

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)	He	ревірив
студент(ка) 2 курсу		20 p.
групи КП-91	В	икладач
Лопаткіна Катерина Олегівна (прізвище, ім'я, по батькові)	Петрашенко Андрій Васі (прізвище, ім'я, по батькові)	
варіант Школа		
(предмети, вчителі, оцінки, учні, група)		

Мета

 $Mетою poбот \in 3$ добуття вмінь програмування прикладних додатків баз даних PostgreSQL.

Загальні вимоги до завдання

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).

Репозиторій:

https://github.com/katerynaLopatkina/Database/tree/main/lab2

Пункт 1

```
Student menu:
Student menu:
                                       1. Show all students;
    1. Show all students;
                                       2. Display student by id
    2. Display student by id
                                       Add student;
    Add student;
                                       4. Update student
                                       Delete student;
    4. Update student
                                       Generate students;
    5. Delete student;
                                       7. Add subject to student;
    Generate students;
                                       0. Exit
    7. Add subject to student;
    0. Exit
                                    Enter student first name : Katya
                                   Enter student last name : Lopatkina
                                    Enter student sex : Female
Enter student id : 5
                                   Enter group id : 8
There is no such student.
                                   Enter year : 2002
                                   Enter month : 6
                                    Enter day : 15
                                    Your provided foreign ids are incorrect
```

Student menu:

```
Student menu:
   1. Show all students;
   2. Display student by id
   3. Add student;
   4. Update student
   5. Delete student;
   6. Generate students;
   7. Add subject to student;
   0. Exit
Enter student first name : Katya
Enter student last name : Lopatkina
Enter student sex : Female
Enter group id : 2
Enter year : 2
Enter month : 1221
Enter day : 454
Your provided value is not a date
Enter year :
```

```
Student menu:

1. Show all students;

2. Display student by id

3. Add student;

4. Update student

5. Delete student;

6. Generate students;

7. Add subject to student;

0. Exit

45

There is no such menu option
```

```
Student menu:

1. Show all students;

2. Display student by id

3. Add student;

4. Update student

5. Delete student;

6. Generate students;

7. Add subject to student;

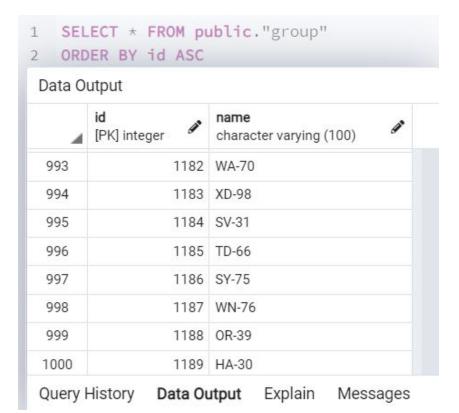
0. Exit

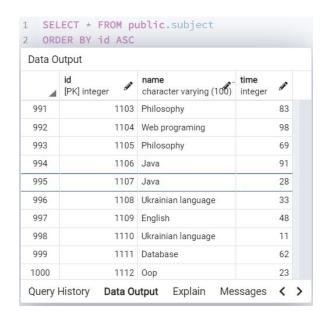
1

--- STUDENTS LIST ---
There is no items in this table - student
```

Пунт 2

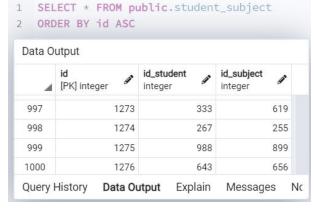
ata O	utput						
4	id [PK] integer	first_name character varying (100)	last_name character varying (100)	sex character varying (100)	id_group	date_of_birth	
1	81	Dima	Lahno	Female	1185	2003-03-06	
2	82	Katya	Lopatkina	Female	547	2007-02-14	
3	83	Dima	Lahno	Male	848	2007-04-22	
4	84	Dima	Kulish	Male	717	2011-10-31	
5	85	Olya	Shupik	Male	956	2002-05-19	
6	86	Olya	Kulish	Female	362	2015-07-28	
7	87	Katya	Kulish	Female	525	2002-11-05	
8	88	Andrey	Lahno	Female	937	2001-02-07	
9	89	Dima	Kulish	Female	759	2011-04-04	
10	90	Dima	Lahno	Male	802	2005-05-31	
11	91	Dima	Kulish	Female	724	2011-05-12	
12	92	Andrey	Lahno	Male	705	2018-01-15	
13	93	Katya	Lopatkina	Male	777	2004-11-10	
14	94	Andrey	Lopatkina	Male	1129	2004-02-05	
15	95	Andrey	Lahno	Male	442	2006-08-20	





)ata (Output		
4	id [PK] integer	id_teacher integer	id_subject integer
996	1065	282	659
997	1066	954	749
998	1067	615	180
999	1068	936	115

1 SELECT * FROM public.teacher 2 ORDER BY id ASC Data Output phone_number id first_name last_name [PK] integer character varying (100) character varying (100) character varying (13) 100991 380294912408 101103 Pavel Malyutin 100992 101104 Dmytro Nazarenko 380614611862 100993 101105 Dmytro 380522757686 Legeza 100994 101106 Oleg Melnik 380372146664 100995 380304973559 101107 Pavel Legeza 100996 101108 Dmytro 380632047148 Nazarenko 100997 101109 Pavel Legeza 380467696861 100998 101110 Dmytro Malyutin 380418622625 101111 Oleg 100999 Nazarenko 380711311371 101112 Pavel 101000 Malyutin 380277225854 Query History Data Output Explain Messages Notifications < >



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

4	id [PK] integer	id_subject integer	id_group integer
997	1145	673	654
998	1146	352	1157
999	1147	736	764
1000	1148	850	737
Ouerv I	History Data Ou	utput Explain	Messag <

Копії SQL запитів:

Генерація випадкових груп

Генерація випадкових предметів

```
sql = "INSERT into subject (name, time) " \
    "SELECT " \
    "(case (random() * 8)::int " \
    "when 0 then 'Oop' " \
    "when 1 then 'Database' " \
    "when 2 then 'Physics' " \
    "when 3 then 'Philosophy' " \
    "when 4 then 'Web programing' " \
    "when 5 then 'Java' " \
    "when 6 then 'Math' " \
    "when 7 then 'English' " \
    "when 8 then 'Ukrainian language' " \
    "end) as name, " \
    "(random() * 100) as time " \
    "from GENERATE_SERIES(1, {}) as seq " \
    "RETURNING *".format(count)
```

Генерація випадкових студентів

```
sql = "INSERT into student (first_name, last_name, sex, id_group, date_of_birth) " \
    "SELECT " \
    "(case (random() * 3)::int " \
    "when 0 then 'Katya' " \
    "when 1 then 'Dima' " \
    "when 2 then 'Andrey' " \
    "when 3 then 'Olya' " \
    "end) as first_name, " \
    "(case (random() * 3)::int " \
    "when 0 then 'Lopatkina' " \
    "when 1 then 'Lahno' " \
    "when 2 then 'Kulish' " \
    "when 3 then 'Shupik' " \
    "end) as last_name, " \
    "(case (random() * 1)::int " \
    "when 0 then 'Female' " \
    "when 1 then 'Male' end) " \
    "as sex, " \
    "get_group_id() as id_group, " \
    "((NOW() + (random() * (NOW() - INTERVAL '20 years' - NOW())))::date) as date_of_birth " \
    "from GENERATE_SERIES(1, {}) as seq " \
    "RETURNING *".format(count)
```

Генерація випадкових вчителів

```
sql = "INSERT into teacher (first_name, last_name, phone_number) " \
    "SELECT " \
    "(case (random() * 3)::int " \
    "when 0 then 'Oleg' " \
    "when 1 then 'Dmytro' " \
    "when 3 then 'Olva' " \
    "end) as first_name, " \
    "(case (random() * 3)::int " \
    "when 0 then 'Melnik' " \
    "when 1 then 'Malyutin' " \
    "when 2 then 'Nazarenko' " \
    "when 3 then 'Legeza' " \
    "end) as last_name, " \
    "('380' " \
    "|| (RANDOM() * 9)::INT " \
```

Генерація випадкових зв'язків вчителів і предметів

Генерація випадкових зв'язків учнів і предметів

Генерація випадкових зв'язків учнів і предметів

Для генерації зв'язків між таблицями потрібно було отримувати їх ідентифікатори, і для цього було створено додаткові функції в базі даних.

```
General Definition Code Options Parameters Security

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

```
General Definition Code Options Parameters Security

DECLARE
student_id int;
BEGIN
SELECT student.id INTO student_id FROM student
ORDER BY RANDOM() LIMIT 1;
RETURN student_id;
FNO;
```

```
{=} get_subject_id()
         Definition Code Options Parameters Security S
General
 1 DECLARE
      subject_id int;
 3 BEGIN
      SELECT subject.id INTO subject_id FROM subject
      ORDER BY RANDOM() LIMIT 1;
      RETURN subject_id;
 7 END;
(=) get_teacher_id()
General Definition
                       Code Options Parameters Security
 1 DECLARE
      teacher_id int;
 3 BEGIN
       SELECT teacher.id INTO teacher_id FROM teacher
 4
      ORDER BY RANDOM() LIMIT 1;
 5
      RETURN teacher_id;
 7 END;
Teacher menu:
  1. Show all teachers;
  2. Display teacher by id
  4. Update teacher
  5. Delete teacher;
  6. Generate teachers;
  7. Add subject to teacher;
Enter teacher Amount of generated data : 100000
* 99994. id - 101106, first name - Oleg, last name - Melnik, phone number - 380372146664
* 99995. id - 101107, first name - Pavel, last name - Legeza, phone number - 380304973559
* 99996. id - 101108, first name - Dmytro, last name - Nazarenko, phone number - 380632047148
* 99997. id - 101109, first name - Pavel, last name - Legeza, phone number - 380467696861
* 99998. id - 101110, first name - Dmytro, last name - Malyutin, phone number - 380418622625
* 99999. id - 101111, first name - Oleg, last name - Nazarenko, phone number - 380711311371
* 100000. id - 101112, first name - Pavel, last name - Malyutin, phone number - 380277225854
```

Пункт 3

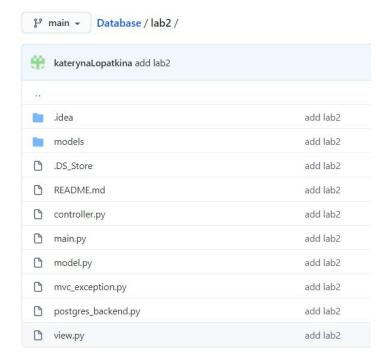
```
Enter student first name
Enter student sex :
Enter start date
Enter year
Enter month
Enter day :
Enter end date
Enter day
* 1. id - 139, first name - Katya, last name - Kulish, sex - Femaledate of birth - 2014-09-24, group name - RS-90, group id - 1132
* 2. id - 341, first name - Katya, last name - Kulish, sex - Femaledate of birth - 2014-11-02, group name - SP-72, group id - 1168
* 4. id - 381, first name - Katya, last name - Lahno, sex - Femaledate of birth - 2010-05-15, group name - NV-98, group id - 797
* 8. id - 673, first name - Katya, last name - Kulish, sex - Femaledate of birth - 2012-11-14, group name - PY-84, group id - 351
* 11. id - 780, first name - Katya, last name - Lopatkina, sex - Femaledate of birth - 2011-05-16, group name - LA-63, group id - 72
* 12. id - 798, first name - Katya, last name - Shupik, sex - Femaledate of birth - 2012-09-01, group name - MB-69, group id - 678
* 13. id - 804, first name - Katya, last name - Lahno, sex - Femaledate of birth - 2014-10-29, group name - KH-84, group id - 1019
* 15. id - 884, first name - Katya, last name - Shupik, sex - Femaledate of birth - 2014-12-01, group name - FD-83, group id - 350
* 18. id - 991, first name - Katya, last name - Lopatkina, sex - Femaledate of birth - 2015-02-28, group name - EV-1, group id - 117
         "order by s.id ASC".format(scrub(first_name), scrub(sex), star_date, end_date)
     1. Search students
     2. Search teachers;
     3. Search students by teacher and group;
 Enter teacher first name : 0
 Enter teacher Subject name :
 Enter subject max credits :
 Enter subject min credits :
 * 1. id - 319, first name - Dmytro, last name - Melnik, phone number - 380836156357subject name - Oop, subject credits - 41, subject id - 305 * 2. id - 1027, first name - Dmytro, last name - Legeza, phone number - 380716065094subject name - Oop, subject credits - 45, subject id - 1065
 * 3. id - 237, first name - Dmytro, last name - Malyutin, phone number - 380564282776subject name - Oop, subject credits - 41, subject id - 1073 * 4. id - 454, first name - Dmytro, last name - Malyutin, phone number - 380944167220subject name - Oop, subject credits - 41, subject id - 1073
   sql = "SELECT teacher.id, first_name, last_name, phone_number, " \)
```

.format(scrub(first_name), scrub(subject_name), min_credits, max_credits)

```
Enter teacher first name : 01ga
Enter teacher Group name : YI-79
Enter start date
Enter year : 2002
Enter month : 5
Enter day : 5
Enter end date
Enter year : 2020
Enter month : 5
Enter day : 6
Enter day : 7
Enter day : 8
Enter day : 8
Enter day : 9
Enter
```

```
sql = "SELECT DISTINCT s.id as id_student, s.first_name as first_name_s, s.last_name as last_name_s," \
    " s.date_of_birth, gr.name as group_name, " \
    "gr.id as id_group, sb.id as id_subject, sb.name as sb_name, teach.id as id_teacher, " \
    "teach.first_name as first_name_t, " \
    "teach.last_name as last_name_t " \
    "from " \
    "student as s " \
    "inner join \"group\" as gr on s.id_group = gr.id " \
    "inner join subject_group as sg on sg.id_group = gr.id " \
    "inner join subject as sb on sg.id_subject = sb.id " \
    "inner join teacher_subject as ts on ts.id_subject = sb.id " \
    "inner join teacher as teach on teach.id = ts.id_teacher " \
    "where teach.first_name like '%{}%' " \
    "and gr.name like '%{}%' " \
    "and s.date_of_birth >= '{}' " \
    "and s.date_of_birth <= '{}' " \
    "order by s.id asc" \
    .format(scrub(first_name), group_name, start_date, end_date)</pre>
```

Пункт 4



மீ main ▼

Database / lab2 / model.py / <> Jump to

A 1 contributor

katerynaLopatkina add lab2

```
226 lines (197 sloc) 8.68 KB
     import postgres_backend
  3 from models.student import Student
  4 from models.subject import Subject
  5 from models.grade import Grade
  6 from models.group import Group
  7 from models.teacher import Teacher
  8 from models.teacher_subject import TeacherSubject
  9 from models.subject_group import SubjectGroup
 10 from models.student_subject import StudentSubject
 12 GROUP_TABLE = "\"group\""
 13 STUDENT_TABLE = "student"
 14 TEACHER_TABLE = "teacher"
 15 SUBJECT_TABLE = "subject"
 16 GRADE_TABLE = "grade"
 17 SUBJECT_GROUP_TABLE = "subject_group"
     TEACHER_SUBJECT_TABLE = "teacher_subject"
     STUDENT_SUBJECT_TABLE = "student_subject"
      def get_value_if_exists(res, key):
          if key in res:
             return res[key]
         else:
             return None
      class Model(object):
          def __init__(self):
              self._cursor = postgres_backend.connect_db()
          def connection(self):
             return self._cursor
```

γ main ≠

Database / lab2 / view.py / ⟨> Jump to ▼



katerynaLopatkina add lab2

८३ 1 contributor

```
352 lines (309 sloc) | 15.1 KB
      from model import GRADE_TABLE
  2 from model import GROUP_TABLE
     from model import SUBJECT_TABLE
     from model import STUDENT_TABLE
     from model import TEACHER_TABLE
     from model import TEACHER_SUBJECT_TABLE
      from model import STUDENT_SUBJECT_TABLE
  8
      from model import SUBJECT GROUP TABLE
 10
      class View(object):
          @staticmethod
          def show_main_menu():
             return input("Main menu:\n"
                           "\t1. Student\n"
                           "\t2. Teacher; \n"
                           "\t3. Subject;\n"
                           "\t4. Group;\n"
                           "\t5. Subject Group;\n"
 20
                           "\t6. Teacher Subject;\n"
                           "\t7. Student Subject;\n"
                           "\t8. Search menu;\n"
                           "\t0. Exit\n")
          @staticmethod
          def show search menu():
             return input("Search menu:\n"
                           "\t1. Search students\n"
                           "\t2. Search teachers;\n"
                           "\t3. Search students by teacher and group;\n"
                           "\t0. Exit\n")
```



katerynaLopatkina add lab2

A३ 1 contributor

```
670 lines (570 sloc) 26.4 KB
     import os
  2 import datetime
  3 import time
  5 from model import GROUP_TABLE
  6 from model import GRADE_TABLE
  7 from model import STUDENT_TABLE
  8 from model import SUBJECT TABLE
  9 from model import TEACHER_TABLE
 10 from model import TEACHER_SUBJECT_TABLE
 11 from model import STUDENT_SUBJECT_TABLE
 12 from model import SUBJECT_GROUP_TABLE
 13 from model import Model
 14 from view import View
 17 class Controller(object):
 18 MAIN_M = -1
       \# GRADE M = 5
 20 STUDENT M = 1
       TEACHER_M = 2
       SUBJECT_M = 3
       GROUP_M = 4
       SUBJECT_GROUP_M = 5
TEACHER_SUBJECT_M = 6
       STUDENT_SUBJECT_M = 7
       SEARCH_MENU = 8
       def __init__(self):
           self.view = View()
 30
            self.model = Model()
        def start(self):
 34
             current_menu = self.MAIN_M
             while current_menu == self.MAIN_M:
                 menu_option = self.view.show_main_menu()
                 if menu_option.isdigit():
 40
                     while int(menu_option) == self.GROUP_M:
 41
                         group_option = self.view.show_group_menu()
                        if group_option.isdigit():
 43
                            current_menu = self.handle_common_inner_menu(int(group_option), GROUP_TABLE, self.GROUP_M)
```