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”

Виконала
студентка II курсу
групи КП-02
Гудзіцька Дарина Сергіївна
варіант №3

Київ 2021

Мета роботи

Здобути вміння програмування прикладних додатків баз даних PostgreSQL.

Постановка завдання

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-контролер).

Вимоги до пункту №1 деталізованого завдання:

```
Введите действие:
add student

Введите данные:
lsfsekf 5
Not correct input data
```

```
Введите действие:
delete student

Введите данные:
dg
Not correct input data
```

```python
elif command_input[0] == "delete":
    print("\nВведите данные:")
    command = input()
    if len(command.split(' ')) == 1 and (isinstance(command.split(' ')[0], int)):
        if command_input[1] == "student":
            control.delete_student(command, connection)
            continue
        elif command_input[1] == "group":
            control.delete_group(command, connection)
            continue
        elif command_input[1] == "subject":
            control.delete_subject(command, connection)
            continue
        else:
            print("Таблица не найдена!")
            continue
    else:
        print("Not correct input data")
        continue
```

Вимоги до пункту №2 деталізованого завдання:

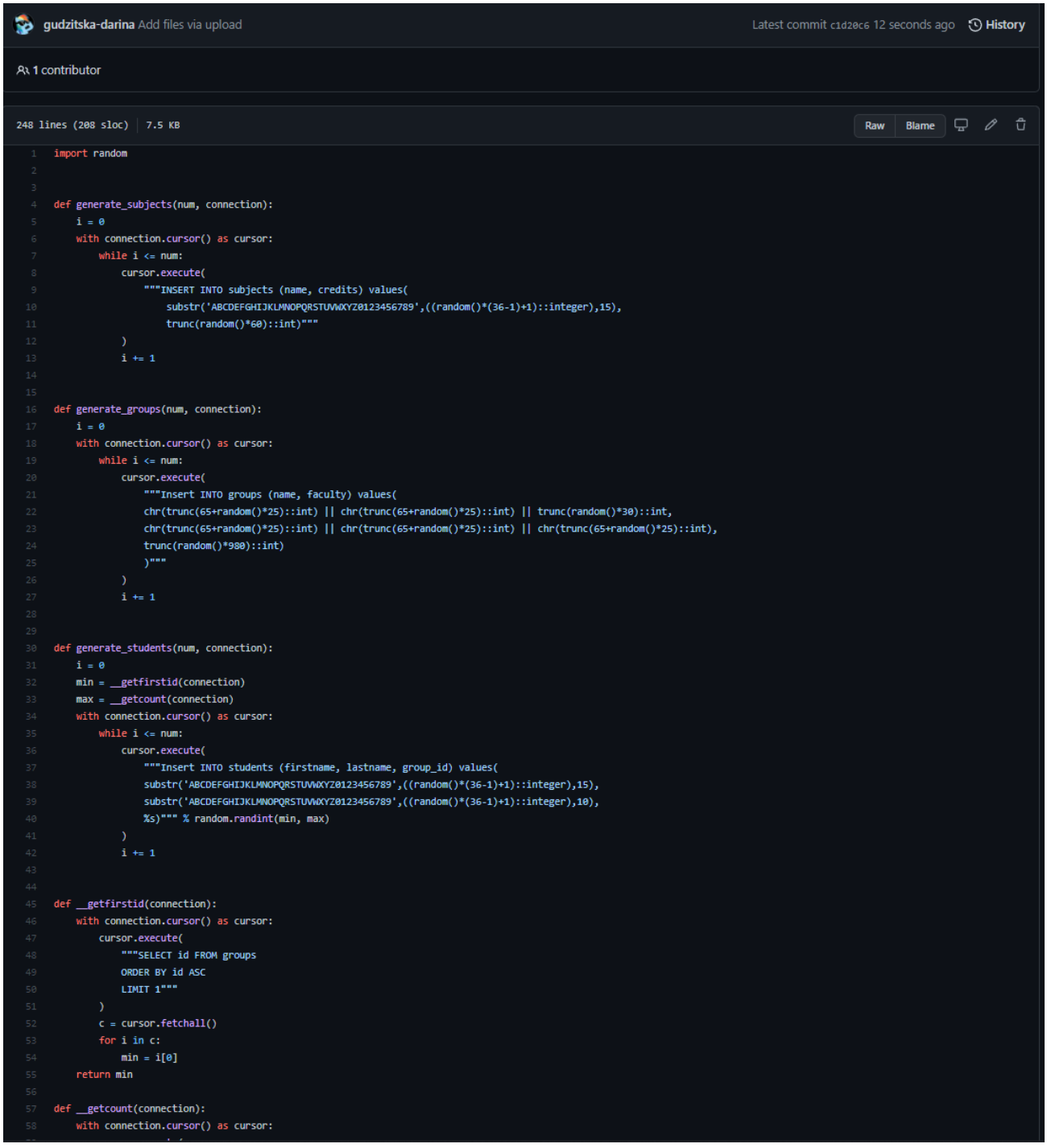
	id [PK] integer	firstname character varying (30)	lastname character varying (30)	group_id integer
2442	2442	LMNOPQRSTUVWXYZ	MNOPQRSTUV	5
2443	2443	0123456789	3456789	5
2444	2444	JKLMNOPQRSTUVWX	TUVWXYZ012	6
2445	2445	QRSTUVWXYZ01234	6789	1
2446	2446	3456789	FGHIJKLMNO	2
2447	2447	GHIJKLMNOPQRSTU	IJKLMNOPQR	5
2448	2448	6789	0123456789	8
2449	2449	LMNOPQRSTUVWXYZ	89	8
2450	2450	56789	BCDEFGHIJK	6
2451	2451	456789	OPQRSTUVWX	5
2452	2452	WXYZ0123456789	BCDEFGHIJK	8
2453	2453	FGHIJKLMNOPQRST	FGHIJKLMNO	1
2454	2454	456789	EFGHIJKLMN	9
2455	2455	UVWXYZ012345678	EFGHIJKLMN	11
2456	2456	CDEFGHIJKLMNOPQ	TUVWXYZ012	10
2457	2457	JKLMNOPQRSTUVWX	BCDEFGHIJK	2
2458	2458	QRSTUVWXYZ01234	VWXYZ01234	3
2459	2459	CDEFGHIJKLMNOPQ	6789	4
2460	2460	CDEFGHIJKLMNOPQ	CDEFGHIJKL	10

	id [PK] integer	name character varying (30)	credits integer
524	375254	WXYZ0123456789	43
525	375255	VWXYZ0123456789	2
526	375256	6789	20
527	375257	Z0123456789	50
528	375258	YZ0123456789	40
529	375259	ABCDEFGHIJKLMNO	53
530	375260	STUVWXYZ0123456	28
531	375261	YZ0123456789	14
532	375262	EFGHIJKLMNOPQRS	21
533	375263	3456789	5
534	375264	DEFGHIJKLMNOPQR	50
535	375265	123456789	37
536	375266	89	34
537	375267	FGHIJKLMNOPQRST	53
538	375268	DEFGHIJKLMNOPQR	23
539	375269	KLMNOPQRSTUVWXY	46
540	375270	123456789	39

Вимоги до пункту №3 деталізованого завдання:

```
33Введите таблицу:
students
Введите данные:
6789 UVWXYZ0123 10
Время выполнения запроса: 0.015794992446899414
| 1 | 6789 | UVWXYZ0123 | 10 |
| 17456 | 6789 | UVWXYZ0123 | 10 |
| 22660 | 6789 | UVWXYZ0123 | 10 |
| 31002 | 6789 | UVWXYZ0123 | 10 |
| 34493 | 6789 | UVWXYZ0123 | 10 |
| 37829 | 6789 | UVWXYZ0123 | 10 |
| 54769 | 6789 | UVWXYZ0123 | 10 |
| 66374 | 6789 | UVWXYZ0123 | 10 |
| 83760 | 6789 | UVWXYZ0123 | 10 |
```

Вимоги до пункту №4 деталізованого завдання:

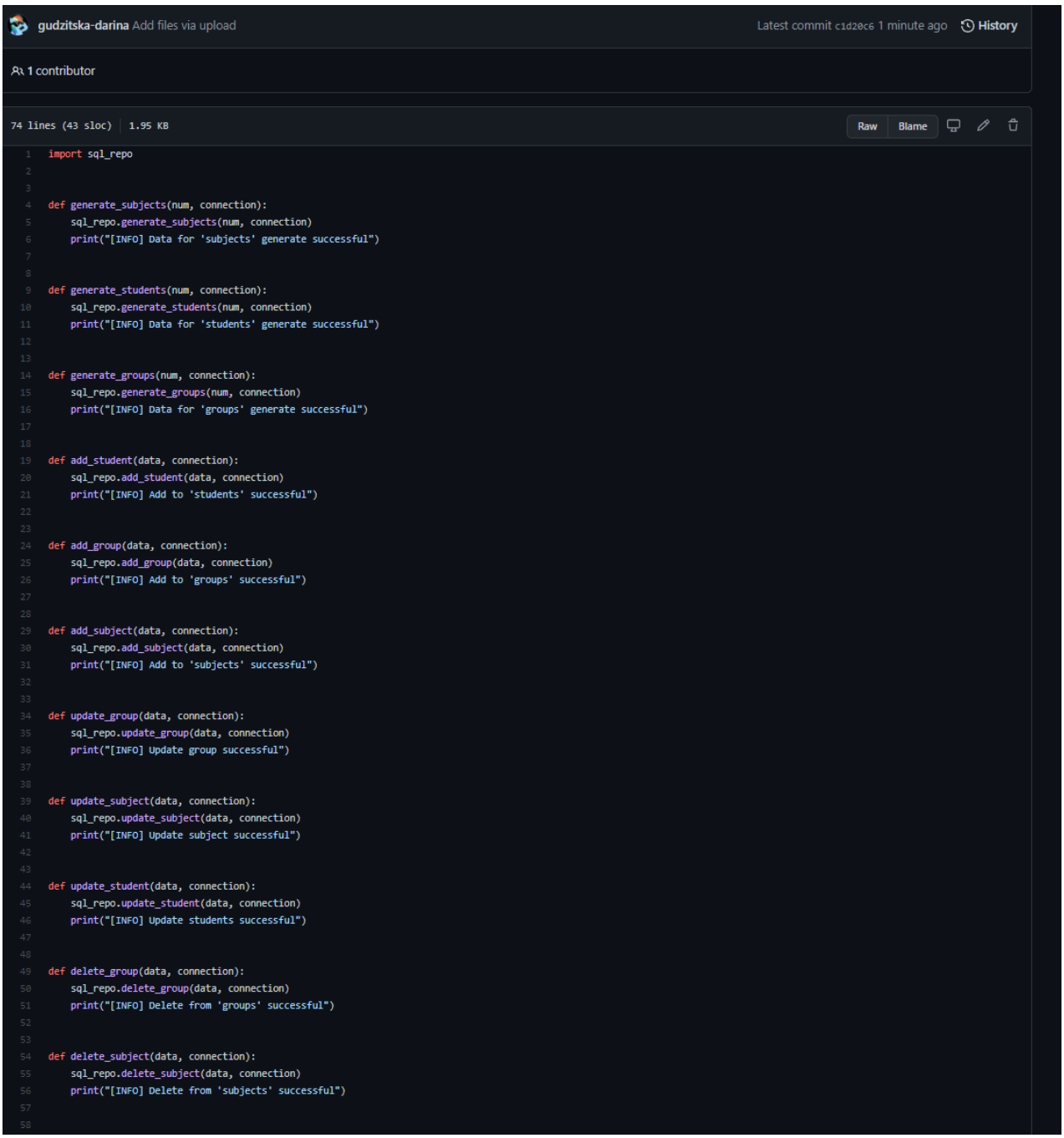


gudzitska-darina Add files via upload — Latest commit c1d20c6 12 seconds ago — History

1 contributor

248 lines (208 sloc) | 7.5 KB — Raw | Blame

```python
import random


def generate_subjects(num, connection):
    i = 0
    with connection.cursor() as cursor:
        while i <= num:
            cursor.execute(
                """INSERT INTO subjects (name, credits) values(
                    substr('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',((random()*(36-1)+1)::integer),15),
                    trunc(random()*60)::int)"""
            )
            i += 1


def generate_groups(num, connection):
    i = 0
    with connection.cursor() as cursor:
        while i <= num:
            cursor.execute(
                """Insert INTO groups (name, faculty) values(
                chr(trunc(65+random()*25)::int) || chr(trunc(65+random()*25)::int) || trunc(random()*30)::int,
                chr(trunc(65+random()*25)::int) || chr(trunc(65+random()*25)::int) || chr(trunc(65+random()*25)::int),
                trunc(random()*980)::int)
                )"""
            )
            i += 1


def generate_students(num, connection):
    i = 0
    min = __getfirstid(connection)
    max = __getcount(connection)
    with connection.cursor() as cursor:
        while i <= num:
            cursor.execute(
                """Insert INTO students (firstname, lastname, group_id) values(
                substr('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',((random()*(36-1)+1)::integer),15),
                substr('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789',((random()*(36-1)+1)::integer),10),
                %s)""" % random.randint(min, max)
            )
            i += 1


def __getfirstid(connection):
    with connection.cursor() as cursor:
        cursor.execute(
            """SELECT id FROM groups
            ORDER BY id ASC
            LIMIT 1"""
        )
        c = cursor.fetchall()
        for i in c:
            min = i[0]
    return min

def __getcount(connection):
    with connection.cursor() as cursor:
```



gudzitska-darina Add files via upload

Latest commit c1d20c6 1 minute ago History

1 contributor

74 lines (43 sloc) | 1.95 KB

Raw Blame

```python
import sql_repo


def generate_subjects(num, connection):
    sql_repo.generate_subjects(num, connection)
    print("[INFO] Data for 'subjects' generate successful")


def generate_students(num, connection):
    sql_repo.generate_students(num, connection)
    print("[INFO] Data for 'students' generate successful")


def generate_groups(num, connection):
    sql_repo.generate_groups(num, connection)
    print("[INFO] Data for 'groups' generate successful")


def add_student(data, connection):
    sql_repo.add_student(data, connection)
    print("[INFO] Add to 'students' successful")


def add_group(data, connection):
    sql_repo.add_group(data, connection)
    print("[INFO] Add to 'groups' successful")


def add_subject(data, connection):
    sql_repo.add_subject(data, connection)
    print("[INFO] Add to 'subjects' successful")


def update_group(data, connection):
    sql_repo.update_group(data, connection)
    print("[INFO] Update group successful")


def update_subject(data, connection):
    sql_repo.update_subject(data, connection)
    print("[INFO] Update subject successful")


def update_student(data, connection):
    sql_repo.update_student(data, connection)
    print("[INFO] Update students successful")


def delete_group(data, connection):
    sql_repo.delete_group(data, connection)
    print("[INFO] Delete from 'groups' successful")


def delete_subject(data, connection):
    sql_repo.delete_subject(data, connection)
    print("[INFO] Delete from 'subjects' successful")


```