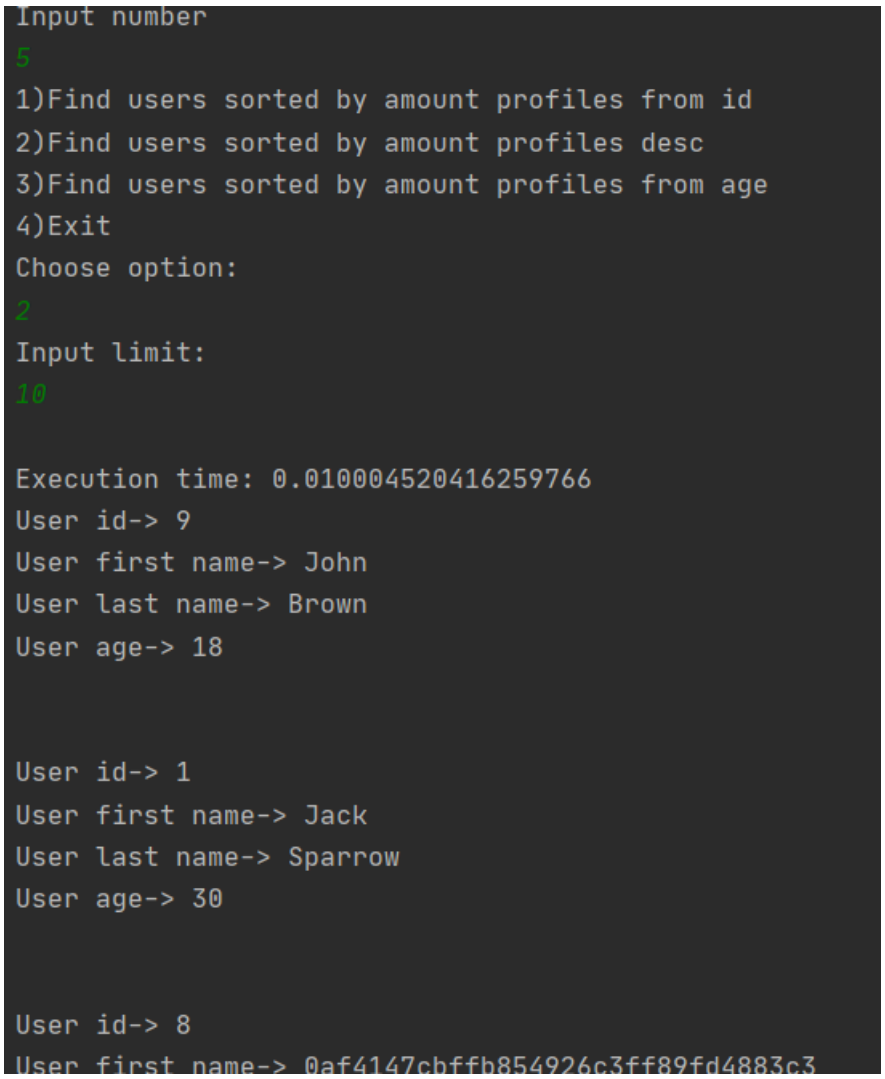
```
1)Find users sorted by amount profiles from id
2)Find users sorted by amount profiles desc
3)Find users sorted by amount profiles from age
4)Exit
Choose option:
1
Input id:
1
Input id:
10

Execution time: 0.009971857070922852
User id-> 9
User first name-> John
User last name-> Brown
User age-> 18


User id-> 1
User first name-> Jack
User last name-> Sparrow
User age-> 30


User id-> 8
User first name-> 0af4147cbffb854926c3ff89fd4883c3
```

Приклад пошукового запиту у User



Приклад пошукового запиту у User

```
1)Find users sorted by amount profiles from id
2)Find users sorted by amount profiles desc
3)Find users sorted by amount profiles from age
4)Exit
Choose option:
3
Input age:
10
Input age:
12

Execution time: 0.0009999275207519531
User id-> 4
User first name-> Jack
User last name-> Sparrow
User age-> 12


Choose number of option:
```

Приклад пошукового запиту у Group

```
Input number
5
1) Find not empty groups sorted by amount of user
2) Find not empty groups sorted by user's age
3) Exit
Choose option:
1

Execution time: 0.0060122013092041016
Group id-> 91110
Group name-> 861b37cce2fcb1cdde223b1d03511a25
Group date_of_creation-> 2005-07-01
Group number_of_members-> 7


Group id-> 91104
Group name-> 2abb56ad4c3665802ba171bef221c9b5
Group date_of_creation-> 2009-04-15
```

Приклад пошукового запиту у Group

```
Input number
5
1) Find not empty groups sorted by amount of users
2) Find not empty groups sorted by user's age
3) Exit
Choose option:
2
Input age:
10
Input age:
40

Execution time: 0.0010266304016113281
Group id-> 91011
Group name-> d14f1cf853b4217ea213af9dfc0345d0
Group date of creation -> 2010-04-08
Group number of members-> 86
User last name -> Sparrow


Group id-> 91012
Group name-> c23dc7f05bbeac47fbc42324e566d4c1
Group date of creation -> 2019-03-11
Group number of members-> 50
User last name -> Sparrow
```

Приклад пошукового запиту у Post

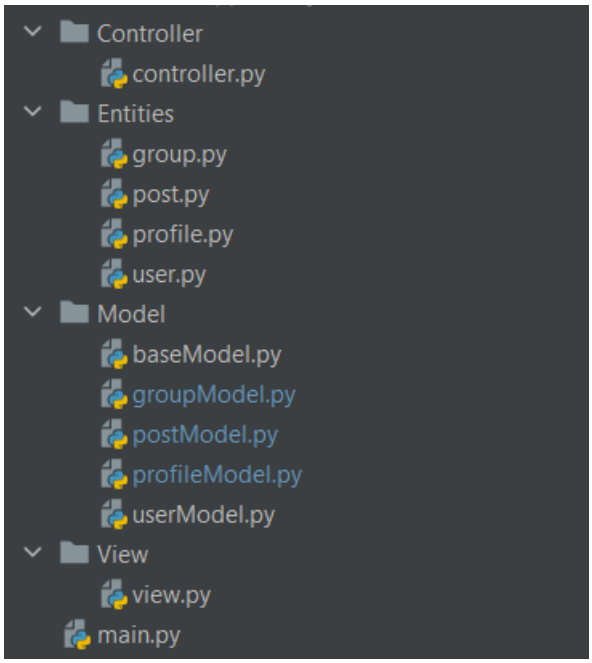
```
Exit
Input number
5

Execution time: 0.001964092254638672
topic-> fbbc98b280f2d91814c9b47c70542730
nickname-> 5ce6fd6d29de726f97786628503afd21


topic-> f92e4fe8a7e2d765e9b4751f90aaca39
nickname-> 2881bf6eb4ac3b7acd1ea921394e05e7


topic-> f8274cfd7732f5260e4c276897378e8e
nickname-> 1fc7e03f46c7f6bc909a8def7042a4bb
```

4) Приклад файлової структури коду MVC



Entities містить описані об’єкти таблиці БД

Файл view.py містить статичні методи класу, які відповідають за обробку вводу клієнта та відображення потрібної інформації

```python
import datetime
from Entities.user import User
from Entities.profile import Profile
from Entities.post import Post
from Entities.group import Group


class View:

    @staticmethod
    def set_link_print():
        print("Input Main Entity, then second")

    @staticmethod
    def delete_link_print():
        print("Input Main Entity id")

    @staticmethod
    def id_find():
        input_id = input("Input id:\n")
        if input_id.isdigit() is False:
            print("Not a number")
            return -1
        else:
            return int(input_id)

    @staticmethod
    def update_profile(update_profile):
        while True:
            info = input("Choose what field do you want to change:\n1)nickname\n2)date_of_registration\n"
                         "3)country\n4)user_id\n5)Exit\n")

            if info == '1':
                update_profile.nickname = input("Input nickname\n")
            elif info == '2':
                try:
                    date = datetime.datetime.strptime(input("Input date of registration(yyyy-mm-dd):\n"),
                                                      '%Y-%m-%d')
                except:
```

Файл controller.py керує викликом статичних методів та приймає рішення на основі вводу користувача

```python
from View.view import View


def menu(userModel, profileModel, postModel, groupModel):
    in_menu = True
    while in_menu:
        info = View.main_menu()
        if info == 'error':
            print("Incorrect input")
            continue
        if info == '1':
            while True:
                sub_info = View.sub_menu()
                if sub_info == -1:
                    continue
                else:
                    break
            if sub_info == '1':
                user = View.add_user()
                userModel.add_entity(user)
                continue
            elif sub_info == '2':
                while True:
                    input_id = View.id_find()
                    if input_id == -1:
                        continue
                    else:
                        break
                temp_user = userModel.get_entity(input_id)
                if temp_user is None:
                    print("No user on this id!")
                else:
                    userModel.update_entity(View.update_user(temp_user))
            elif sub_info == '3':
                while True:
                    input_id = View.id_find()
                    if input_id == -1:
                        continue
```

Файли моделей(один базовий клас та чотири наслідники) містять методи для спілкування із СУБД та створення запитів

```python
from abc import ABC, abstractmethod
import psycopg2


class BaseModel(ABC):
    def __init__(self, dbname, user, password, host):
        self.dbname_ = dbname
        self.user_ = user
        self.password_ = password
        self.host_ = host
        try:
            self.conn = psycopg2.connect(dbname=self.dbname_, user=self.user_, password=self.password_, host=self.host_)
        except (Exception, psycopg2.DatabaseError) as error:
            print(error)

    def __del__(self):
        try:
            self.conn.close()
        except (Exception, psycopg2.DatabaseError) as error:
            print(error)

    @abstractmethod
    def add_entity(self, new_entity):
        pass

    @abstractmethod
    def get_entity(self, entity_id):
        pass

    @abstractmethod
    def get_entities(self):
        pass

    @abstractmethod
```

```python
from Model.baseModel import BaseModel
from Entities.user import User
import psycopg2
import time


def does_contain(entity_id, data_list):
    contains = False
    for record in data_list:
        if entity_id == record.id:
            contains = True
    return contains


class UserModel(BaseModel):
    def __init__(self, dbname, user, password, host):
        super(UserModel, self).__init__(dbname, user, password, host)
        try:
            self.cursor = self.conn.cursor()
        except (Exception, psycopg2.DatabaseError) as error:
            self.cursor.execute('ROLLBACK')
            print(error)

    def __del__(self):
        try:
            self.cursor.close()
            self.conn.close()
        except (Exception, psycopg2.DatabaseError) as error:
            self.cursor.execute('ROLLBACK')
            print(error)

    def get_entities(self):
        request = 'SELECT * FROM users'
        users = list()
```

Файл main.py підключає моделі до СУБД та викликає функцію контролера

```python
from Controller.controller import menu
from Model.userModel import UserModel
from Model.groupModel import GroupModel
from Model.postModel import PostModel
from Model.profileModel import ProfileModel


def get_password():
    f = open(r"D:\Job\Apps\password.txt", "r")
    data = f.read()
    f.close()
    return data


password = get_password()

groupModel = GroupModel('Lab1_db', 'postgres', password, 'localhost')
postModel = PostModel('Lab1_db', 'postgres', password, 'localhost')
userModel = UserModel('Lab1_db', 'postgres', password, 'localhost')
profileModel = ProfileModel('Lab1_db', 'postgres', password, 'localhost')

menu(userModel, profileModel, postModel, groupModel)

userModel.__del__()
postModel.__del__()
groupModel.__del__()
profileModel.__del__()
```