

Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

# Розрахункова графічна робота

***з дисципліни «Бази даних»***

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

Виконав студент групи: КВ-11

ПІБ: Гультяєв Дмитро Антонович

Telegram: @dimagultiaev

GitHub: [Link](https://github.com/dmytro-hultiaiev/kpi_db/tree/main/RGR)

Перевірив: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Київ 2023**

**Постановка задачі:**

1. Реалізувати функції перегляду, 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).

***Структура бази даних:***

**Сутності:**

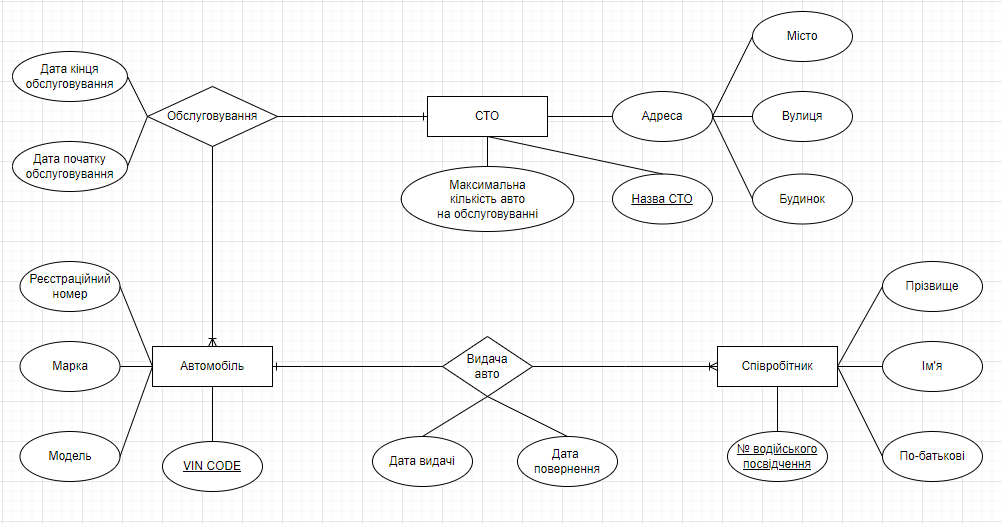
* Автомобіль (Cars) – ця сутність представляє автопарк компанії. У цій сутності ми маємо такі атрибути, як: vin, реєстраційний номер автомобіля, марку та модель.
* Співробітник (Employee) – ця сутність представляє працівників, які мають водійське посвідчення. Ці працівники можуть отримувати автомобілі в розпорядження з автопарку компанії. У цій сутності ми маємо такі атрибути, як: номер водійського посвідчення та ФІО.
* СТО (Service Station) – ця сутність представляє станцію технічного обслуговування, на якій обслуговується автопарк компанії. У цій сутності ми маємо такі атрибути, як: назва автосервісу, адреса та максимальна кількість автомобілей, яку автосервіс може обслуговувати одночасно.

**Зв’язки:**

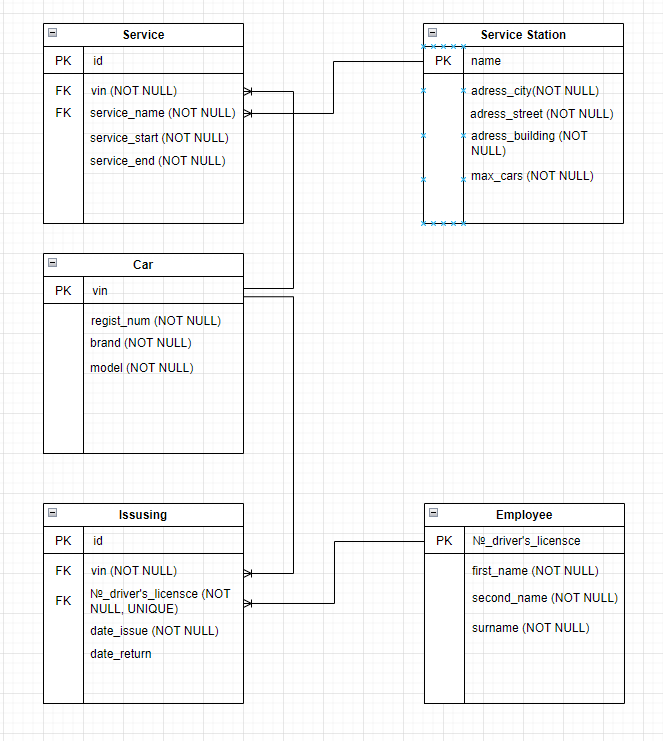
* 1:N – між автомобілем та співробітником (1 автомобіль може з часом переходити до різних співробітників).
* 1:N – між сервісом та автомобілем (1 сервіс обслуговує весь автопарк компанії).

**Також я виділив такі зв’язки з атрибутами, як:**

* Обслуговування (Service) – цей зв’язок поєднує сутність автомобіль та СТО
* Видача авто (Issuing) - цей зв’язок поєднує сутність автомобіль та співробітник



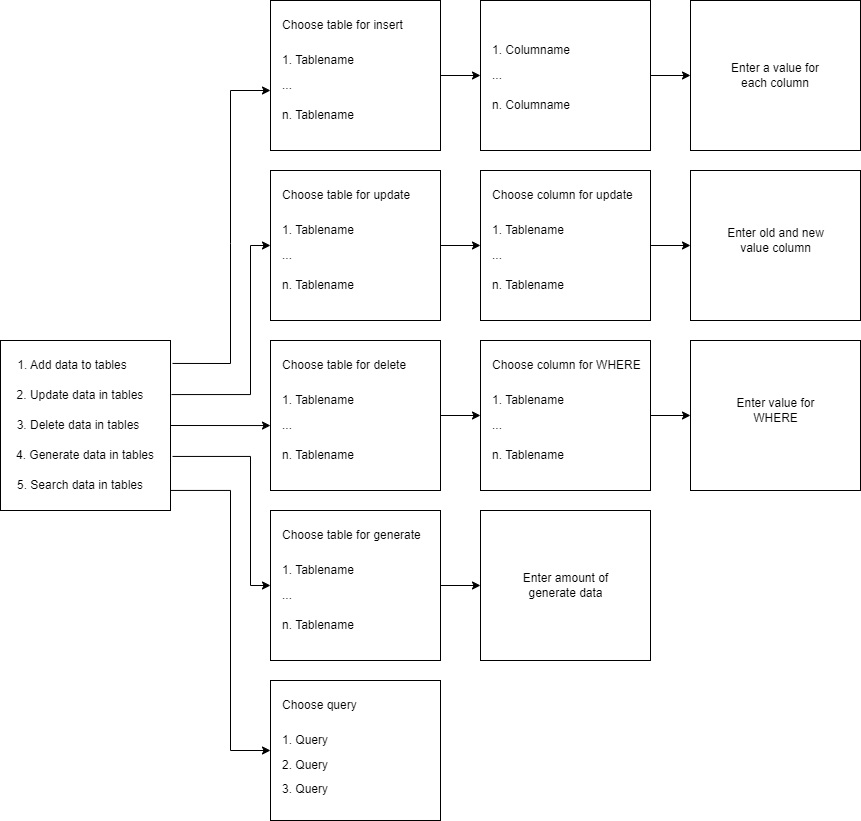
*ER діаграма виконана за нотацією “Пташина лапка”*



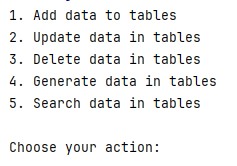
*Структура бази даних*

**Опис меню користувача:**

1. Add data to tables. Цей пункт призначений для додавання даних в таблиці. Після вибору цього пункту, користувач вибирає таблицю для вставки даних, а також вводить необхідні данні
2. Update data in tables. Цей пункт призначений для зміни даних в таблиці. Після вибору цього пункту, користувач вибирає таблицю, в якій треба змінити данні, вибирає потрібний аргумент таблиці для зміни, далі вводить старе та нове значення.
3. Delete data in tables. Цей пункт призначений для видалення даних в таблиці. Після вибору цього пункту, користувач вибирає таблицю, в якій треба видалити данні, вибирає потрібний аргумент таблиці для видалення, далі вводить значення цього елементу.
4. Generate data in tables. Цей пункт призначений для генерації даних в таблиці. Після вибору цього пункту, користувач вибирає таблицю, в якій треба згенерувати данні, а далі вводить кількість, яку необхідно згенерувати.
5. Search data in tables. Цей пункт призначений для реалізації пункту 3 завдання (пошук даних). Після вибору цього пункту, користувач вибирає один із 3 запитів, а далі вводить потрібні фільтри для цього запиту.

******

*Графічна схема меню*

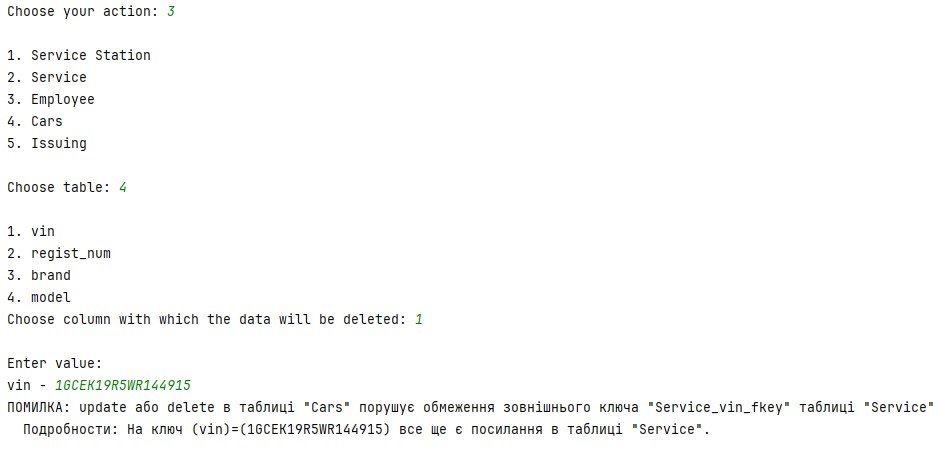


*Головне меню в консолі*

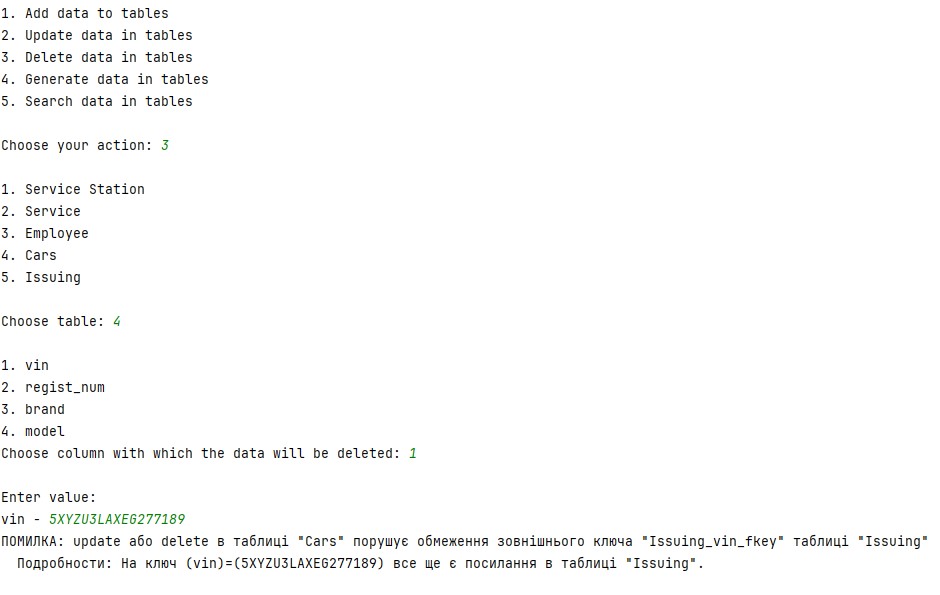
Для створення програми була обрана мова Java. А для посилання запитів до бази даних був вибраний інструмент JDBC

***Пункт №1:***

***Невдале вилучення записів:***

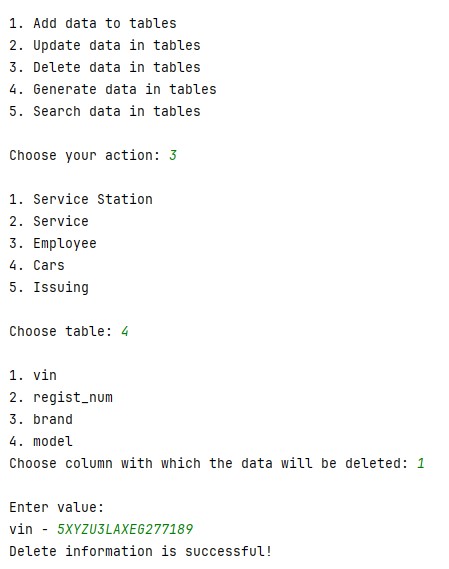


Помилка виникає через те, що ми намагаємося видалити автомобіль з таблиці “Cars”, при тому що він на даний момент обслуговується і знаходиться в таблиці Service

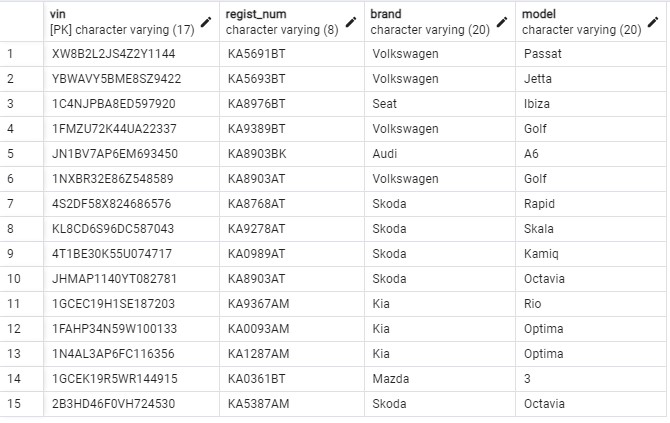
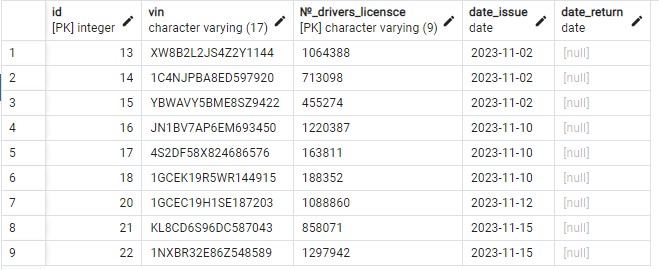


Помилка виникає через те, що ми намагаємося видалити автомобіль з таблиці “Cars”, при тому що він на даний момент знаходиться в користуванні і розміщається в таблиці Issuing.

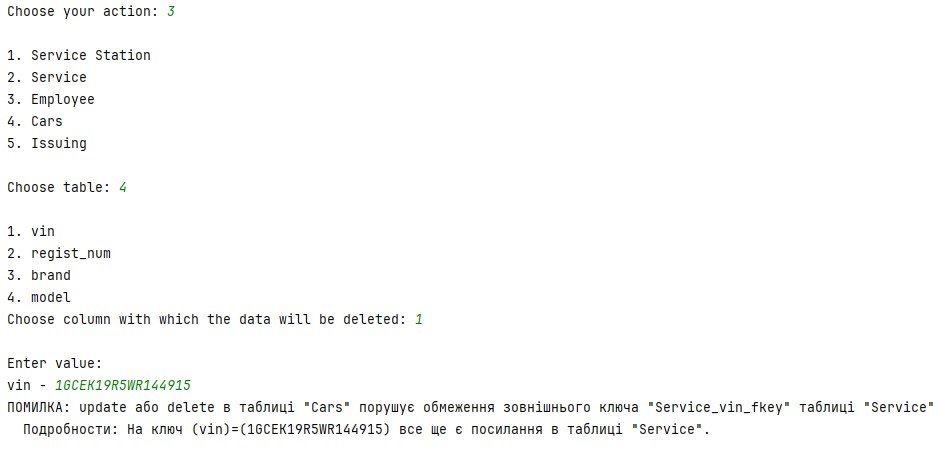
***Вдале вилучення записів:***



Вигляд таблиці Issuing та Cars після видалення відповідно:

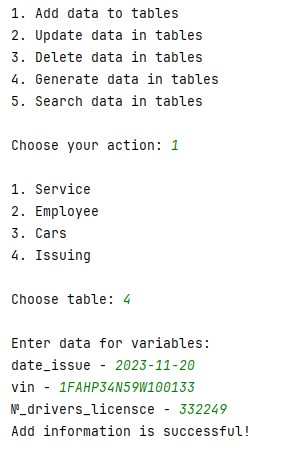


***Невдале додавання даних в таблицю:***



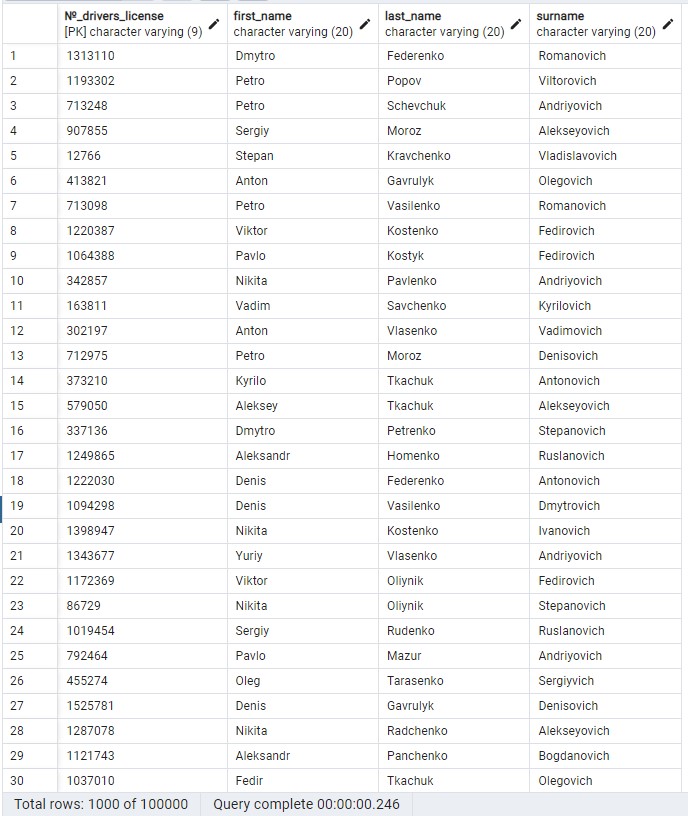
Ця помилка пов’язана з тим, що ми намагаємося видати автомобіль в користування (додати відповідний запис в таблицю Issuing), при цьому не маючи відповідного автомобіля в автопарку (таблиці “Cars”).

***Вдале додавання даних в таблицю:***



***Пункт №2:***

Генерація 100 000 працівників для таблиці Employee:



SQL запит для генерації працівників:

**insert** **into** "Employee" (№\_drivers\_license, first\_name, last\_name, surname)

**select**

row\_number() OVER () **as** №\_drivers\_license,

first\_name,

last\_name,

surname

**from** generate\_series(**1**, **100**) **as** №\_drivers\_license

**cross** **join**

(**select** **unnest**(array['Ivan','Dmytro','Aleksandr','Aleksey','Vadim','Viktor','Kyrilo','Denis','Nikita','Oleg','Pavlo','Petro','Roman','Ruslan','Sergiy','Stepan','Fedir','Anton','Andriy','Yuriy']) **as** first\_name) **as** f

**cross** **join**

(**select** **unnest**(array['Melnyk','Schevchenko','Kovalenko','Bondarenko','Tkaschenko','Kravchenko','Kovalchuk','Schevchuk','Oliynik','Tkachuk','Savchenko','Bondar','Marchenko','Rudenko','Moroz','Petrenko','Pavlenko','Vasilenko','Levchenko','Karpenko','Gavrulyk','Popov','Panchenko','Mazur','Homenko','Goncharenko','Kostenko','Kostyk','Kozak','Federenko','Kovtun','Bilous','Nesterenko','Tarasenko','Vovk','Zhuk','Vlasenko','Radchenko','Voloshyn','Velyshko']) **as** last\_name) **as** l

**cross** **join**

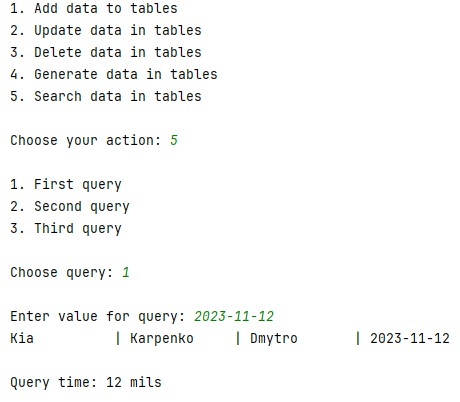
(**select** **unnest**(array['Alexandrovich','Sergiyvich','Petrovich','Bogdanovich','Denisovich','Olegovich','Ruslanovich','Romanovich','Stepanovich','Fedirovich','Antonovich','Andriyovich','Viltorovich','Ivanovich','Vadimovich','Kyrilovich','Alekseyovich','Dmytrovich','Maksymovich','Vladislavovich']) **as** surname) **as** s

**order** **by** random()

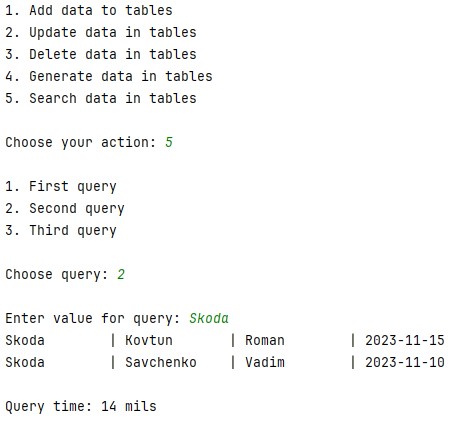
**LIMIT** **100000**;

***Пункт №3:***

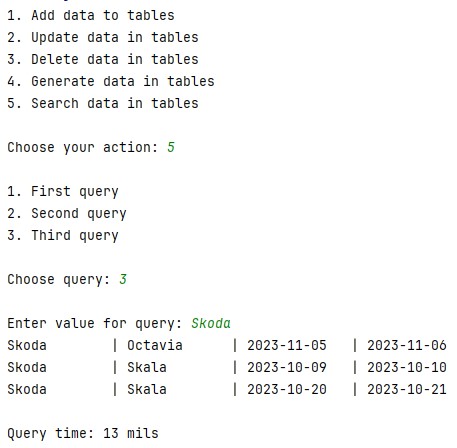
Перший запит:



Другий запит:



Третій запит:



Перший запит:

**SELECT** "Cars".brand, "Employee".last\_name, "Employee".first\_name, "Issuing".date\_issue **from** "Issuing"

**JOIN** "Cars" **ON** "Issuing".vin = "Cars".vin

**JOIN** "Employee" **ON** "Issuing".№\_drivers\_licensce = "Employee".№\_drivers\_license

**WHERE** "Issuing".date\_issue = '2023-11-12'

**GROUP** **BY** "Cars".brand, "Employee".first\_name, "Employee".last\_name, "Issuing".date\_issue

Другий запит:

**SELECT** "Cars".brand, "Employee".last\_name, "Employee".first\_name, "Issuing".date\_issue **from** "Issuing"

**JOIN** "Cars" **ON** "Issuing".vin = "Cars".vin

**JOIN** "Employee" **ON** "Issuing".№\_drivers\_licensce = "Employee".№\_drivers\_license

**WHERE** "Cars".brand = 'Skoda'

**GROUP** **BY** "Cars".brand, "Employee".first\_name, "Employee".last\_name, "Issuing".date\_issue

Третій запит:

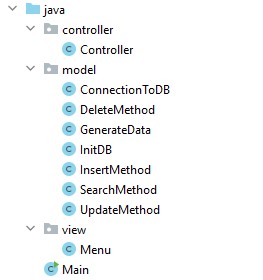
**SELECT** "Cars".brand, "Cars".model, "Service".service\_start\_date, "Service".service\_end\_date **FROM** "Service"

**JOIN** "Cars" **ON** "Cars".vin = "Service".vin

**WHERE** "Cars".brand = 'Skoda'

**GROUP** **BY** "Cars".brand, "Cars".model, "Service".service\_start\_date, "Service".service\_end\_date

***Пункт №4:***



*Файлова структура програми*

* ConnectionToDB – в цьому файлі створюється підключення до бази даних. Для цього використовується адреса БД, ім’я користувача та пароль.



* DeleteMethod – в цьому файлі прописаний метод для видалення даних з таблиці. Також, якщо ми видаляємо данні з таблиці Issuing чи Service, то перед цим перевіряється чи не використовується автомобіль або чи не стоїть автомобіль на даний момент на обслуговуванні.



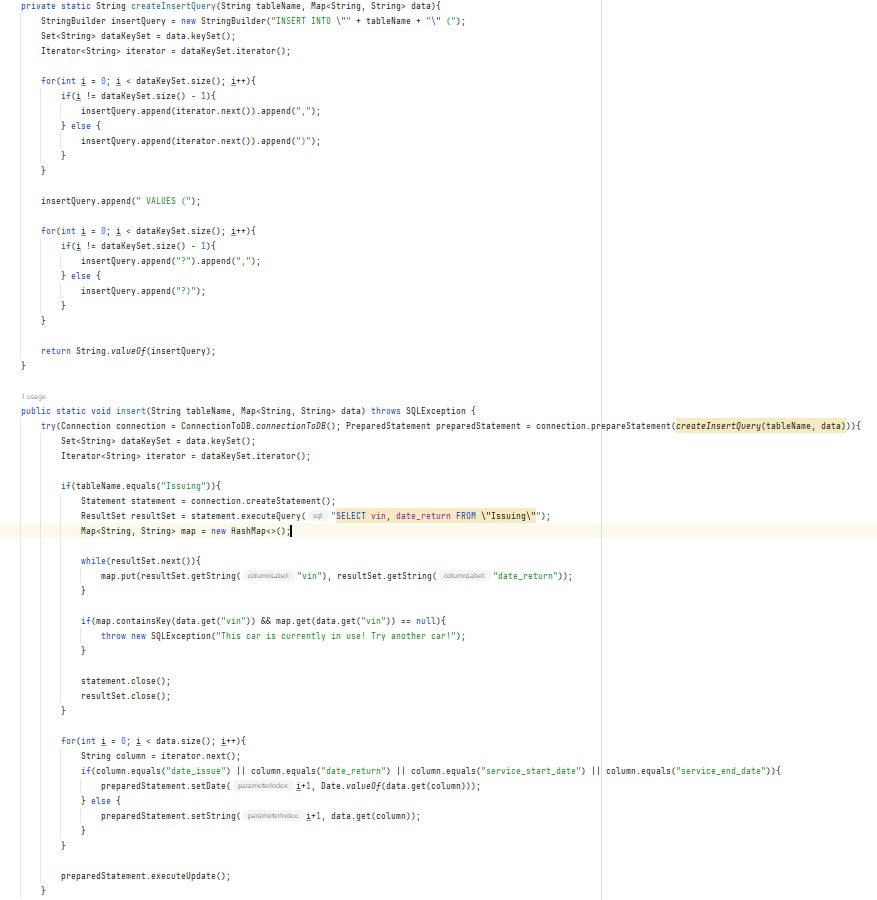
* GenerateMethod - в цьому файлі прописаний метод для генерування даних в таблиці.



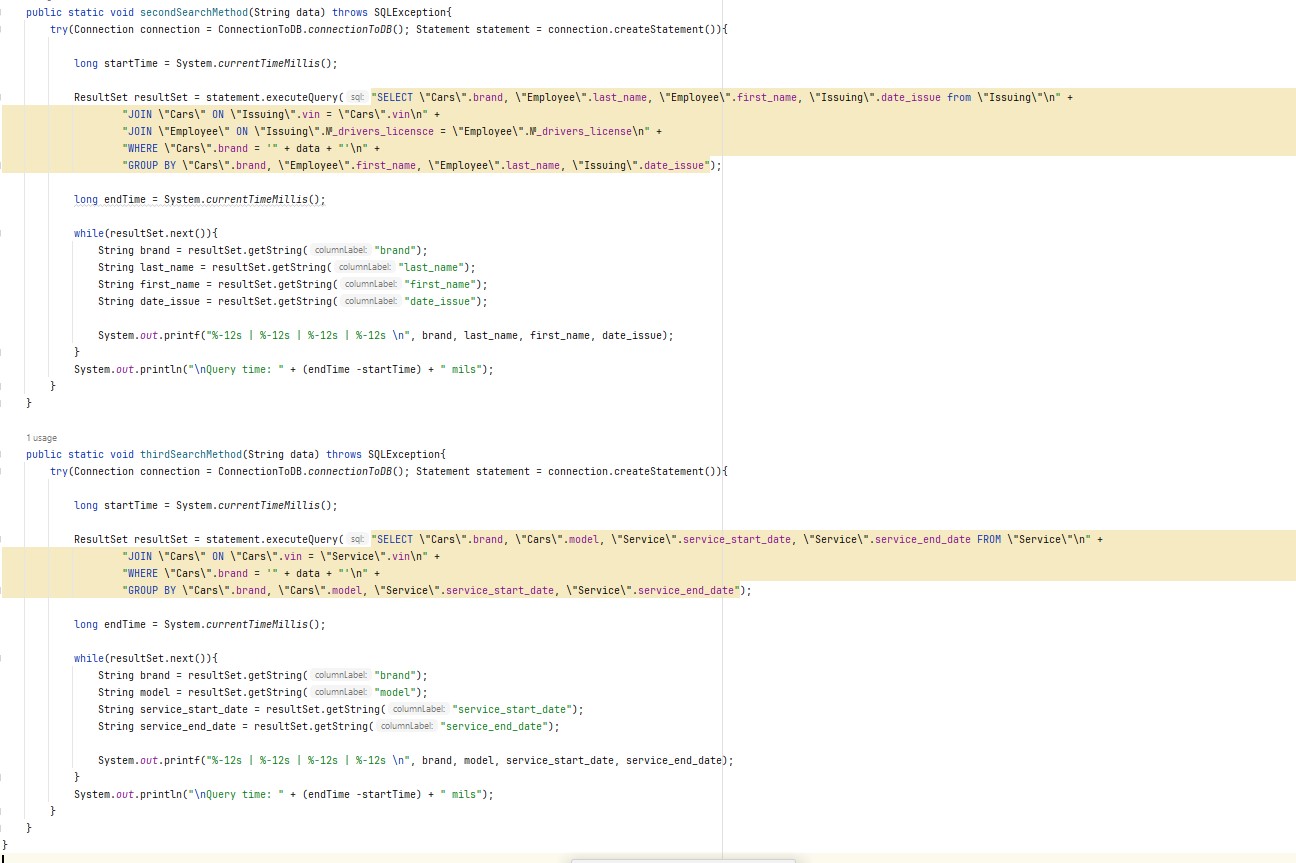
* InitDB - в цьому файлі прописані два методи: для отримання множини таблиць, для отримання множини колонок певної таблиці.



* InsertMethod – в цьому файлі прописана функція для вставки значень в таблицю, а також для генерації самого запиту Insert



* SearchMethod – в цьому файлі прописані 3 функції-запиту для пошуку інформації в базі даних.



* UpdateMethod – в цьому файлі прописана функція для оновлення даних в таблиці.

