

Міністерство освіти і науки України

Національний технічний університет України „КПІ імені Ігоря Сікорського ”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**Звіт до комп’ютерного практикуму №6**

**З дисципліни «Основи Back-end технологій»**

Прийняв: Виконав:

Викладач Зубко Роман Студент 3 курсу Бондарчук Анастасія, гр. ІП-12

«\_29\_»\_\_травня\_\_2024 р.

**2024 р.**

**Лабораторна робота №6.**

**GraphQL. Створення Schema GraphQL та Resolvers. Створення Query та Mutation.**

**Завдання.**

1. На своїй БД (розробленої в лаб. роб. #5) за допомогою Schema Definition Language (SDL) створити схему GraphQL.
2. Додати Resolvers для виконання операцій GraphQL.
3. Створити та виконати Query та Mutation для виконання операцій додавання, редагування та видалення інформації (CRUD) в БД.
4. Виконати дослідження роботи створених query та mutation за допомогою Postman.

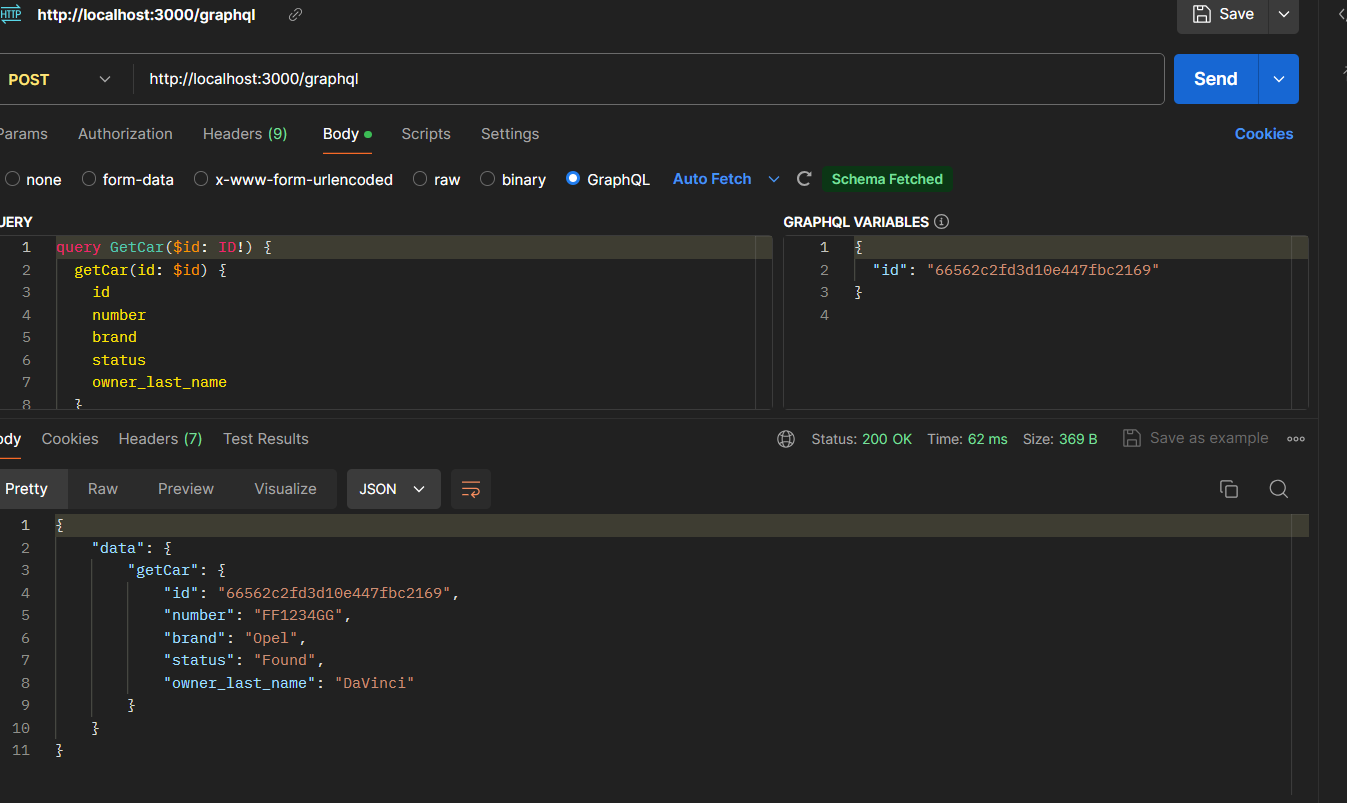
**Хід роботи**

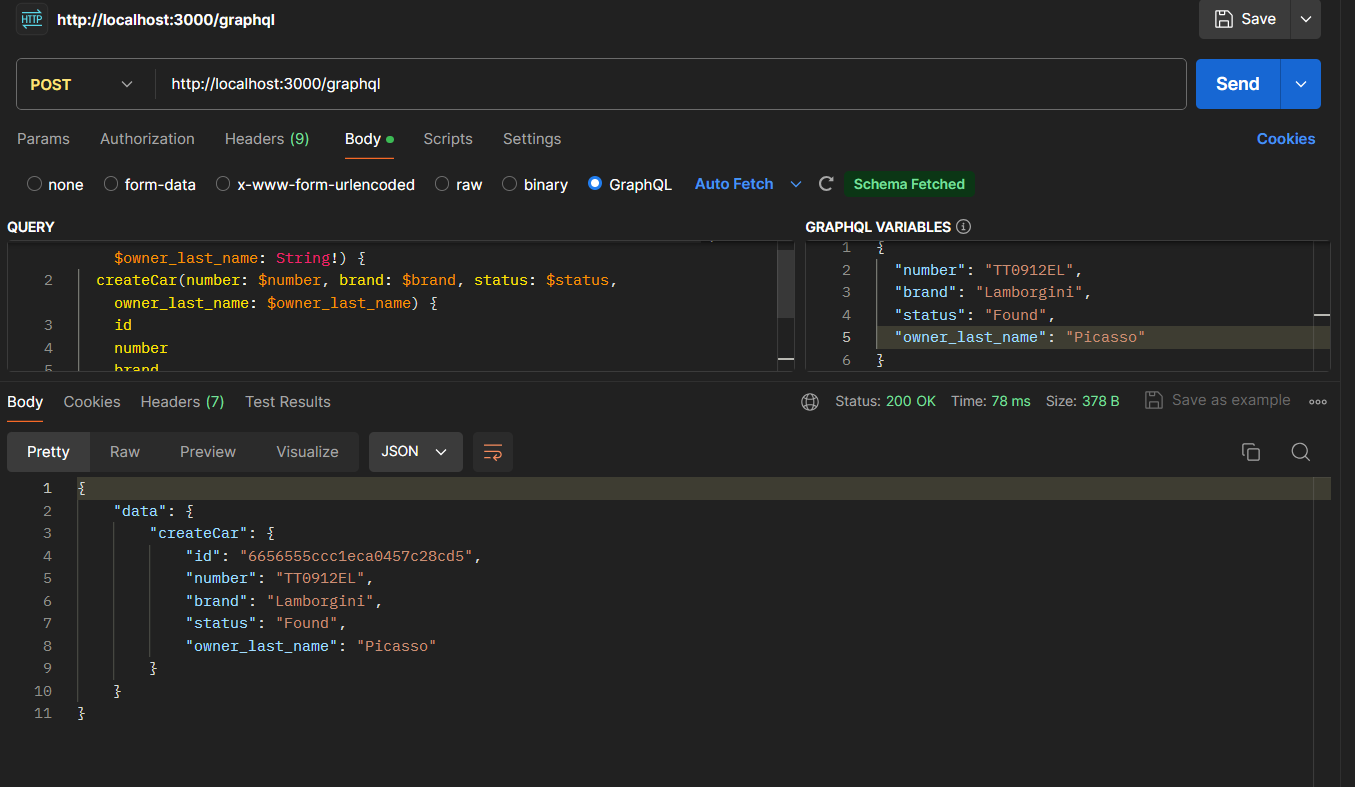
1. На своїй БД за допомогою Schema Definition Language (SDL) створено схему GraphQL.

const { buildSchema } = ***require***('graphql');  
  
***module***.exports = buildSchema(`  
 type StolenCar {  
 id: ID!  
 number: String!  
 brand: String!  
 status: String!  
 owner\_last\_name: String!  
 }  
  
 type Query {  
 getCar(id: ID!): StolenCar  
 getCars: [StolenCar]  
 }  
  
 type Mutation {  
 createCar(number: String!, brand: String!, status: String!, owner\_last\_name: String!): StolenCar  
 updateCar(id: ID!, number: String, brand: String, status: String, owner\_last\_name: String): StolenCar  
 deleteCar(id: ID!): StolenCar  
 }  
`);

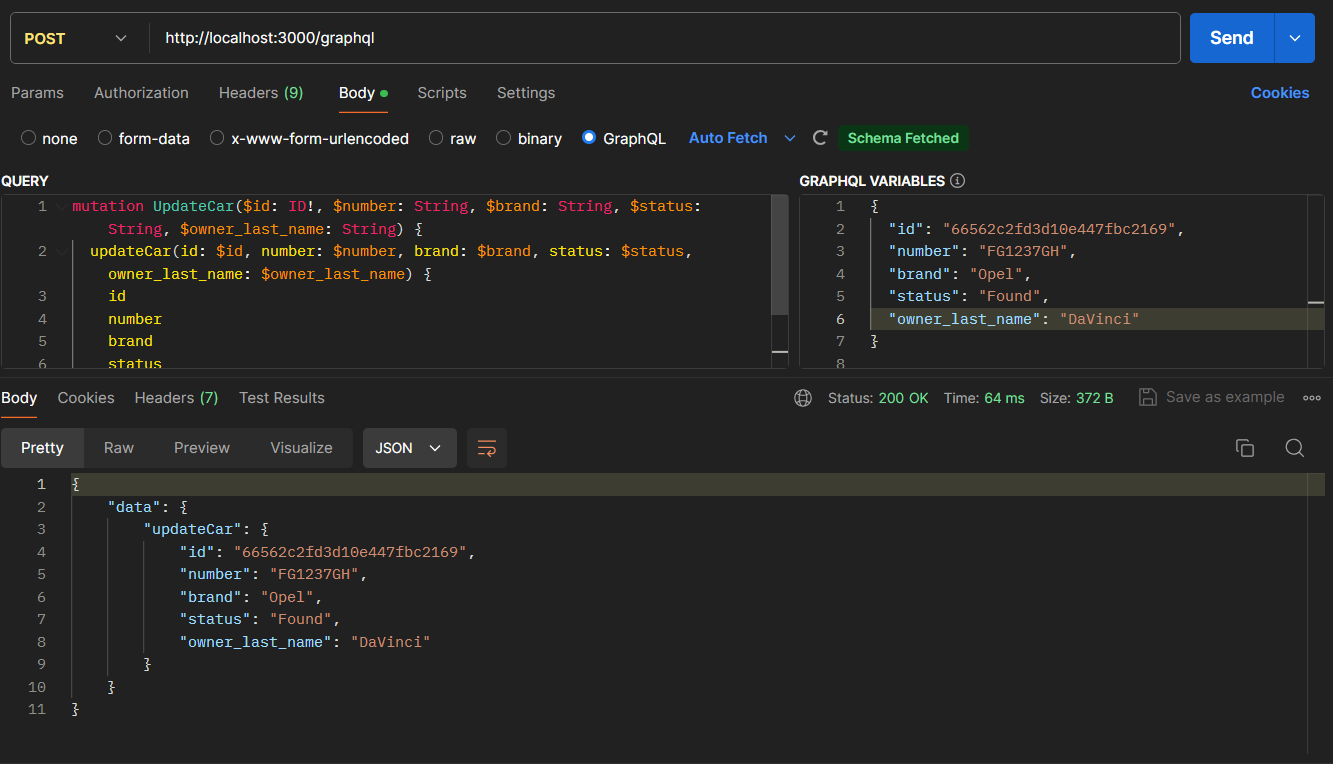
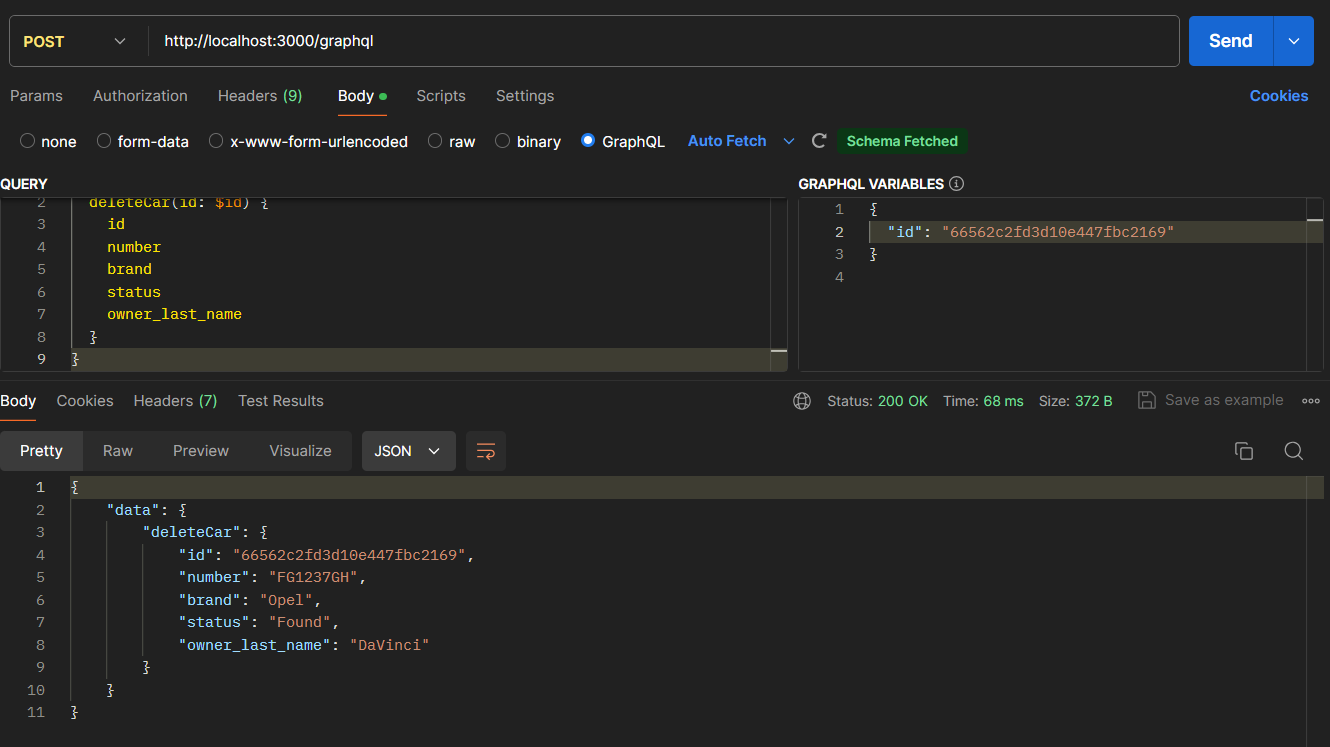
1. Додано Resolvers для виконання операцій GraphQL.

const ***StolenCar*** = ***require***('../models/StolenCar');  
  
const ***resolvers*** = {  
 getCar: async ({ id }) => {  
 return await ***StolenCar***.findById(id);  
 },  
 getCars: async () => {  
 return await StolenCar.find();  
 },  
 createCar: async ({ number, brand, status, owner\_last\_name }) => {  
 const newCar = new StolenCar({ number, brand, status, owner\_last\_name });  
 return await newCar.save();  
 },  
 updateCar: async ({ id, number, brand, status, owner\_last\_name }) => {  
 return await StolenCar.findByIdAndUpdate(id, { number, brand, status, owner\_last\_name }, { new: true });  
 },  
 deleteCar: async ({ id }) => {  
 return await StolenCar.findByIdAndDelete(id);  
 }  
};  
  
module.exports = resolvers;

1. Створено та виконано Query та Mutation для виконання операцій додавання, редагування та видалення інформації (CRUD) в БД.
2. Виконати дослідження роботи створених query та mutation за допомогою Postman.

Мал.1 – Результат операції читання даних

Мал.2 – Результат операції додавання даних

Мал.3 – Результат виконання операції редагування даних

Мал.4 – Результат виконання операції видалення даних

**Висновок:** Під час виконання лабораторної роботи було додано технологію graphql до існуючого проєкту. За допомогою mutations та query було реалізовано CRUD операції над базою даних, а також перевірено їх працездатність за допомогою Postman.

**Код програми додається:** <https://github.com/NastasaBondarchuck/BackEnd_Lab5-6>