НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ
Кафедра системного програмування та спеціалізованих комп'ютерних систем

Лабораторна робота №2

з дисципліни

«Бази даних і засоби управління»

На тему «Створення додатку бази даних, орієнтованого на взаємодію з СУБД PostgreSQL»

Виконав: студент III курсу
ФПМ групи КВ-82
Гришко Валерій Валерійович
Перевірив: Павловський В. І.

Київ – 2020

Мета роботи: здобуття практичних навичок проектування та побудови реляційних баз даних та створення прикладних програм з базами даних

Завдання роботи полягає у наступному:

1. Реалізувати функціональні вимоги, наведені нижче.

Функціональні вимоги:

1. Реалізувати внесення, редагування та вилучення даних у базі засобами консольного інтерфейсу;
2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі;
3. Забезпечити реалізацію пошуку за двома-трьома атрибутами з двох сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як перелічення, для логічного типу – значення True/False, для дат – у рамках діапазону дат;

Додаткові вимоги:

1. Передбачити перехоплення помилок. Унеможливити виведення програмою системних помилок на екрані шляхом їх перехоплення і адекватної обробки;
2. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

Вимоги до інтерфейсу користувача:

1. Використовувати консольний інтерфейс користувача.

Логічна модель даних БД «Порушення ПДР»

На рисунку 1 зображена логічна модель даних БД «Порушення ПДР»

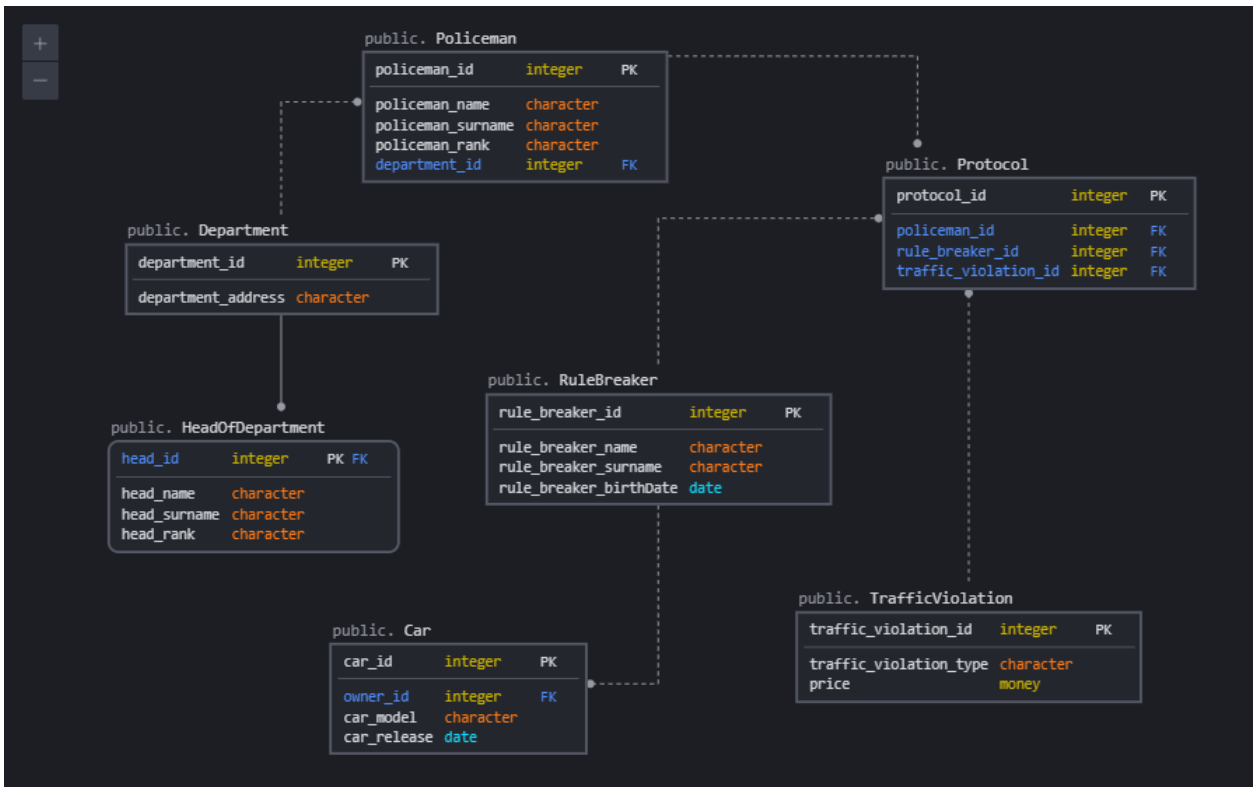

Рисунок 1 – Логічна модель (структура) БД «Порушення ПДР»
Нотація: Модель побудована засобами програми SqlDBM.

Опис програми

Програма створена за допомогою мови програмування Python в середовищі розробки PyCharm Community Edition 2020.2.3

Програма створена за патерном MVC (Model-View-Controller). Складається відповідно з класів Model, View та Controller. Програма створена для управління базою даних за допомогою базових операцій СУБД PostgreSQL та реалізовує функціональні вимоги, що наведені у завданні. Вона складається з 4 модулів:

1. main.py – точка входу до програми, викликає функцію головного меню із Controller.py;
2. Model.py – модуль Model, який містить методи для управління даними програми та БД;
3. View.py – модуль View, який містить методи для відображення результатів роботи Model на екран;
4. Controller.py – модуль Controller, який містить методи для контролю даних введених користувачем та контролю викликів методів Model.

Опис структури меню програми

Меню програми можна розглядати як її концептуальну модель.

На рисунку 2 зображена концептуальна модель програми

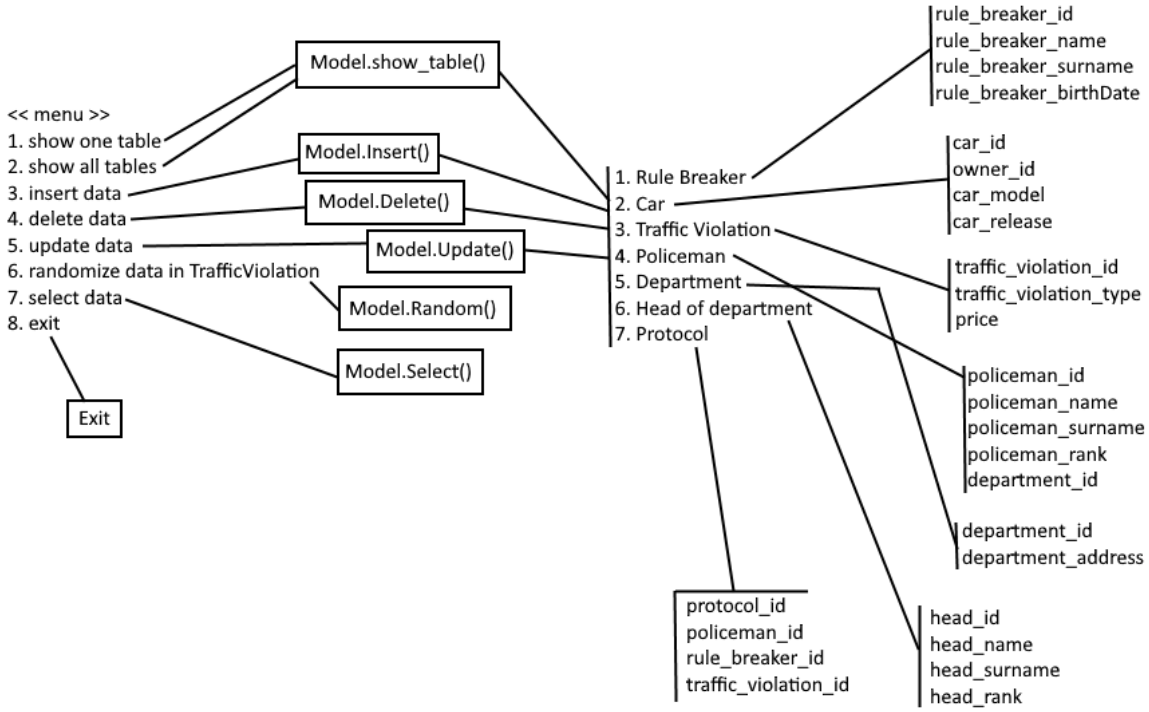

Рисунок 2 – Концептуальна модель програми

Нотація: Модель побудована засобами програми paint.net

Посилання для навігації

Лістинг модуля main.py

```
from Controller import Controller

Controller.my_menu()
```

Лістинг модуля Controller.py

```
import psycopg2
from Model import Model

class Controller:
    @staticmethod
    def my_menu():
        flag = 0
        while flag == 0:
            print("<< menu >>")
            print("1. show one table")
            print("2. show all tables")
            print("3. insert data")
            print("4. delete data")
            print("5. update data")
            print("6. randomize data in TrafficViolation")
            print("7. select data")
            print("8. exit")
            number = int(input('\nMake your number: '))
            if number == 1 or number == 2:
                Model.show_table(number)
            elif number == 3:
                try:
                    Model.Insert()
                except Exception:
                    print("\n... Key Error  ... Please try again ...\n")
            elif number == 4:
                try:
                    Model.Delete()
                except Exception:
                    print("\n... Key Error ... Please try again ...\n")
            elif number == 5:
                try:
                    Model.Update()
                except Exception:
                    print("\n... Key Error ... Please try again ...\n")
            elif number == 6:
                try:
                    Model.Random()
                except Exception:
                    print("\n... Key Error ... Please try again ...\n")
            elif number == 7:
                try:
                    Model.Select()
                except Exception:
                    print("\n... Key Error ... Please try again ...\n")
            elif number == 8:
                flag = 1
```

Лістинг модуля View.py

```
class View:
    def __init__(self, table, records):
        self.table = table
        self.records = records

    @staticmethod
    def list():
        print(" 1. Rule Breaker\n", "2. Car\n", "3. Traffic Violation\n",
              "4. Policeman\n", "5. Department\n", "6. Head of department\n",
"7. Protocol\n")
        number = input("\nMake your number: ")
        return int(number)

    def show(self):
        if self.table == 1:
            for row in self.records:
                print("\nrule_breaker_id =", row[0])
                print("rule_breaker_name =", row[1])
                print("rule_breaker_surname =", row[2])
                print("rule_breaker_birthDate =", row[3])
        elif self.table == 2:
            for row in self.records:
                print("\ncar_id =", row[0])
                print("owner_id =", row[1])
                print("car_model =", row[2])
                print("car_release =", row[3])
        elif self.table == 3:
            for row in self.records:
                print("\ntraffic_violation_id =", row[0])
                print("traffic_violation_type =", row[1])
                print("price = ", row[2])
        elif self.table == 4:
            for row in self.records:
                print("\npoliceman_id =", row[0])
                print("policeman_name =", row[1])
                print("policeman_surname =", row[2])
                print("policeman_rank =", row[3])
                print("department_id =", row[4])
        elif self.table == 5:
            for row in self.records:
                print("\ndepartment_id =", row[0])
                print("department_address =", row[1])
        elif self.table == 6:
            for row in self.records:
                print("\nhead_id =", row[0])
                print("head_name =", row[1])
                print("head_surname =", row[2])
                print("head_rank =", row[3])
        elif self.table == 7:
            for row in self.records:
                print("\nprotocol_id =", row[0])
                print("policeman_id =", row[1])
                print("rule_breaker_id =", row[2])
                print("traffic_violation_id =", row[3])

    @staticmethod
    def attribute_list(table):
        if table == 1:
            print(" 1. rule_breaker_id\n", "2. rule_breaker_name\n",
                  "3. rule_breaker_surname\n", "4. rule_breaker_birthDate\n")
        if table == 2:
```

```
            print(" 1. car_id\n", "2. owner_id\n", "3. car_model\n", "4.
car_release\n")
        if table == 3:
            print(" 1. traffic_violation_id\n", "2.
traffic_violation_type\n", "3. price\n")
        if table == 4:
            print(" 1. policeman_id\n", "2. policeman_name\n",
                  "3. policeman_surname\n", "4. policeman_rank\n", "5.
department_id\n")
        if table == 5:
            print(" 1. department_id\n", "2. department_address\n")
        if table == 6:
            print(" 1. head_id\n", "2. head_name\n", "3. head_surname\n", "4.
head_rank\n")
        if table == 7:
            print(" 1. protocol_id\n", "2. policeman_id\n", "3.
rule_breaker_id\n", "4. traffic_violation_id\n")
        number = input('Number of attribute: ')
        return int(number)
```

Лістинг модуля Model.py

```
import psycopg2
import random
import string
from View import View

class Model:
    @staticmethod
    def Insert():
        flag = 0
        connection = psycopg2.connect(host="localhost", port="5432",
database="Penalty for violation of traffic rules",
                                      user="postgres",
password="dfkthfuhbirj")
        cursor = connection.cursor()
        while flag == 0:
            table = View.list()
            if table < 1 or table > 7:
                print("\n...Incorrect input, try again...")
                continue
            elif table == 1:
                id = input("rule_breaker_id = ")
                name = "'" + input("rule_breaker_name = ") + "'"
                surname = "'" + input("rule_breaker_surname = ") + "'"
                birth_date = "'" + input("rule_breaker_birthDate = ") + "'"
                cursor.execute('INSERT INTO public."RuleBreaker"
(rule_breaker_id, rule_breaker_name, ' \
                               f'rule_breaker_surname,
"rule_breaker_birthDate") VALUES ({id}, {name}, {surname}, {birth_date});')
                connection.commit()
                cursor.close()
                connection.close()
                flag = 1
            elif table == 2:
                id = input("car_id = ")
                owner_id = input("owner_id = ")
                model = "'" + input("car_model = ") + "'"
                release = "'" + input("car_release = ") + "'"
                cursor.execute('INSERT INTO public."Car" (car_id, owner_id,
car_model, ' \
                               f'car_release) VALUES ({id}, {owner_id},
{model}, {release});')
```

```
                connection.commit()
                cursor.close()
                connection.close()
                flag = 1
            elif table == 3:
                id = input("traffic_violation_id = ")
                type = "'" + input("traffic_violation_type = ") + "'"
                price = input("price = ")
                cursor.execute('INSERT INTO public."TrafficViolation"
(traffic_violation_id, traffic_violation_type, ' \
                               f'price) VALUES ({id}, {type}, {price});')
                connection.commit()
                cursor.close()
                connection.close()
                flag = 1
            elif table == 4:
                id = input("policeman_id = ")
                name = "'" + input("policeman_name = ") + "'"
                surname = "'" + input("policeman_surname = ") + "'"
                rank = "'" + input("policeman_rank = ") + "'"
                dep_id = input("department_id = ")
                cursor.execute('INSERT INTO public."Policeman" (policeman_id,
policeman_name, policeman_surname, ' \
                               f'policeman_rank, department_id) VALUES ({id},
{name}, {surname}, {rank}, {dep_id});')
                connection.commit()
                cursor.close()
                connection.close()
                flag = 1
            elif table == 5:
                id = input("department_id = ")
                address = "'" + input("department_address = ") + "'"
                cursor.execute('INSERT INTO public."Department"
(department_id, ' \
                               f'department_address) VALUES ({id},
{address});')
                connection.commit()
                cursor.close()
                connection.close()
                flag = 1
            elif table == 6:
                id = input("head_id = ")
                name = "'" + input("head_name = ") + "'"
                surname = "'" + input("head_surname = ") + "'"
                rank = "'" + input("head_rank = ") + "'"
                cursor.execute('INSERT INTO public."HeadOfDepartment"
(head_id, head_name, head_surname, ' \
                               f'head_rank) VALUES ({id}, {name}, {surname},
{rank});')
                connection.commit()
                cursor.close()
                connection.close()
                flag = 1
            elif table == 7:
                id = input("protocol_id = ")
                pol_id = input("policeman_id = ")
                rule_br_id = input("rule_breaker_id = ")
                tr_viol_id = input("traffic_violation_id = ")
                cursor.execute('INSERT INTO public."Protocol" (protocol_id,
policeman_id, rule_breaker_id, ' \
                               f'traffic_violation_id) VALUES ({id},
{pol_id}, {rule_br_id}, {tr_viol_id});')
                connection.commit()
                cursor.close()
```

```
                connection.close()
                flag = 1

    @staticmethod
    def Delete():
        flag1 = 0
        flag2 = 0
        connection = psycopg2.connect(host="localhost", port="5432",
database="Penalty for violation of traffic rules",
                                      user="postgres",
password="dfkthfuhbirj")
        cursor = connection.cursor()
        while flag1 == 0:
            table = View.list()
            if table < 1 or table > 7:
                print("\n...Incorrect input, try again...")
                continue
            elif table == 1:
                while flag2 == 0:
                    attribute = View.attribute_list(1)
                    if attribute < 1 or attribute > 4:
                        print("\n...Incorrect input, try again...")
                    elif attribute == 1:
                        value = "'" + input('rule_breaker_id value to delete
= ') + "'"
                        where = f'"rule_breaker_id" = {value}'
                        flag2 = 1
                    elif attribute == 2:
                        value = "'" + input('rule_breaker_name value to
delete = ') + "'"
                        where = f'"rule_breaker_name" = {value}'
                        flag2 = 1
                    elif attribute == 3:
                        value = "'" + input('rule_breaker_surname value to
delete = ') + "'"
                        where = f'"rule_breaker_surname" = {value}'
                        flag2 = 1
                    elif attribute == 4:
                        value = "'" + input('rule_breaker_birthDate value to
delete = ') + "'"
                        where = f'"rule_breaker_birthDate" = {value}'
                        flag2 = 1
                cursor.execute(f'DELETE FROM public."RuleBreaker" WHERE
{where}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 2:
                while flag2 == 0:
                    attribute = View.attribute_list(2)
                    if attribute < 1 or attribute > 4:
                        print("\n...Incorrect input, try again...")
                    elif attribute == 1:
                        value = "'" + input('car_id value to delete = ') +
"'"
                        where = f'"car_id" = {value}'
                        flag2 = 1
                    elif attribute == 2:
                        value = "'" + input('owner_id value to delete = ') +
"'"
                        where = f'"owner_id" = {value}'
                        flag2 = 1
                    elif attribute == 3:
```

```
                        value = "'" + input('car_model value to delete = ') +
"'"
                        where = f'"car_model" = {value}'
                        flag2 = 1
                    elif attribute == 4:
                        value = "'" + input('rule_breaker_birthDate value to
delete = ') + "'"
                        where = f'"car_model" = {value}'
                        flag2 = 1
                cursor.execute(f'DELETE FROM public."Car" WHERE {where}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 3:
                while flag2 == 0:
                    attribute = View.attribute_list(3)
                    if attribute < 1 or attribute > 3:
                        print("\n...Incorrect input, try again...")
                    elif attribute == 1:
                        value = "'" + input('traffic_violation_id value to
delete = ') + "'"
                        where = f'"traffic_violation_id" = {value}'
                        flag2 = 1
                    elif attribute == 2:
                        value = "'" + input('traffic_violation_type value to
delete = ') + "'"
                        where = f'"traffic_violation_type" = {value}'
                        flag2 = 1
                    elif attribute == 3:
                        value = "'" + input('price value to delete = ') + "'"
                        where = f'"price" = {value}'
                        flag2 = 1
                cursor.execute(f'DELETE FROM public."TrafficViolation" WHERE
{where}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 4:
                while flag2 == 0:
                    attribute = View.attribute_list(4)
                    if attribute < 1 or attribute > 5:
                        print("\n...Incorrect input, try again...")
                    elif attribute == 1:
                        value = "'" + input('policeman_id value to delete =
') + "'"
                        where = f'"policeman_id" = {value}'
                        flag2 = 1
                    elif attribute == 2:
                        value = "'" + input('policeman_name value to delete =
') + "'"
                        where = f'"policeman_name" = {value}'
                        flag2 = 1
                    elif attribute == 3:
                        value = "'" + input('policeman_surname value to
delete = ') + "'"
                        where = f'"policeman_surname" = {value}'
                        flag2 = 1
                    elif attribute == 4:
                        value = "'" + input('policeman_rank value to delete =
') + "'"
                        where = f'"policeman_rank" = {value}'
                        flag2 = 1
```

```
                    elif attribute == 5:
                        value = "'" + input('department_id value to delete =
') + "'"
                        where = f'"department_id" = {value}'
                        flag2 = 1
                cursor.execute(f'DELETE FROM public."Policeman" WHERE
{where}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 5:
                while flag2 == 0:
                    attribute = View.attribute_list(5)
                    if attribute < 1 or attribute > 2:
                        print("\n...Incorrect input, try again...")
                    elif attribute == 1:
                        value = "'" + input('department_id value to delete =
') + "'"
                        where = f'"department_id" = {value}'
                        flag2 = 1
                    elif attribute == 2:
                        value = "'" + input('department_address value to
delete = ') + "'"
                        where = f'"department_address" = {value}'
                        flag2 = 1
                cursor.execute(f'DELETE FROM public."Department" WHERE
{where}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 6:
                while flag2 == 0:
                    attribute = View.attribute_list(6)
                    if attribute < 1 or attribute > 4:
                        print("\n...Incorrect input, try again...")
                    elif attribute == 1:
                        value = "'" + input('head_id value to delete = ') +
"'"
                        where = f'"head_id" = {value}'
                        flag2 = 1
                    elif attribute == 2:
                        value = "'" + input('head_name value to delete = ') +
"'"
                        where = f'"head_name" = {value}'
                        flag2 = 1
                    elif attribute == 3:
                        value = "'" + input('head_surname value to delete =
') + "'"
                        where = f'"head_surname" = {value}'
                        flag2 = 1
                    elif attribute == 4:
                        value = "'" + input('head_rank value to delete = ') +
"'"
                        where = f'"head_rank" = {value}'
                        flag2 = 1
                cursor.execute(f'DELETE FROM public."HeadOfDepartment" WHERE
{where}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 7:
```

```
                while flag2 == 0:
                    attribute = View.attribute_list(7)
                    if attribute < 1 or attribute > 4:
                        print("\n...Incorrect input, try again...")
                    elif attribute == 1:
                        value = "'" + input('protocol_id value to delete = ')
+ "'"
                        where = f'"protocol_id" = {value}'
                        flag2 = 1
                    elif attribute == 2:
                        value = "'" + input('policeman_id value to delete =
') + "'"
                        where = f'"policeman_id" = {value}'
                        flag2 = 1
                    elif attribute == 3:
                        value = "'" + input('rule_breaker_id value to delete
= ') + "'"
                        where = f'"rule_breaker_id" = {value}'
                        flag2 = 1
                    elif attribute == 4:
                        value = "'" + input('traffic_violation_id value to
delete = ') + "'"
                        where = f'"traffic_violation_id" = {value}'
                        flag2 = 1
                cursor.execute(f'DELETE FROM public."Protocol" WHERE
{where}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1

    @staticmethod
    def Update():
        flag1 = 0
        flag2 = 0
        flag3 = 0
        connection = psycopg2.connect(host="localhost", port="5432",
database="Penalty for violation of traffic rules",
                                      user="postgres",
password="dfkthfuhbirj")
        cursor = connection.cursor()
        while flag1 == 0:
            table = View.list()
            if table < 1 or table > 7:
                print("\n...Incorrect input, try again...")
                continue
            elif table == 1:
                while flag2 == 0:
                    where = View.attribute_list(1)
                    if where < 1 or where > 4:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif where == 1:
                        rule_breaker_id = "'" + input('Attribute to update
where rule_breaker_id = ') + "'"
                        flag2 = 1
                    elif where == 2:
                        rule_breaker_name = "'" + input('Attribute to update
where rule_breaker_name = ') + "'"
                        flag2 = 1
                    elif where == 3:
                        rule_breaker_surname = "'" + input('Attribute to
update where rule_breaker_surname = ') + "'"
                        flag2 = 1
```

```
                    elif where == 4:
                        rule_breaker_birthDate = "'" + input('Attribute to
update where rule_breaker_birthDate = ') + "'"
                        flag2 = 1
                while flag3 == 0:
                    attribute = View.attribute_list(1)
                    if attribute < 1 or attribute > 4:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif attribute == 1:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"rule_breaker_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 2:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"rule_breaker_name" = {new_value}'
                        flag3 = 1
                    elif attribute == 3:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"rule_breaker_surname" = {new_value}'
                        flag3 = 1
                    elif attribute == 4:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"rule_breaker_birthDate" = {new_value}'
                        flag3 = 1
                if where == 1:
                    cursor.execute(
                        f'UPDATE public."RuleBreaker" SET {set} WHERE
"rule_breaker_id" = {rule_breaker_id}')
                elif where == 2:
                    cursor.execute(
                        f'UPDATE public."RuleBreaker" SET {set} WHERE
"rule_breaker_name" = {rule_breaker_name}')
                elif where == 3:
                    cursor.execute(
                        f'UPDATE public."RuleBreaker" SET {set} WHERE
"rule_breaker_surname" = {rule_breaker_surname}')
                elif where == 4:
                    cursor.execute(
                        f'UPDATE public."RuleBreaker" SET {set} WHERE
"rule_breaker_birthDate" = {rule_breaker_birthDate}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 2:
                while flag2 == 0:
                    where = View.attribute_list(2)
                    if where < 1 or where > 4:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif where == 1:
                        car_id = "'" + input('Attribute to update where
car_id = ') + "'"
                        flag2 = 1
                    elif where == 2:
                        owner_id = "'" + input('Attribute to update where
owner_id = ') + "'"
                        flag2 = 1
                    elif where == 3:
```

```
                        car_model = "'" + input('Attribute to update where
car_model = ') + "'"
                        flag2 = 1
                    elif where == 4:
                        car_release = "'" + input(
                            'Attribute to update where car_release = ') + "'"
                        flag2 = 1
                while flag3 == 0:
                    attribute = View.attribute_list(2)
                    if attribute < 1 or attribute > 4:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif attribute == 1:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"car_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 2:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"owner_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 3:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"car_model" = {new_value}'
                        flag3 = 1
                    elif attribute == 4:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"car_release" = {new_value}'
                        flag3 = 1
                if where == 1:
                    cursor.execute(
                        f'UPDATE public."Car" SET {set} WHERE "car_id" =
{car_id}')
                elif where == 2:
                    cursor.execute(
                        f'UPDATE public."Car" SET {set} WHERE "owner_id" =
{owner_id}')
                elif where == 3:
                    cursor.execute(
                        f'UPDATE public."Car" SET {set} WHERE "car_model" =
{car_model}')
                elif where == 4:
                    cursor.execute(
                        f'UPDATE public."Car" SET {set} WHERE "car_release" =
{car_release}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 3:
                while flag2 == 0:
                    where = View.attribute_list(3)
                    if where < 1 or where > 3:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif where == 1:
                        traffic_violation_id = "'" + input('Attribute to
update where traffic_violation_id = ') + "'"
                        flag2 = 1
                    elif where == 2:
                        traffic_violation_type = "'" + input('Attribute to
```

```
update where traffic_violation_type = ') + "'"
                        flag2 = 1
                    elif where == 3:
                        price = "'" + input('Attribute to update where price
= ') + "'"
                        flag2 = 1
                while flag3 == 0:
                    attribute = View.attribute_list(3)
                    if attribute < 1 or attribute > 3:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif attribute == 1:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"traffic_violation_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 2:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"traffic_violation_type" = {new_value}'
                        flag3 = 1
                    elif attribute == 3:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"price" = {new_value}'
                        flag3 = 1
                if where == 1:
                    cursor.execute(
                        f'UPDATE public."TrafficViolation" SET {set} WHERE
"traffic_violation_id" = {traffic_violation_id}')
                elif where == 2:
                    cursor.execute(
                        f'UPDATE public."TrafficViolation" SET {set} WHERE
"traffic_violation_type" = {traffic_violation_type}')
                elif where == 3:
                    cursor.execute(
                        f'UPDATE public."TrafficViolation" SET {set} WHERE
"price" = {price}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 4:
                while flag2 == 0:
                    where = View.attribute_list(4)
                    if where < 1 or where > 5:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif where == 1:
                        policeman_id = "'" + input('Attribute to update where
policeman_id = ') + "'"
                        flag2 = 1
                    elif where == 2:
                        policeman_name = "'" + input('Attribute to update
where policeman_name = ') + "'"
                        flag2 = 1
                    elif where == 3:
                        policeman_surname = "'" + input('Attribute to update
where policeman_surname = ') + "'"
                        flag2 = 1
                    elif where == 4:
                        policeman_rank = "'" + input('Attribute to update
where policeman_rank = ') + "'"
                        flag2 = 1
```

```
                    elif where == 5:
                        department_id = "'" + input('Attribute to update
where department_id = ') + "'"
                        flag2 = 1
                while flag3 == 0:
                    attribute = View.attribute_list(4)
                    if attribute < 1 or attribute > 5:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif attribute == 1:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"policeman_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 2:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"policeman_name" = {new_value}'
                        flag3 = 1
                    elif attribute == 3:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"policeman_surname" = {new_value}'
                        flag3 = 1
                    elif attribute == 4:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"policeman_rank" = {new_value}'
                        flag3 = 1
                    elif attribute == 5:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"department_id" = {new_value}'
                        flag3 = 1
                if where == 1:
                    cursor.execute(
                        f'UPDATE public."Policeman" SET {set} WHERE
"policeman_id" = {policeman_id}')
                elif where == 2:
                    cursor.execute(
                        f'UPDATE public."Policeman" SET {set} WHERE
"policeman_name" = {policeman_name}')
                elif where == 3:
                    cursor.execute(
                        f'UPDATE public."Policeman" SET {set} WHERE
"policeman_surname" = {policeman_surname}')
                elif where == 4:
                    cursor.execute(
                        f'UPDATE public."Policeman" SET {set} WHERE
"policeman_rank" = {policeman_rank}')
                elif where == 5:
                    cursor.execute(
                        f'UPDATE public."Policeman" SET {set} WHERE
"department_id" = {department_id}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 5:
                while flag2 == 0:
                    where = View.attribute_list(5)
                    if where < 1 or where > 2:
                        print("\n...Incorrect input, try again...")
                        continue
```

```
                    elif where == 1:
                        department_id = "'" + input('Attribute to update
where department_id = ') + "'"
                        flag2 = 1
                    elif where == 2:
                        department_address = "'" + input('Attribute to update
where department_address = ') + "'"
                        flag2 = 1
                while flag3 == 0:
                    attribute = View.attribute_list(5)
                    if attribute < 1 or attribute > 2:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif attribute == 1:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"department_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 2:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"department_address" = {new_value}'
                        flag3 = 1
                if where == 1:
                    cursor.execute(
                        f'UPDATE public."Department" SET {set} WHERE
"department_id" = {department_id}')
                elif where == 2:
                    cursor.execute(
                        f'UPDATE public."Department" SET {set} WHERE
"department_address" = {department_address}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 6:
                while flag2 == 0:
                    where = View.attribute_list(6)
                    if where < 1 or where > 4:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif where == 1:
                        head_id = "'" + input('Attribute to update where
head_id = ') + "'"
                        flag2 = 1
                    elif where == 2:
                        head_name = "'" + input('Attribute to update where
head_name = ') + "'"
                        flag2 = 1
                    elif where == 3:
                        head_surname = "'" + input('Attribute to update where
head_surname = ') + "'"
                        flag2 = 1
                    elif where == 4:
                        head_rank = "'" + input('Attribute to update where
head_rank = ') + "'"
                        flag2 = 1
                while flag3 == 0:
                    attribute = View.attribute_list(6)
                    if attribute < 1 or attribute > 4:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif attribute == 1:
                        new_value = "'" + input('New value of attribute = ')
```

```
+ "'"
                        set = f'"head_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 2:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"head_name" = {new_value}'
                        flag3 = 1
                    elif attribute == 3:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"head_surname" = {new_value}'
                        flag3 = 1
                    elif attribute == 4:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"head_rank" = {new_value}'
                        flag3 = 1
                if where == 1:
                    cursor.execute(
                        f'UPDATE public."HeadOfDepartment" SET {set} WHERE
"head_id" = {head_id}')
                elif where == 2:
                    cursor.execute(
                        f'UPDATE public."HeadOfDepartment" SET {set} WHERE
"head_name" = {head_name}')
                elif where == 3:
                    cursor.execute(
                        f'UPDATE public."HeadOfDepartment" SET {set} WHERE
"head_surname" = {head_surname}')
                elif where == 4:
                    cursor.execute(
                        f'UPDATE public."HeadOfDepartment" SET {set} WHERE
"head_rank" = {head_rank}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1
            elif table == 7:
                while flag2 == 0:
                    where = View.attribute_list(7)
                    if where < 1 or where > 4:
                        print("\n...Incorrect input, try again...")
                        continue
                    elif where == 1:
                        protocol_id = "'" + input('Attribute to update where
protocol_id = ') + "'"
                        flag2 = 1
                    elif where == 2:
                        policeman_id = "'" + input('Attribute to update where
policeman_id = ') + "'"
                        flag2 = 1
                    elif where == 3:
                        rule_breaker_id = "'" + input('Attribute to update
where rule_breaker_id = ') + "'"
                        flag2 = 1
                    elif where == 4:
                        traffic_violation_id = "'" + input('Attribute to
update where traffic_violation_id = ') + "'"
                        flag2 = 1
                while flag3 == 0:
                    attribute = View.attribute_list(7)
                    if attribute < 1 or attribute > 4:
                        print("\n...Incorrect input, try again...")
```

```
                        continue
                    elif attribute == 1:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"protocol_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 2:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"policeman_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 3:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"rule_breaker_id" = {new_value}'
                        flag3 = 1
                    elif attribute == 4:
                        new_value = "'" + input('New value of attribute = ')
+ "'"
                        set = f'"traffic_violation_id" = {new_value}'
                        flag3 = 1
                if where == 1:
                    cursor.execute(
                        f'UPDATE public."Protocol" SET {set} WHERE
"protocol_id" = {protocol_id}')
                elif where == 2:
                    cursor.execute(
                        f'UPDATE public."Protocol" SET {set} WHERE
"policeman_id" = {policeman_id}')
                elif where == 3:
                    cursor.execute(
                        f'UPDATE public."Protocol" SET {set} WHERE
"rule_breaker_id" = {rule_breaker_id}')
                elif where == 4:
                    cursor.execute(
                        f'UPDATE public."Protocol" SET {set} WHERE
"traffic_violation_id" = {traffic_violation_id}')
                connection.commit()
                cursor.close()
                connection.close()
                flag1 = 1

    @staticmethod
    def show_table(number):
        flag = 0
        table = 0
        while flag == 0:
            if number == 1:
                table = View.list()
                flag = 1
            elif number == 2:
                table += 1
                if table == 7:
                    flag = 1
            connection = psycopg2.connect(host="localhost", port="5432",
                                          database="Penalty for violation of
traffic rules",
                                          user="postgres",
password="dfkthfuhbirj")
            cursor = connection.cursor()
            if table == 1:
                print("table: RuleBreaker")
                cursor.execute('SELECT * FROM public."RuleBreaker"')
                rows = cursor.fetchall()
```

```
                View(table, rows).show()
                print("\n")
            elif table == 2:
                print("table: Car")
                cursor.execute('SELECT * FROM public."Car"')
                rows = cursor.fetchall()
                View(table, rows).show()
                print("\n")
            elif table == 3:
                print("table: TrafficViolation")
                cursor.execute('SELECT * FROM public."TrafficViolation"')
                rows = cursor.fetchall()
                View(table, rows).show()
                print("\n")
            elif table == 4:
                print("table: Policeman")
                cursor.execute('SELECT * FROM public."Policeman"')
                rows = cursor.fetchall()
                View(table, rows).show()
                print("\n")
            elif table == 5:
                print("table: Department")
                cursor.execute('SELECT * FROM public."Department"')
                rows = cursor.fetchall()
                View(table, rows).show()
                print("\n")
            elif table == 6:
                print("table: HeadOfDepartment")
                cursor.execute('SELECT * FROM public."HeadOfDepartment"')
                rows = cursor.fetchall()
                View(table, rows).show()
                print("\n")
            elif table == 7:
                print("table: Protocol")
                cursor.execute('SELECT * FROM public."Protocol"')
                rows = cursor.fetchall()
                View(table, rows).show()
                print("\n")

    @staticmethod
    def Random():
        flag = 0
        connection = psycopg2.connect(host="localhost", port="5432",
                                      database="Penalty for violation of
traffic rules",
                                      user="postgres",
password="dfkthfuhbirj")
        cursor = connection.cursor()
        while flag == 0:
            counter = int(input('How much data do you need to generate?
Input: '))
            if counter > 1:
                flag = 1
            else:
                print('\n...Incorrect input, try again...\n')
        for i in range(1, counter + 1):
            tr_viol_id = random.randint(20, 100000)
            tr_viol_type = "'" + ''.join(random.choice(string.ascii_letters)
for _ in range(5)) + "'"
            price = random.randint(100, 10000)
            cursor.execute('INSERT INTO public."TrafficViolation" ' \
                    f'(traffic_violation_id, traffic_violation_type, price)
VALUES ({tr_viol_id}, {tr_viol_type}, {price});')
            connection.commit()
```

```
            cursor.close()
            connection.close()

    @staticmethod
    def Select():
        connection = psycopg2.connect(host="localhost", port="5432",
                                      database="Penalty for violation of
traffic rules",
                                      user="postgres",
password="dfkthfuhbirj")
        cursor = connection.cursor()
        border1 = "'" + input("first border: ") + "'"
        border2 = "'" + input("second border: ") + "'"
        select = 'SELECT * FROM public."RuleBreaker" AS r1 ' \
                 'INNER JOIN (SELECT * FROM public."Car") AS c1 ON
r1.rule_breaker_id = c1.owner_id ' \
                 f'WHERE rule_breaker_id BETWEEN {border1} AND {border2}'
        cursor.execute(select)
        records = cursor.fetchall()
        cursor.close()
        connection.close()
        for row in records:
            print("\nrule_breaker_id =", row[0])
            print("rule_breaker_name =", row[1])
            print("rule_breaker_surname =", row[2])
            print("rule_breaker_birthDate =", row[3])
            print("car_id =", row[4])
            print("owner_id =", row[5])
            print("car_model =", row[6])
            print("car_release =", row[7])
```

Результати роботи програми

В даному розділі на рисунках зображені результати роботи програми

```
<< menu >>
1. show one table
2. show all tables
3. insert data
4. delete data
5. update data
6. randomize data in TrafficViolation
7. select data
8. exit

Make your number:
```

Рисунок 3 – Запуск програми. Головне меню

```
 1. Rule Breaker
 2. Car
 3. Traffic Violation
 4. Policeman
 5. Department
 6. Head of department
 7. Protocol


Make your number: 1
rule_breaker_id = 11
rule_breaker_name = Vlad
rule_breaker_surname = Ivanov
rule_breaker_birthDate = 15.12.1995
```

Рисунок 4 – Додавання даних до БД

```
table: RuleBreaker

rule_breaker_id = 1
rule_breaker_name = valera
rule_breaker_surname = grishko
rule_breaker_birthDate = 2000-11-06

rule_breaker_id = 2
rule_breaker_name = kirill
rule_breaker_surname = slusarenko
rule_breaker_birthDate = 2001-04-04

rule_breaker_id = 3
rule_breaker_name = ivan
rule_breaker_surname = ivanov
rule_breaker_birthDate = 1995-10-01

rule_breaker_id = 6
rule_breaker_name = Vlad
rule_breaker_surname = Zubko
rule_breaker_birthDate = 1999-10-08

rule_breaker_id = 11
rule_breaker_name = Vlad
rule_breaker_surname = Ivanov
rule_breaker_birthDate = 1995-12-15
```

Рисунок 5 – Вміст таблиці «RuleBreaker» після додавання нових даних

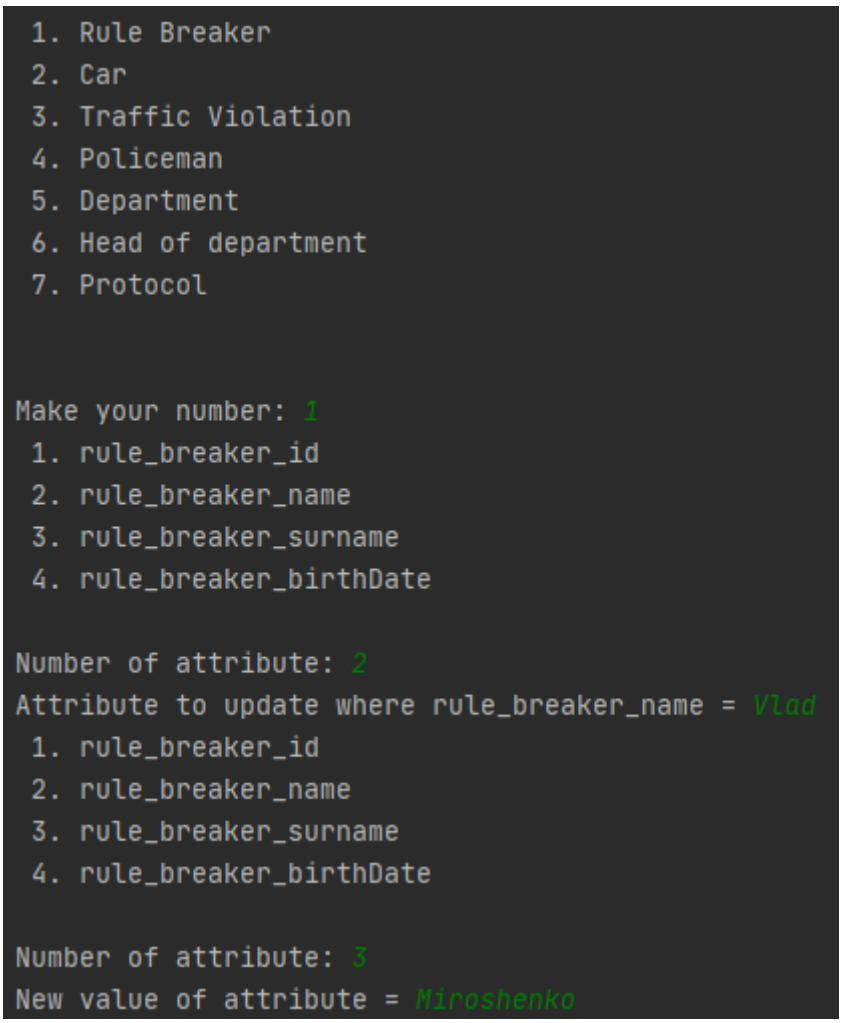

Рисунок 6 – Редагування даних у БД

Нотація: В даному прикладі користувач змінює прізвища усіх Vlad у таблиці.

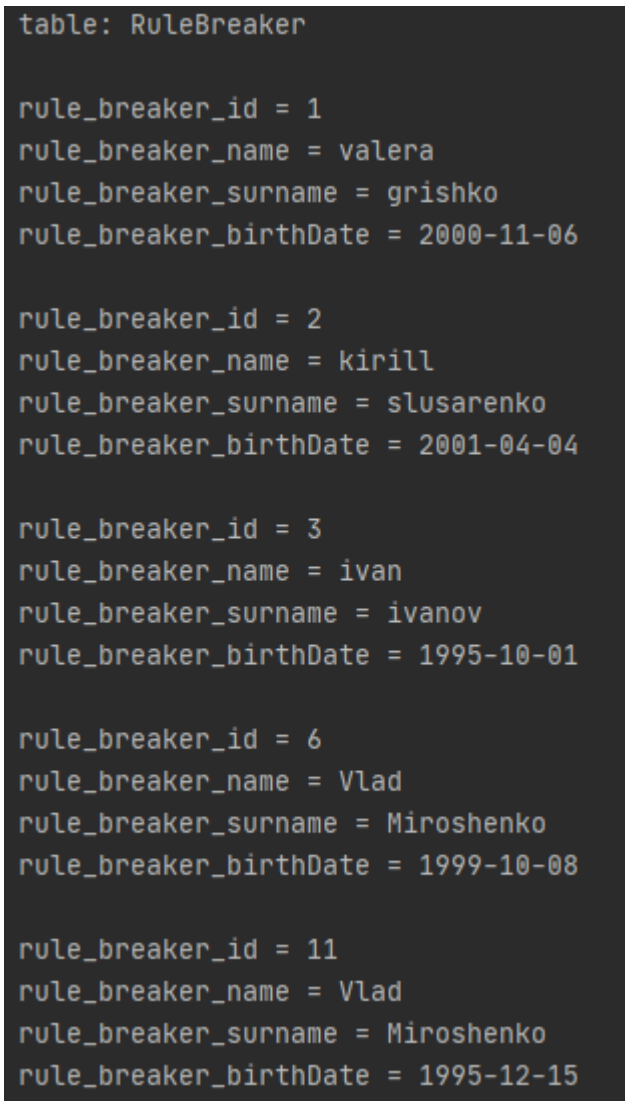

Рисунок 7 - Вміст таблиці «RuleBreaker» після редагування даних

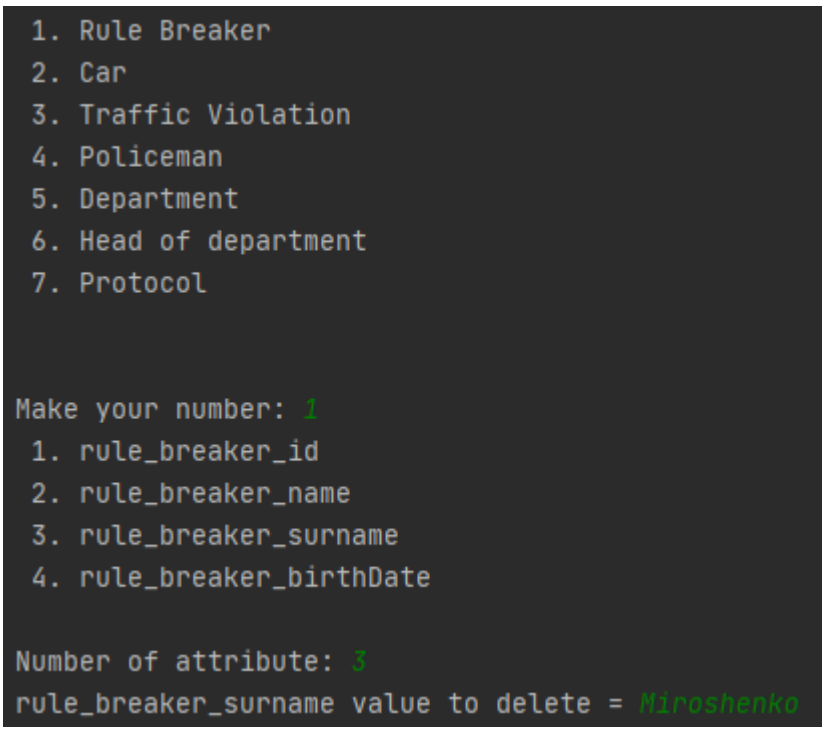

Рисунок 8 – Видалення даних з таблиці

Нотація: В даному прикладі ми видаляємо дані за прізвищем: будуть видалені всі дані порушників з прізвищем Miroshenko. Програмою передбачено видалення за будь-яким атрибутом.

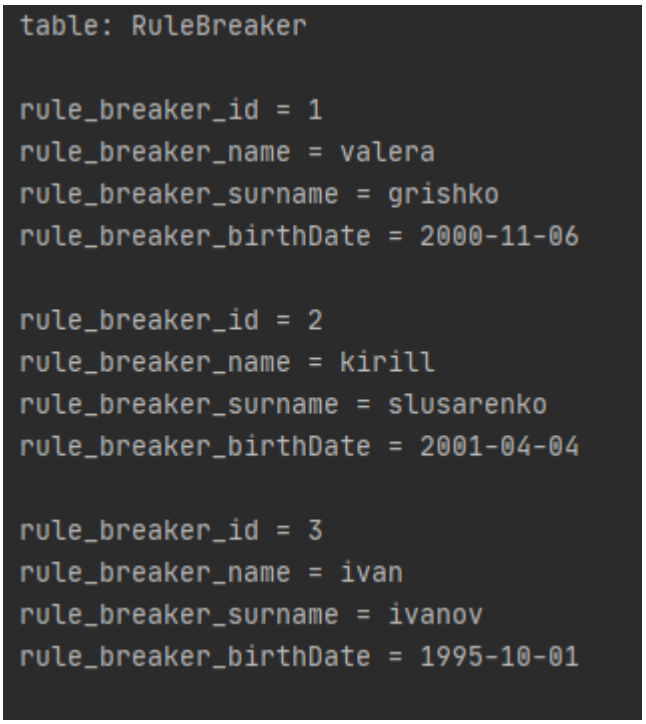

Рисунок 9 – Вміст таблиці «RuleBreaker» після видалення даних

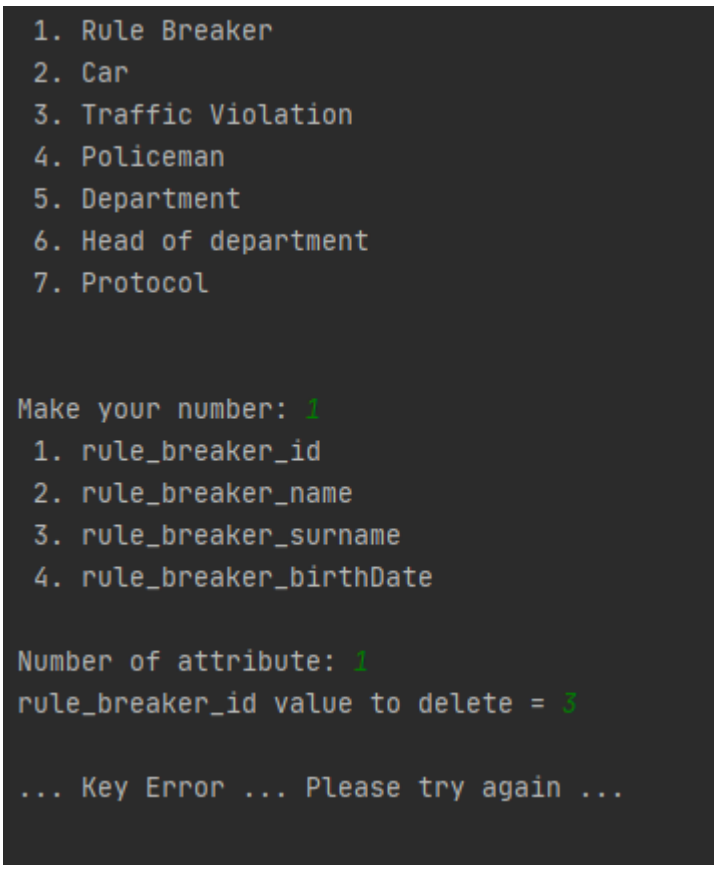

Рисунок 10 – Спроба видалити дані, які пов’язані з даними іншої таблиці зовнішнім ключем

Нотація: Програмою передбачено перехоплення помилок. Якщо є спроба додати дані до таблиці за ключем, який вже існує – програма виведе на екран повідомлення про помилку. Якщо є спроба видалити з таблиці дані, не видаливши при цьому дані у дочірній таблиці, – програма також виведе на екран повідомлення про помилку.

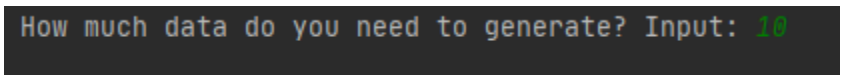

Рисунок 11 – Регенерування випадкових даних у таблиці «TrafficViolation»

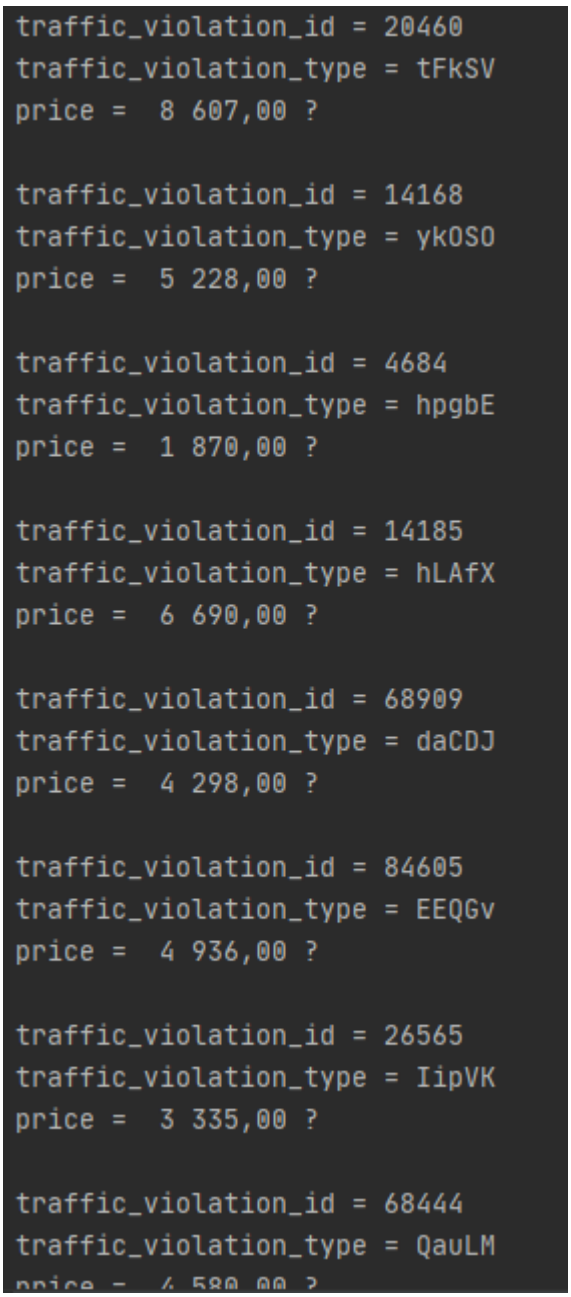

Рисунок 12 – Вміст таблиці «TrafficViolation» після додавання рандомізованих даних

```
first border: 1
second border: 5

rule_breaker_id = 3
rule_breaker_name = ivan
rule_breaker_surname = ivanov
rule_breaker_birthDate = 1995-10-01
car_id = 1
owner_id = 3
car_model = bmw
car_release = 2010-04-20
```

Рисунок 13 – Пошук з декількох сутностей одночасно

Нотація: В даному прикладі користувач з клавіатури вводить діапазон значень унікального id порушника. В результаті програма виведе на екран дані про автомобілі та порушників, яким ці автомобілі належать.