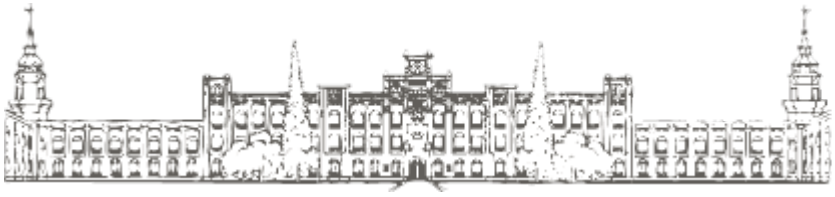


Міністерство освіти і науки України
Національний технічний університет України „КПІ
імені Ігоря Сікорського ”
Факультет інформатики та обчислювальної техніки
Кафедра інформатики і програмної інженерії

ЗВІТ

лабораторної роботи №5

з курсу «Основи WEB - технологій»

Тема: «GraphQL. Створення Schema GraphQL та Resolvers.
Створення Query та Mutation.»

Перевірив:

Викл. Альбрехт Й.О.

Виконала:

ст. Мельникова
Катерина гр. ІС-12

Київ 2024

1. **Завдання 1.**

Завдання.

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

Хід роботи

Повний код проєкту можна переглянути за посиланням:
https://github.com/katerynamelnykova/web-technologies-labs

Файл worker.resolver.js:

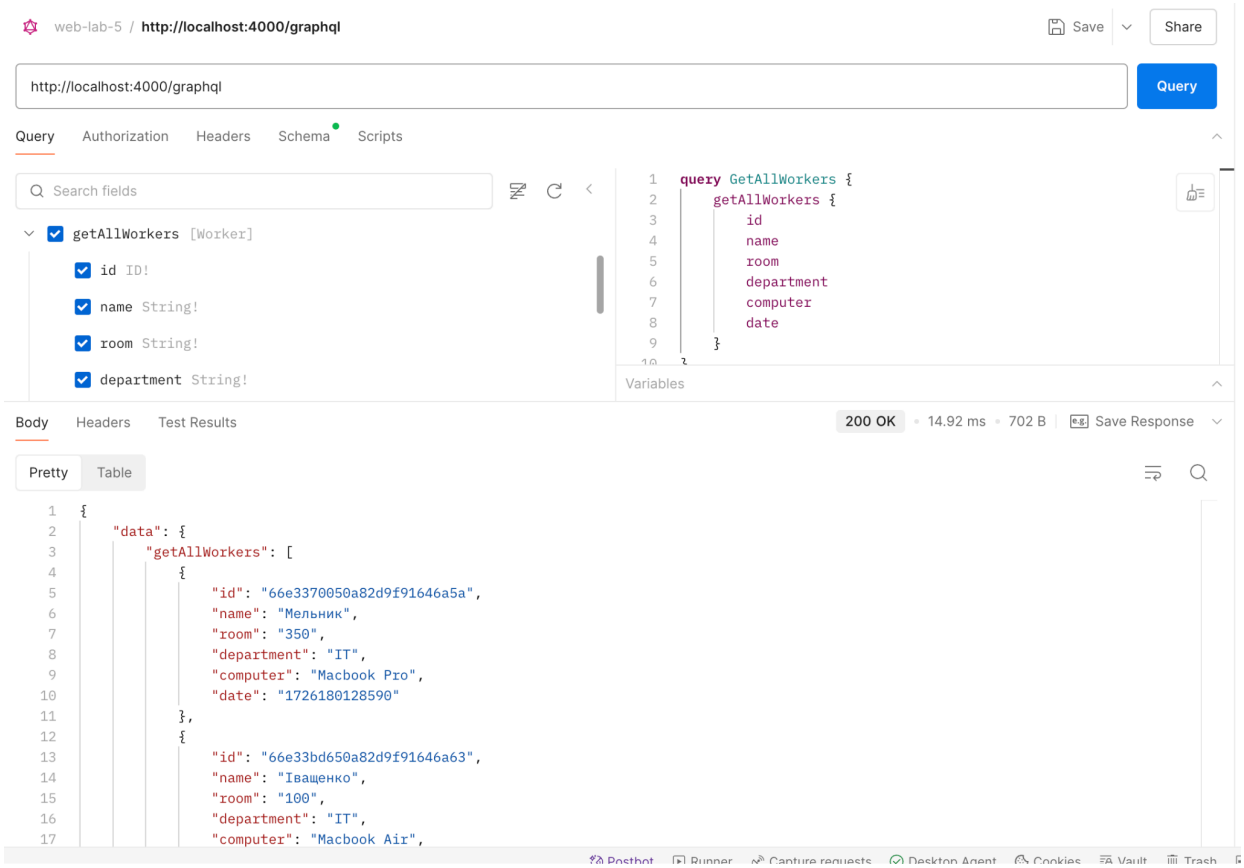
```
import { buildSchema } from 'graphql';
import { WorkerModel } from '../models/worker.model.js';

const schema = buildSchema(`
 type Worker {
   id: ID!
   name: String!
   room: String!
   department: String!
   computer: String!
   date: String
 }

 type Query {
   getWorkerById(id: ID!): Worker
   getAllWorkers: [Worker]
 }

 type Mutation {
   createWorker(name: String!, room: String!, department: String!, computer: String!): Worker
   updateWorker(id: ID!, name: String, room: String, department: String, computer: String):
Worker
   deleteWorker(id: ID!): Worker
 }
`);
```

```
const resolver = {
 getWorkerById: async ({ id }) => {
   try {
     const worker = await WorkerModel.findOne({ _id: id });
     return worker;
   } catch (error) {
     throw new Error('Error fetching worker by ID');
   }
 },
 getAllWorkers: async () => {
   try {
     const workers = await WorkerModel.find();
     return workers;
   } catch (error) {
     throw new Error('Error fetching all workers');
   }
 },
 createWorker: async ({ name, room, department, computer }) => {
   try {
     const newWorker = new WorkerModel({ name, room, department, computer });
     return await newWorker.save();
   } catch (error) {
     throw new Error('Error creating worker');
   }
 },
 updateWorker: async ({ id, name, room, department, computer }) => {
   try {
     return await WorkerModel.findOneAndUpdate(
       { _id: id },
       { name, room, department, computer },
       { new: true }
     );
   } catch (error) {
     throw new Error('Error updating worker');
   }
 },
 deleteWorker: async ({ id }) => {
   try {
     return await WorkerModel.findOneAndDelete({ _id: id });
   } catch (error) {
     throw new Error('Error deleting worker');
   }
 }
};
```

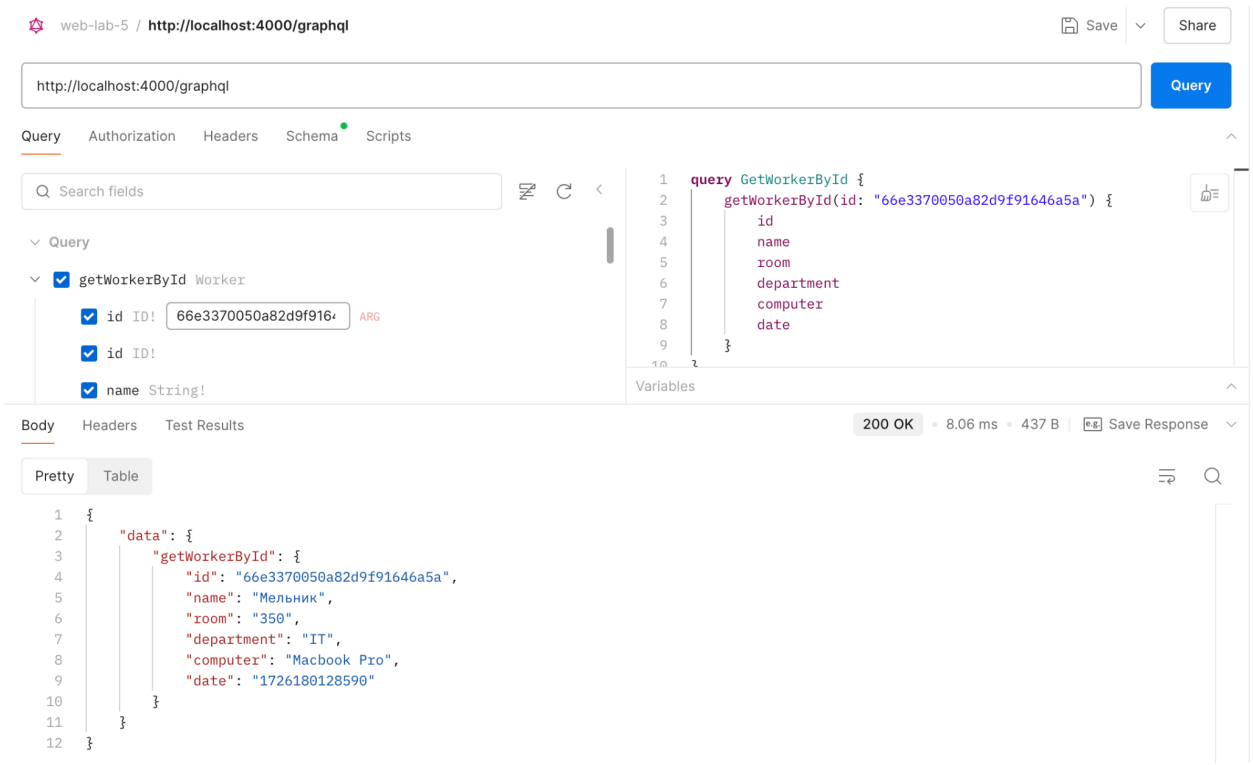
```
export { schema, resolver };
```

Отримані результати:

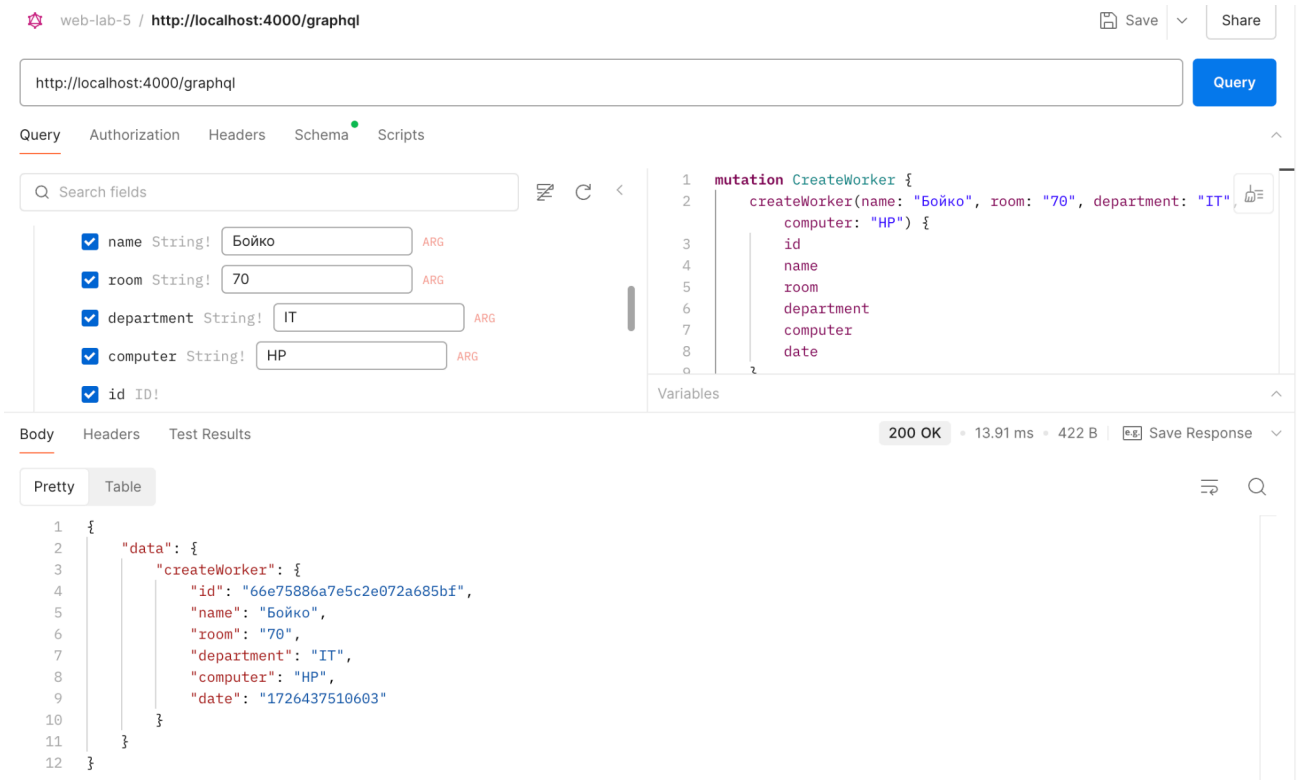
GetAllWorkers:



GetWorkerById:



CreateWorker:



web-technologies-labs.workers

4 DOCUMENTS 1 INDEXES

Documents | Aggregations | Schema | Explain Plan | Indexes | Validation

Filter | Type a query: { field: 'value' } | Reset | Find | More Options

ADD DATA | EXPORT COLLECTION | 1 - 4 of 4

```
department: "IT"
computer: "Macbook Air"
date: 2024-09-12T23:05:36.761+00:00
__v: 0
```

```
_id: ObjectId('66e741765cc9685707ae8231')
name: "Мазур"
room: "23"
department: "HR"
computer: "HP"
date: 2024-09-15T20:20:06.602+00:00
__v: 0
```

HP

```
_id: ObjectId('66e75886a7e5c2e072a685bf')
name: "Бойко"
room: "70"
department: "IT"
computer: "HP"
date: 2024-09-15T21:58:30.603+00:00
__v: 0
```

UpdateWorker:

web-lab-5 / http://localhost:4000/graphql | Save | Share

http://localhost:4000/graphql | Query

Query | Authorization | Headers | Schema | Scripts

Search fields

name String ARG

room String 60 ARG

department String ARG

computer String ARG

id ID!

```
mutation UpdateWorker {
  updateWorker(id: "66e75886a7e5c2e072a685bf", room: "60") {
    id
    name
    room
    department
    computer
    date
  }
}
```

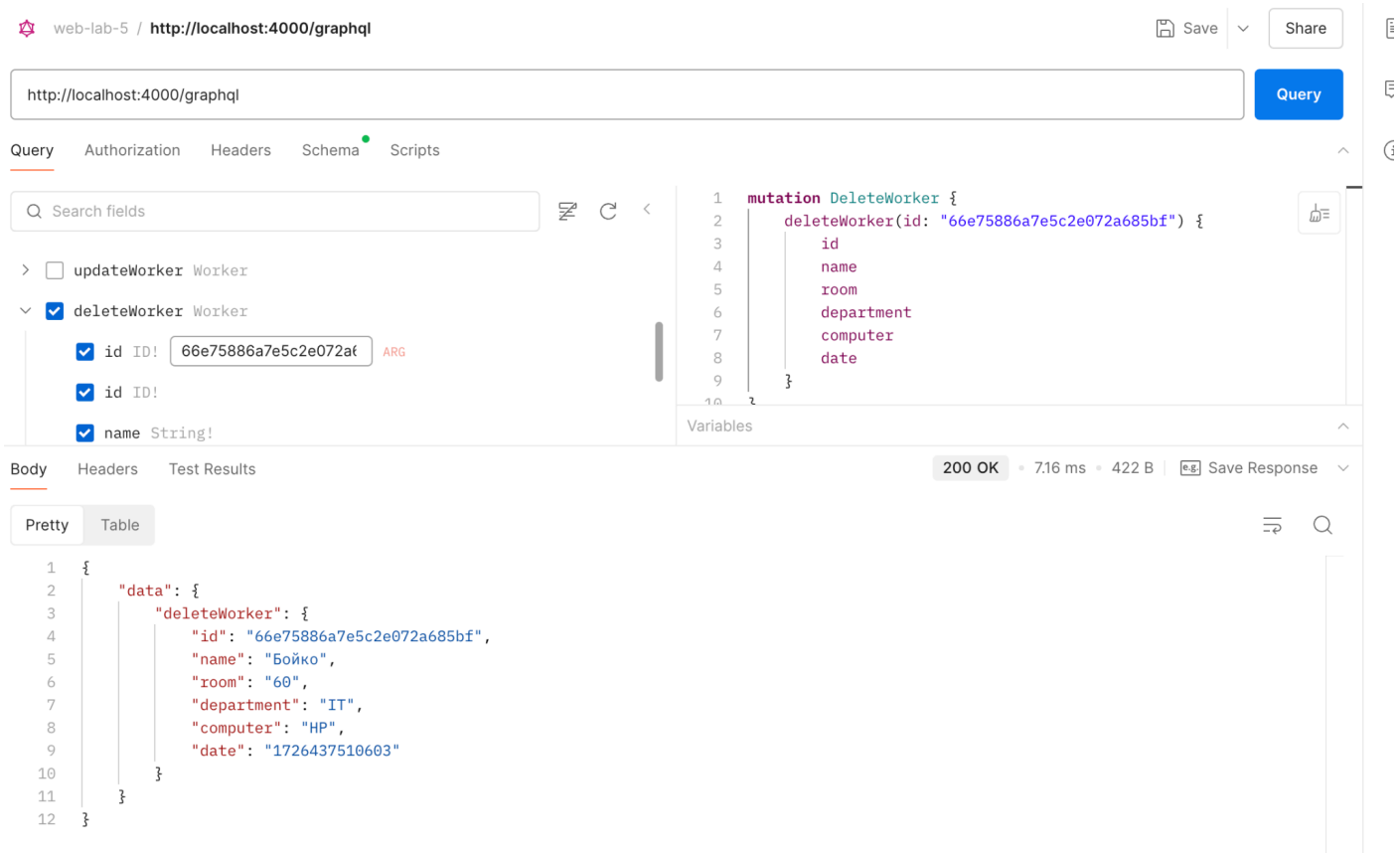
Variables

Body | Headers | Test Results | 200 OK | 9.55 ms | 422 B | Save Response

Pretty | Table


```
{
  "data": {
    "updateWorker": {
      "id": "66e75886a7e5c2e072a685bf",
      "name": "Бойко",
      "room": "60",
      "department": "IT",
      "computer": "HP",
      "date": "1726437510603"
    }
  }
}
```

DeleteWorker:



Висновок

Під час виконання даної лабораторної роботи я створила схему GraphQL за допомогою SDL, додала resolvers для виконання CRUD операцій у БД та виконала дослідження створених query та mutations за допомогою Postman.