



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-03

Серьодкіна Анна Євгенівна
(прізвище, ім'я, по батькові)

Київ 2021

Мета роботи

Метою роботи є здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.

Завдання

Загальне завдання роботи полягає у наступному:

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).

Деталізоване завдання:

1. Забезпечити можливість введення/редагування/вилучення даних у таблицях бази даних з можливістю контролю відповідності типів даних атрибутів таблиць (рядків, чисел, дати/часу). Для контролю пропонується два варіанти: контроль при введенні (валідація даних) та перехоплення помилок (try..except) від сервера PostgreSQL при виконанні відповідної команди SQL. Особливу увагу варто звернути на дані таблиць, що мають зв'язок 1:N. При цьому з боку батьківської таблиці необхідно контролювати вилучення рядків за умови наявності даних у підлеглий таблиці. З точки зору підлеглої таблиці варто контролювати наявність відповідного рядка у батьківській таблиці при виконанні внесення нових даних. Унеможливити виведення програмою системних помилок на екрані шляхом їх перехоплення і адекватної обробки. Внесення даних виконується користувачем у консольному вікні програми.

2. Забезпечити можливість автоматичної генерації великої кількості даних у таблицях за допомогою вбудованих у PostgreSQL функцій роботи з псевдовипадковими числами. Дані мають бути згенерованими не мовою програмування, а відповідним SQL-запитом!

Кількість даних для генерування має вводити користувач з клавіатури.

Для тесту взяти 100 000 записів для однієї-двох таблиць.

Особливу увагу слід звернути на відповідність даних вимогам зовнішніх ключів з метою уникнення помилок порушення обмежень цілісності (foreign key).

3. Для реалізації пошуку необхідно підготувати 3 запити, що включають

дані з декількох таблиць і фільтрують рядки за 3-4 атрибутами цих таблиць. Забезпечити можливість введення конкретних значень констант для фільтрації з клавіатури користувачем. Крім того, після виведення даних необхідно вивести час виконання запиту у мілісекундах.

Перевірити швидкодію роботи запитів на попередньо згенерованих даних.

4. Програмний код організувати згідно шаблону Model-View-Controller(MVC). Приклад організації коду згідно шаблону доступний за даним посиланням . При цьому модель, подання та контролер мають бути реалізовані у окремих файлах. Для доступу до бази даних використовувати лише мову SQL (без ORM).

Результати роботи

Завдання 1

Обробка виняткових ситуацій (помилки) при введенні/вилученні даних;

```
if model.faculty_exists(number):  
    model.delete_faculty(number)  
else:  
    raise EntityException(">> sorry, faculty does not exist.")
```




```
if model.faculty_exists(number):  
    raise EntityException(">> sorry, faculty with this number already  
exists.")  
else:  
    model.add_faculty(name, number)
```

Валідація даних при введенні користувачем:

```
try:  
    st_id = int(input("student id: "))  
    first_name = input("first name: ")  
    last_name = input("last name: ")  
except:  
    raise InputException(">> incorrect input.")
```

Завдання 2

Фрагменти згенерованих даних таблиць:

	 name text	 number [PK] integer 
1	philosophy	13
2	39cb15c2	30
3	d401fefe	86
4	PHN	99
5	ITL	104
6	320ff8a6741db706a1dc1c3069ad3e95	111
7	04d37e21	116

	id [PK] integer	name text	department text
1	1	name3	10
2	2	super teacher	department
3	13	name1	1
4	507	VAU	362c8b94
5	529	NKO	970e3e9c
6	712	SSJ	d7bd0c13
7	996	BIO	5694b87d

Завдання 3

Введення пошукового запиту та результати виконання запитів:

Запит 1:

```
"select * from subjects cross join teachers where subjects.teacher_id = teachers.id and department > (%s)", (str(n),)
```

Результат:

n=1

	name text	lecture_hall_number text	time text	faculty_number integer	teacher_id integer	id integer	name text	department text
1	matan	1212	14-25	121	1	1	name3	10
2	name	1	time	121	1	1	name3	10

Запит 2:

```
"select * from subjects cross join faculties where subjects.faculty_number = faculties.number and lecture_hall_number = (%s)", (str(n),)
```

Результат:

n=1

	name text	lecture_hall_number text	time text	faculty_number integer	teacher_id integer	name text	number integer
1	name	1	time	121	1	math	121

Запит 3:

```
"select subjects.name, lecture_hall_number, time, teachers.name, department from subjects cross join teachers where subjects.teacher_id = teachers.id"
```

Результат:

	name text	lecture_hall_number text	time text	name text	department text
1	proga	44	12-00	name1	1
2	matan	1212	14-25	name3	10
3	name	1	time	name3	10

