



**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:
Мричко Богдан Тарасович
ФПМ, КП-83
варіант 15**

КИЇВ 2020

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).

Результати виконання програми

```
[1] View data
[2] Generate random data
[3] Delete data in range
[4] Enter custom query
[5] Insert data
[0] To quite

Choose your next action: █
```

Рис. 1, Головне меню

```
[1] Cars
[2] Cases
[3] Departments
[4] Employee-Car
[5] Employees
[6] Offenders
[0] To quite

Choose table: █
```

Рис. 2, Вибір таблиці для перегляду даних

Fill up basic query "SELECT * FROM public.cars " or type Enter to skip:

--- cars ---

[1]

car_id: 2
licence_plate: w3h3qq3
vin_code: dthstvw3g54h334
department_id: 1

[2]

car_id: 3
licence_plate: rth34tb4
vin_code: g34qg434g43g
department_id: 2

[3]

car_id: 4
licence_plate: qwd12e1
vin_code: 324g234g234
department_id: 3

[4]

car_id: 5
licence_plate: fwef241f
vin_code: gwrgregerg32423g2
department_id: 3

Рис. 3, Перелік записів таблиці


```
{
  car_id: '2',
  licence_plate: 'w3h3qq3',
  vin_code: 'dthstvw3g54h334',
  department_id: '1'
}

Press [0] to go back or [1] to manage this record: 1
[0] To quite
[1] Delete record
[2] Update car_id
[3] Update licence_plate
[4] Update vin_code
[5] Update department_id
Choose your next action: █
```

Рис. 4, Меню взаємодії з записом

```
Enter new value for licence_plate : newLicenceplate
text
UPDATE public.cars SET licence_plate = 'newLicenceplate' WHERE car_id = 2

Success!
```

Рис. 5, Процес зміни даних запису

```
{
  car_id: '2',
  licence_plate: 'newLicenceplate',
  vin_code: 'dthstvw3g54h334',
  department_id: '1'
}

Press [0] to go back or [1] to manage this record: █
```

Рис. 6, Нові дані запису


```
Enter new value for car_id : 1
num
UPDATE public.cars SET car_id = 1 WHERE car_id = 2
повторювані значення ключа порушують обмеження унікальності "Cars_pkey"
```

```
Enter new value for department_id : 7
num
UPDATE public.cars SET department_id = 7 WHERE car_id = 2
insert або update в таблиці "cars" порушує обмеження зовнішнього ключа "department_id"
```

Рис. 7(1, 2), Повідомлення у випадку введення некоректних даних

```
{
  car_id: '469',
  licence_plate: '88dfd094f4',
  vin_code: 'ce73ec5216767f9',
  department_id: '3'
}

Press [0] to go back or [1] to manage this record: 1
[0] To quite
[1] Delete record
[2] Update car_id
[3] Update licence_plate
[4] Update vin_code
[5] Update department_id
Choose your next action: 1

Success!
```

Рис. 8, Процес видалення даних


```
Choose table: 1
cars
Enter value for car_id column:

888
Enter value for licence_plate column:

qwertyqwerty
Enter value for vin_code column:

samplevincode
Enter value for department_id column:

1

Success!
```

Рис. 9, Створення запису з коректними данимиб повідомлення про успішний запис

```
Choose table: 1
cars
Enter value for car_id column:

2
Enter value for licence_plate column:

qwertyqwerty
Enter value for vin_code column:

samplevincode
Enter value for department_id column:

1
повторювані значення ключа порушують обмеження унікальності "Cars_pkey"
```

Рис. 10, Створення запису з некоректних даних, повідомлення про помилку


```

--- Executet query ---

CREATE OR REPLACE FUNCTION randomCars()
  RETURNS void AS $$
DECLARE
  step integer := 0;
  id integer := random() * (500 - 1) + 1;
BEGIN
  LOOP EXIT WHEN step > 10;
    INSERT INTO public.cars (car_id, licence_plate, vin_code, department_id)
    VALUES (
      (id + step)::integer,
      substring(md5(random()::text), 1, 10),
      substring(md5(random()::text), 1, 10),
      (random() * (3 - 1) + 1)::integer
    );
    step := step + 1;
  END LOOP ;
END;
$$ LANGUAGE PLPGSQL;
SELECT randomCars();

Success!

```

Рис. 11, Виконання запиту на створення 10 випадкових записів

(Коментар: було створено 10 запитів щоб показати, що sql запит коректний, для створення більшої кількості даних достатньо змінити число ітерацій циклу створення)

```

DELETE FROM public.cars WHERE car_id BETWEEN 400 AND
Write ending range: 500
DELETE FROM public.cars WHERE car_id BETWEEN 400 AND 500;

Success!

```

Рис. 12, Видалення даних в певному діапазоні

Підсумок

Лабораторна робота виконана мовою TypeScript

Код програми розміщений на GitHub за покликанням:

<https://github.com/Bodichelly/DataBaseTermFive>

