Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Інститут прикладного системного аналізу

Курсова робота

з курсу «Організація баз даних та знань»

з теми «База даних ЗВО»

Виконав: студент 3 курсу

групи КА-01

Гаврилюк І.В.

Прийняв:

Корнага Я. І.

**Київ 2023**

**ЗМІСТ**

[ВСТУП3](#Вступ)

[РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ4](#Розділ1)

[РОЗДІЛ 2 АРХІТЕКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ БД6](#Розділ2)

[2.1 Аналіз функціонування та організаційні засади підприємства6](#Аналіз2)

[2.2 Проектування структури бази даних6](#Проектування2)

[2.3 Життєві цикли бази даних7](#Цикли2)

[РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ВЗАЄМОДІЇ З БД9](#Розділ3)

[3.1 Посібник користувача9](#Посібник3)

[3.2 Реалізація механізмів БД10](#Механізми3)

[3.3 Вимоги до апаратних і програмних засобів12](#Вимоги3)

[3.4 Випробування розроблених програм12](#Випробовування3)

[3.5 Опис тестової бази даних19](#Опис3)

[РОЗДІЛ 4 ВИСНОВОК20](#Розділ4)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ21](#Література)

[ДОДАТКИ 22](#Додатки)

[Додаток А Створення схеми бази даних22](#ДодаткиА)

[Додаток Б Реалізація SQL запитів у веб інтерфейсі44](#ДодаткиВ)

ВСТУП

У автопідприємства є достатньо багато систематичних даних, підпорядкованостей і т.п. Для розумного розподілу роботи, аналізу та керування створювалася дана система, яка має на меті пришвидшення та полегшення управління.

**Актуальність.** Така система дозволить швидше та легше отримувати доступ до потрібних даних, у випадку коли автопідприємство є достатньо великим. Важко уявити сучасне підприємство будь-якого типу, яке не використовувало подібну систему в своєму керуванні.

**Мета.** Метою роботи є розробка автоматизованої інформаційної системи для автопідприємства, яка дозволить зберігати всю необхідну інформацію, забезпечить виконання всіх видів інформаційних запитів, які необхідні при експлуатації даної системи.

**Завдання.** Спроектувати базу даних та підготувати усі необхідні запити та процедури для роботи з нею.

**Програмне забезпечення.** При виконанні роботи було використано СУБД MySQL 8.0, веб-інтерфейс реалізовано з використанням програми Open Server. Використовувалися ОС Windows 10, середовища розробки Visual Studio Code і MySQL Workbench.

**РОЗДІЛ 1**

**ПОСТАНОВКА ЗАДАЧІ**

Автопідприємство міста займається організацією пасажирських і вантажних перевезень у місті. У віданні підприємства знаходиться автотранспорт різного призначення: автобуси, таксі, маршрутні таксі та інший легковий транспорт. Вантажний транспорт, транспорт допоміжного характеру представлені різними марками. Кожна з перерахованих категорій транспорту має характеристики, властиві тільки цій категорії: наприклад, до характеристик тільки вантажного транспорту відноситься вантажопідйомність, пасажирський транспорт характеризується місткістю тощо. З плином часу, з одного боку, транспорт старіє і списується (можливо, продається), а з іншого, - підприємство поповнюється новим автотранспортом.

Підприємство має штат водіїв, закріплених за автомобілями (за одним автомобілем може бути закріплено більш одного водія). Обслуговуючий персонал (техніки, зварники, слюсарі, складальники тощо) займається технічним обслуговуванням автомобільної техніки, при цьому різні перераховані вище категорії також можуть мати унікальні для даної категорії атрибути. Обслуговуючий персонал і водії об'єднується в бригади, якими керують бригадири, далі йдуть майстри, потім начальники ділянок і цехів. У віданні підприємства знаходяться об'єкти гаражного господарства (цеху, гаражі, бокси тощо), де міститься і ремонтується автомобільна техніка.

Пасажирський автотранспорт (автобуси, маршрутні таксі) перевозить пасажирів за визначеними маршрутами, за кожним з них закріплені окремі одиниці автотранспорту. Ведеться облік числа перевезених пасажирів, на підставі чого проводиться перерозподіл транспорту з одного маршруту на інший. Враховується також пробіг, число ремонтів і витрати на ремонт по всьому автотранспорту, обсяг вантажних перевезень для вантажного транспорту, інтенсивність використання транспорту допоміжного призначення. Враховується інтенсивність роботи бригад по ремонту (число ремонтів, обсяг виконаних робіт), число замінених і відремонтованих вузлів і агрегатів (двигунів, КП, мости, шасі тощо) по кожній автомашині, і сумарно по ділянці, цеху та підприємству.

Види запитів в інформаційній системі:

1. Отримати дані про автопарк підприємства.
2. Отримати перелік і загальне число водіїв по підприємству, або для зазначеної автомашині.
3. Отримати розподіл водіїв по автомобілях.
4. Отримати дані про розподіл пасажирського автотранспорту по маршрутах.
5. Отримати відомості про пробіг автотранспорту певної категорії або конкретної автомашини за вказаний день, місяць та рік.
6. Отримати дані про число ремонтів та їх вартості для автотранспорту певної категорії, окремої марки автотранспорту або вказаної автомашини за вказаний період.
7. Отримати дані про підпорядкованість персоналу: робітники-бригадири - майстри - начальники ділянок і цехів.
8. Отримати відомості про наявність гаражного господарства в цілому і по кожній категорії транспорту.
9. Отримати дані про розподіл автотранспорту на підприємстві.
10. Отримати відомості про вантажоперевезення, виконаних зазначеною автомашиною за встановлений період.
11. Отримати дані про число використаних для ремонту вказаних вузлів і агрегатів для транспорту певної категорії, окремої марки автотранспорту або конкретної автомашини за вказаний період.
12. Отримати відомості про отримання та списання автотехніки за вказаний період.
13. Отримати склад підлеглих зазначеного бригадира, майстра тощо.
14. Отримати дані про роботи, виконані зазначеним фахівцем (зварником, слюсарем тощо) за означений період в цілому і по конкретній автомашині.

**РОЗДІЛ 2**

**АРХІТЕКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ БД**

**2.1 Аналіз функціонування та організаційні засади підприємства**

Центровою таблицею у нашій схемі є transport, де зберігаються всі авто. Іншою важливою таблицею є staff, де є всі робітники, що займаються ремонтом авто. Також створено driver з іменами всіх водіїв. За умовою задачі робітники і водії об’єднуються в бригади, створено таблицю brigada. Бригади працюють у відповідних цехах, створена для цього таблиця garage. Для контролю пробігу всіх авто створена таблиця race. Потрібно було розприділити пасажирські авто за маршрутами, за це відповідає таблиця route. І для вантажних авто потрібно відомості про перевезення, записані у таблиці vantaj. Також для запису даних про ремонтні роботи є таблиця repair.

**2.2 Проектування структури бази даних**

Структуру таблиць БД та зв’язки між ними зображено на EER-діаграмі.

Зв’язки між різними таблицями реалізовано через механізм Foreign Key. Наприклад, в таблиці vantaj та transport з’єднанні через Foreign Key у полях auto з таблиці vantaj та id з таблиці transport.

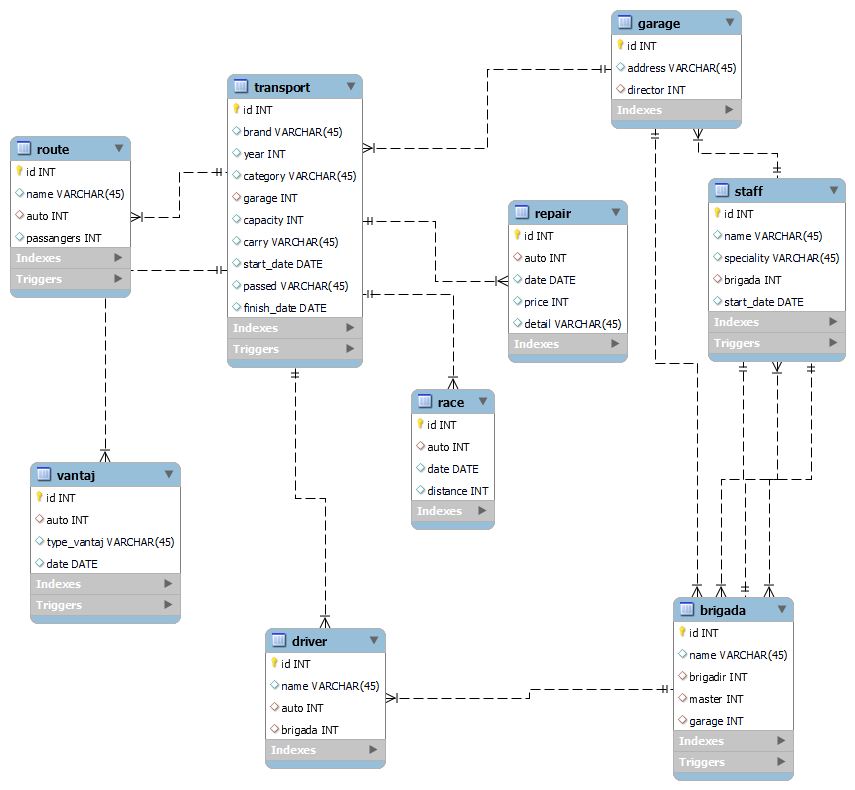


Рисунок 2.1 – EER-діаграма

**2.3 Життєві цикли бази даних**

1. Попереднє планування. На даному етапі було сформовано модель структури бази даних, визначено сутності та проаналізовано зв’язки між ними.

2. Перевірка здійсненності. Для розробки даної системи необхідно мати встановлений сервер MySQL 8.0, програму Open Server для запуску веб-сервера та середовище розробки MySQL Workbench.

3. Вимоги до даної інформаційної системи були сформовані в постановці завдання, зокрема, вказана структура організації та види інформаційних заходів, які мають бути реалізовані в системі.

4. Проектування. Для реалізації завдання обрано реляційну модель бази даних. В якості СУБД обрано MySQL. Створення початкової структури бази даних (сутності та зв’язки між ними) буде проведено в середовищі MySQL Workbench, а реалізацію веб-інтерфейсу доступу до БД та зв’язку між веб-інтерфейсом та MySQL-сервером – в середовищі Visual Studio Code з використанням мови PHP та програмного забезпечення Open Server.

**РОЗДІЛ 3**

**РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ВЗАЄМОДІЇ З БД**

**3.1 Посібник користувача**

Потрібно, щоб на ПК був встановлений MySQL Workbench 8.0 та Open Server. Запускаємо сервер у MySQL Workbench 8.0 та створюємо таблиці, процедури, тригери (Додаток А, ст.22) в моделі. Після цього у Open Server відкриваємо папку з проєктами та створюємо свою папку з назвою проєкту. У ній створюємо connection.php за наступним шаблоном:

<?php

$host="localhost:3306";//порт сервера Workbench

$user="root";//ім’я користувача сервера Workbench

$password="password";//пароль сервера Workbench

$database="auto";//назва схеми

?>

Потім уже додаємо у цю ж папку файли з розширенням .php, код яких копіюємо з Додатків. До цих файлів входить головна сторінка та сторінки з потрібними запитами. Потім запускаємо Open Server і у вкладці “Мої проекти” обираємо свій. Має відкритися головна сторінка застосунку у браузері

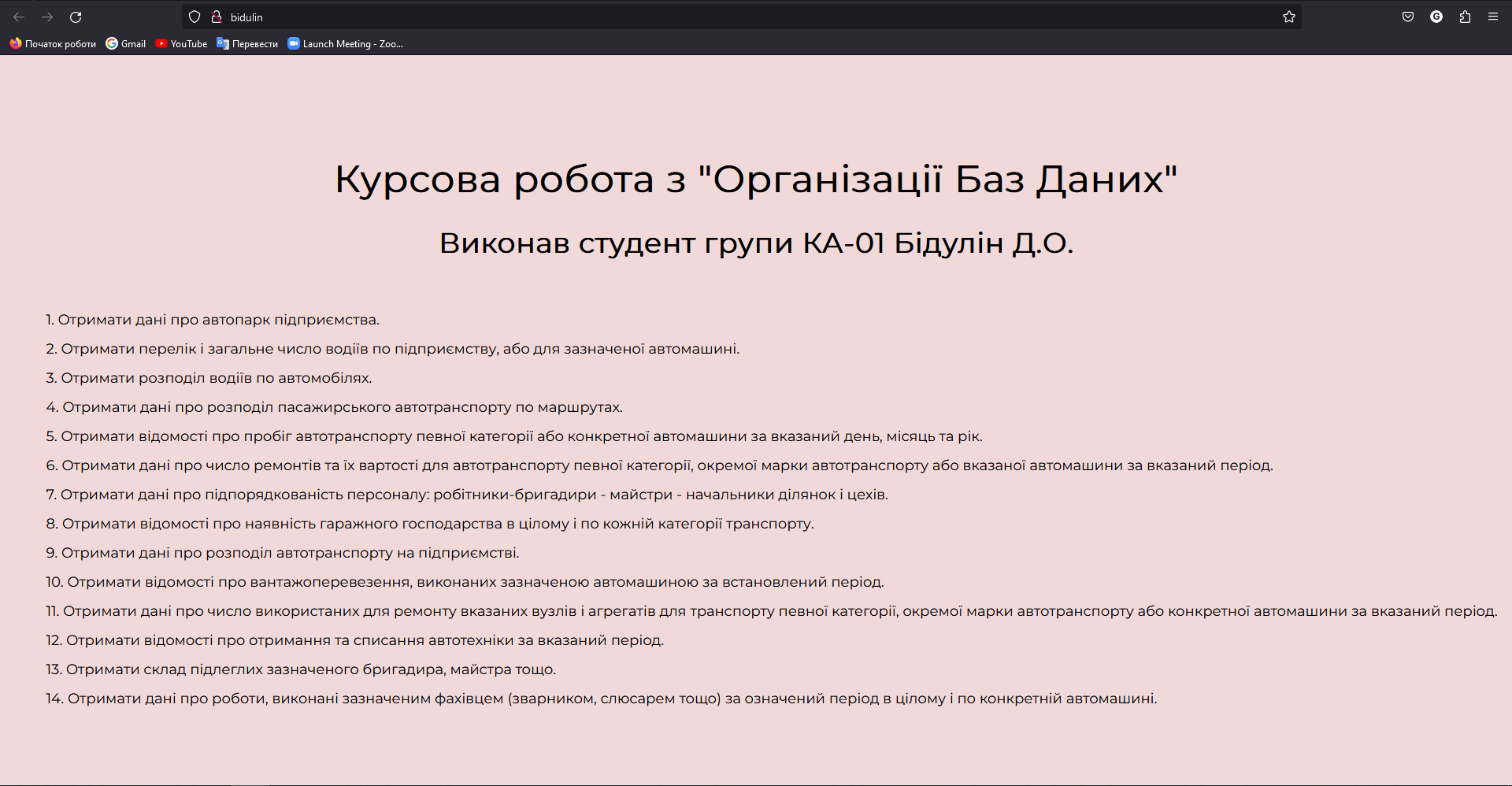


Рисунок 3.1– Головна сторінка веб-інтерфейсу

**3.2 Реалізація механізмів БД**

Були створені 12 тригерів для різних таблиць. Всі вони були створенні для вставки і оновлення в таблицях, тобто виходить лише 6. Для таблиці vantaj перевіряємо чи авто є вантажним. Так само для route чи він є пасажирським. У таблиці brigada перевіряємо чи робітник належить до даної бригади та чи не є вже бригадиром чи майстром(це унеможливлює випадок, коли одна людина є одночасно і майстром і бригадиром). Для таблиці staff додане поле start\_date (початок роботи на посаді) лише для спеціальностей Технік та Слюсар – це також реалізовано тригером. В таблиці transport так само реалізовано доступ до полів carry та capacity для ватнажних авто та пасажирських. І в цій же таблиці так само створено доступ до поля finish\_date, якщо авто є списаним.

**3.3 Вимоги до апаратних і програмних засобів**

Для роботи з інформаційною системою потрібно, щоб на ПК був встановлений MySQL Workbench 8.0 та Open Server. Також має бути присутнім сучасний браузер по типу Google Chrome або Firefox.

**3.4 Випробування розроблених програм**

Було перевірено роботу усіх 14 запитів, які необхідно було реалізувати. Підписи відповідно до нумерації запитів на ст. 4.

1. Oтримати дані про автопарк підприємства.

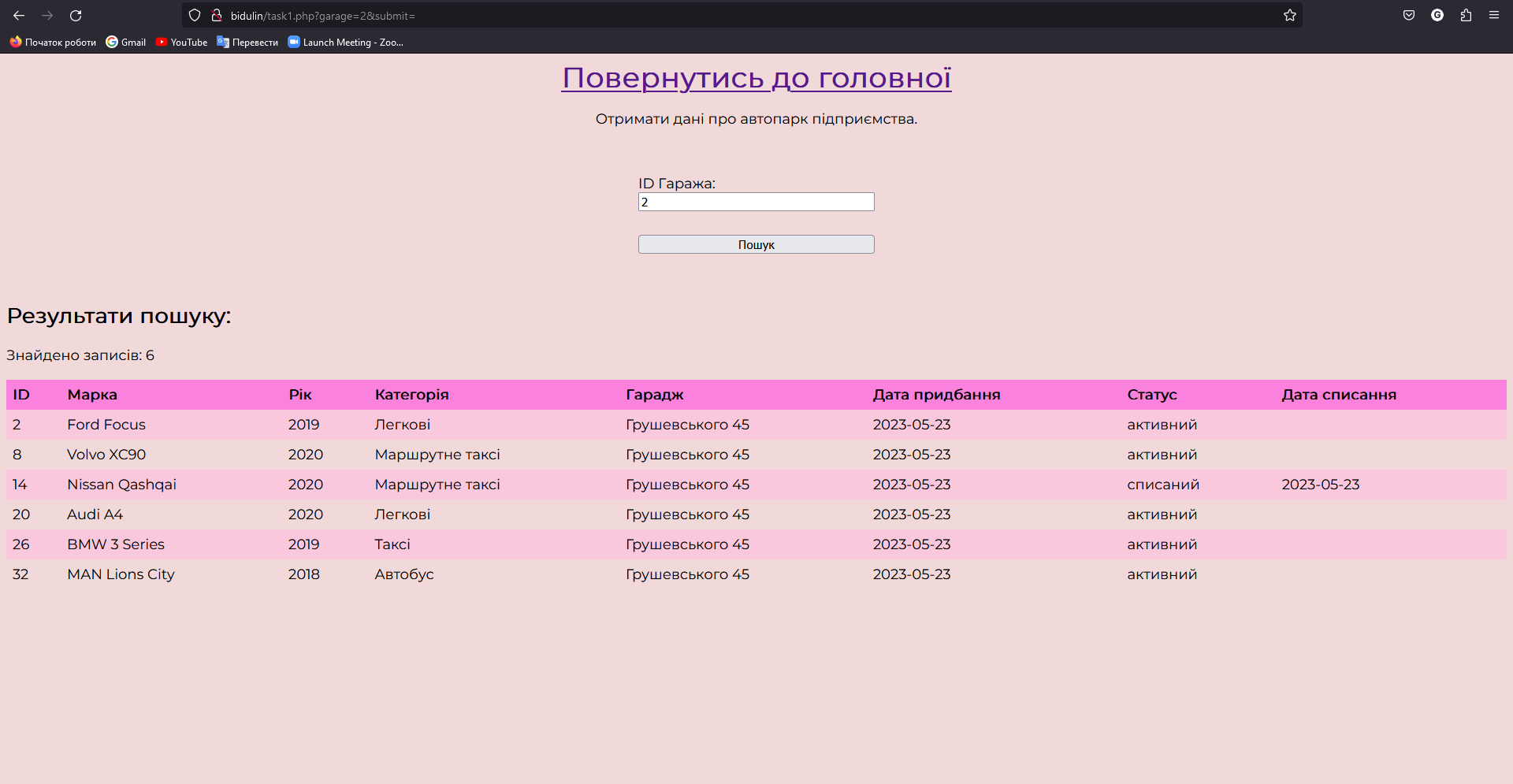


Рисунок 3.2 – Запит 1

1. Отримати перелік і загальне число водіїв по підприємству, або для зазначеної автомашині.

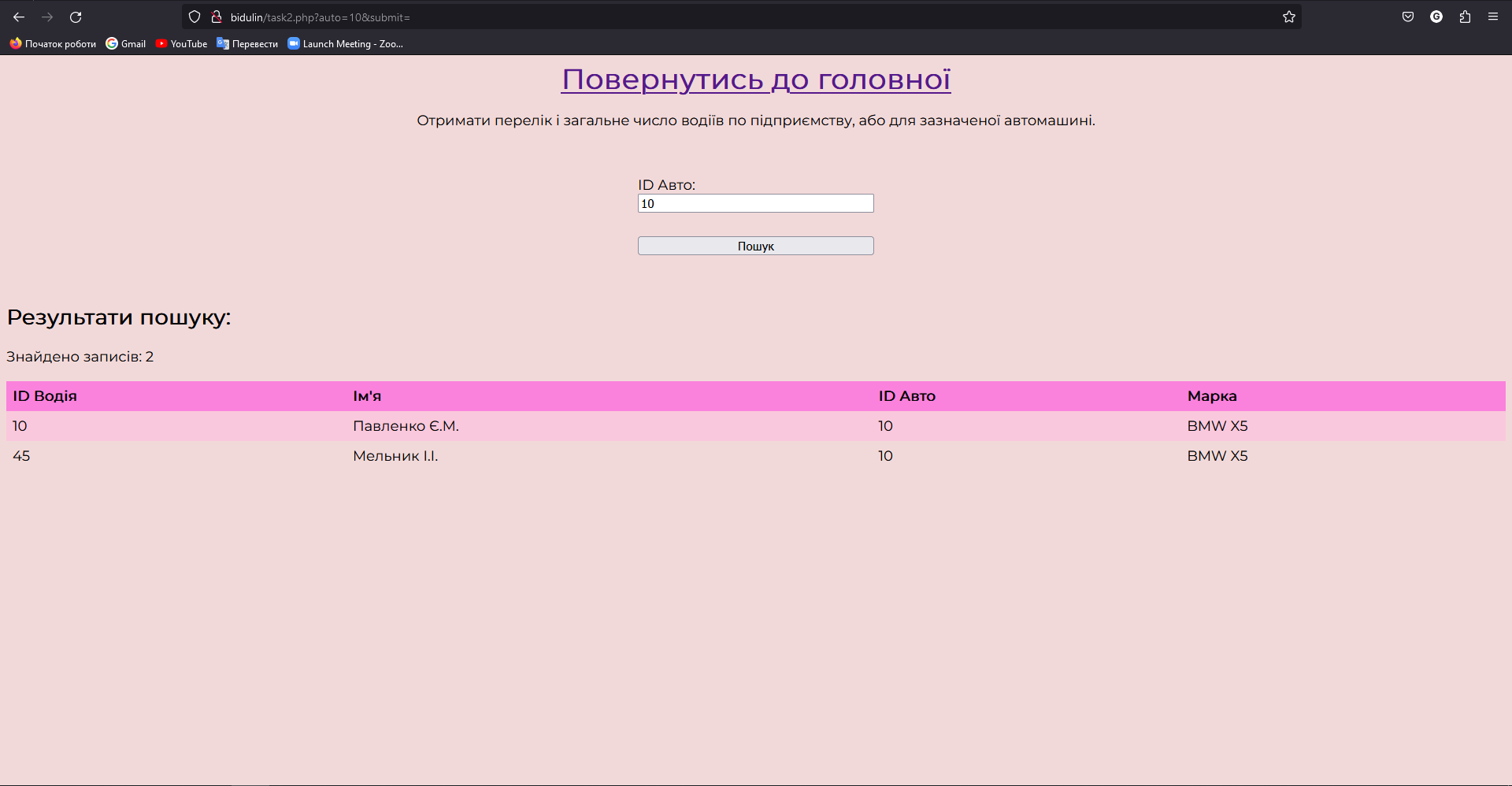


Рисунок 3.3 – Запит 2

1. Отримати розподіл водіїв по автомобілях.

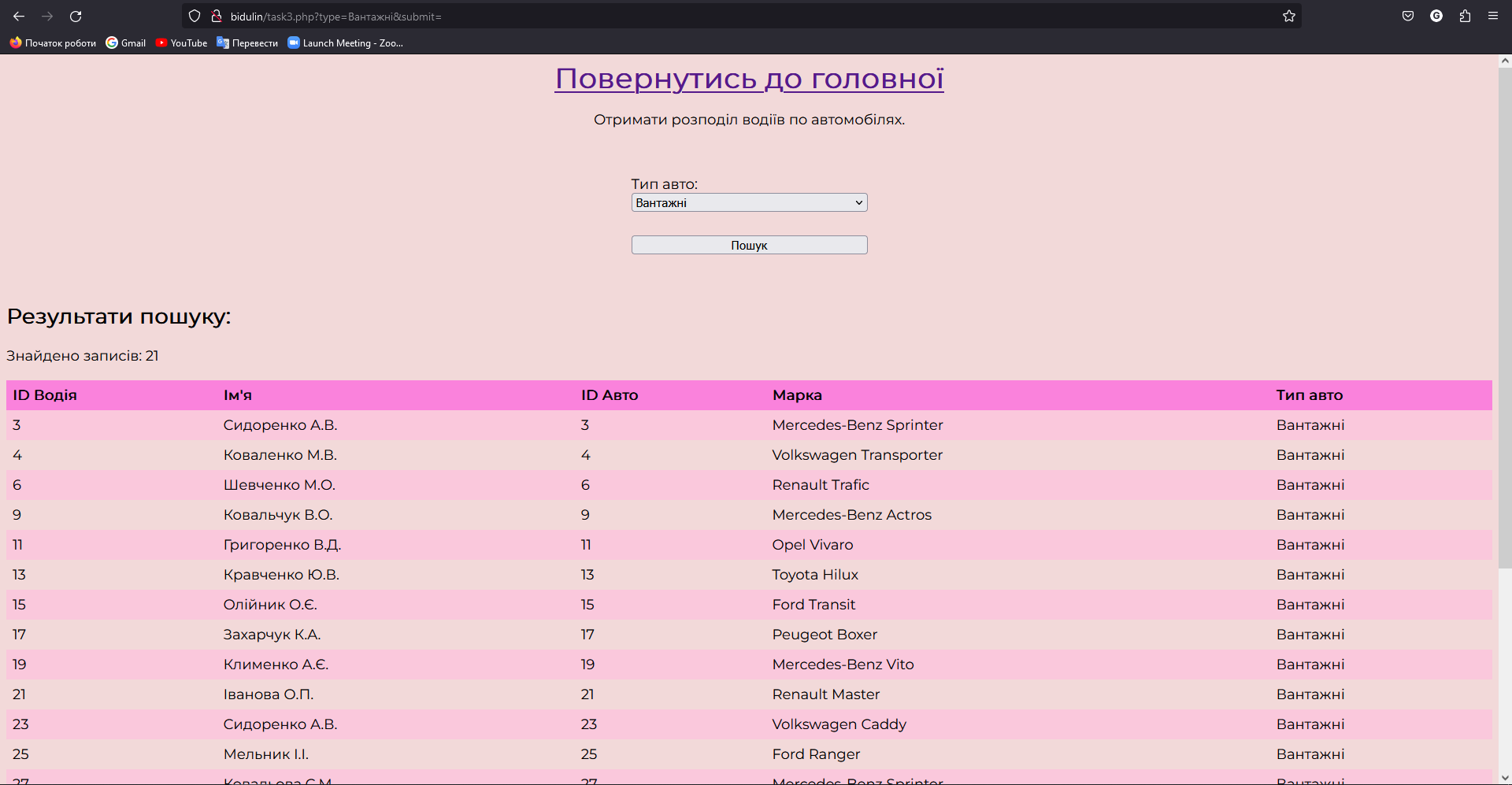


Рисунок 3.4 – Запит 3

1. Отримати дані про розподіл пасажирського автотранспорту по маршрутах.

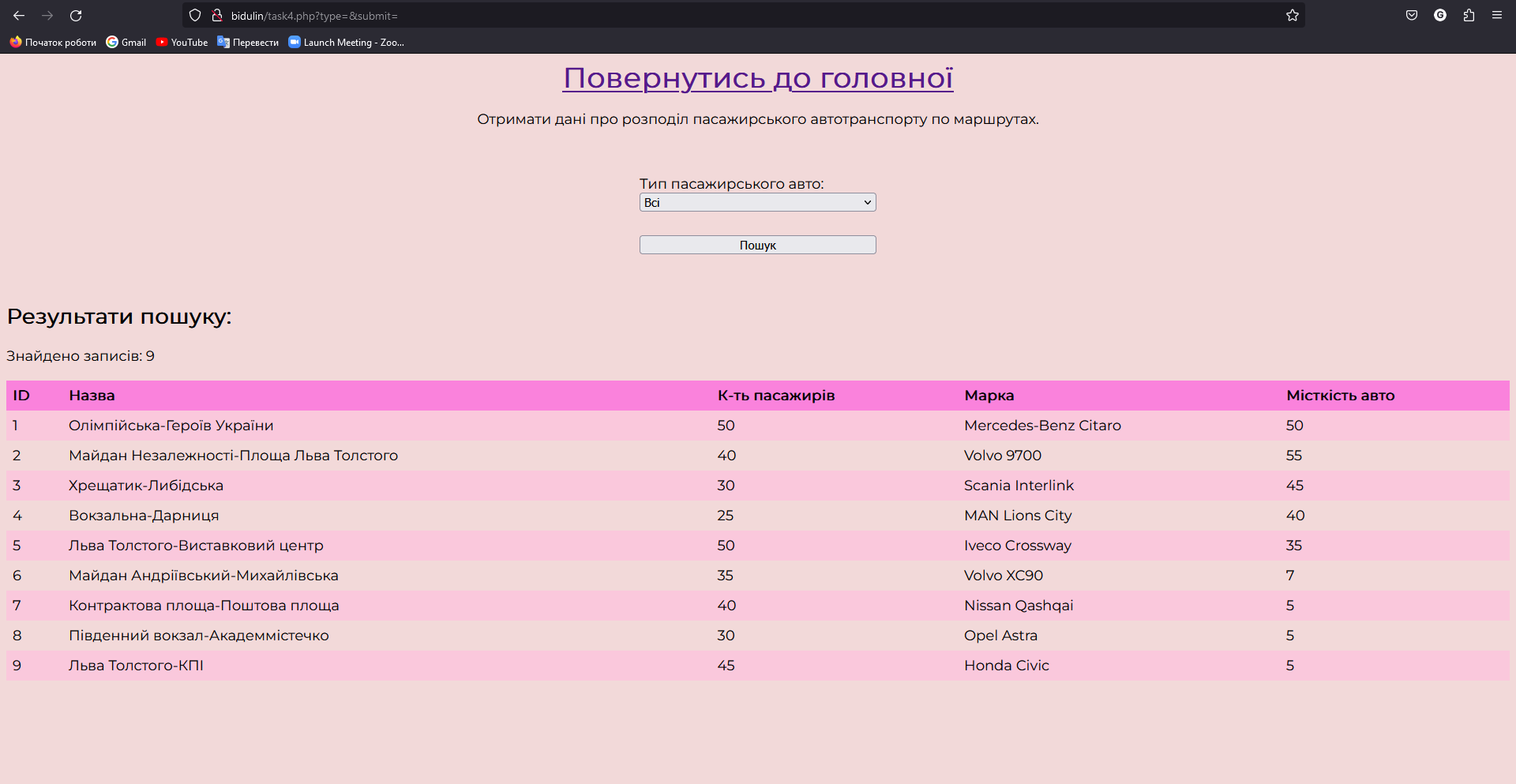


Рисунок 3.5 – Запит 4

1. Отримати відомості про пробіг автотранспорту певної категорії або конкретної автомашини за вказаний день, місяць та рік.

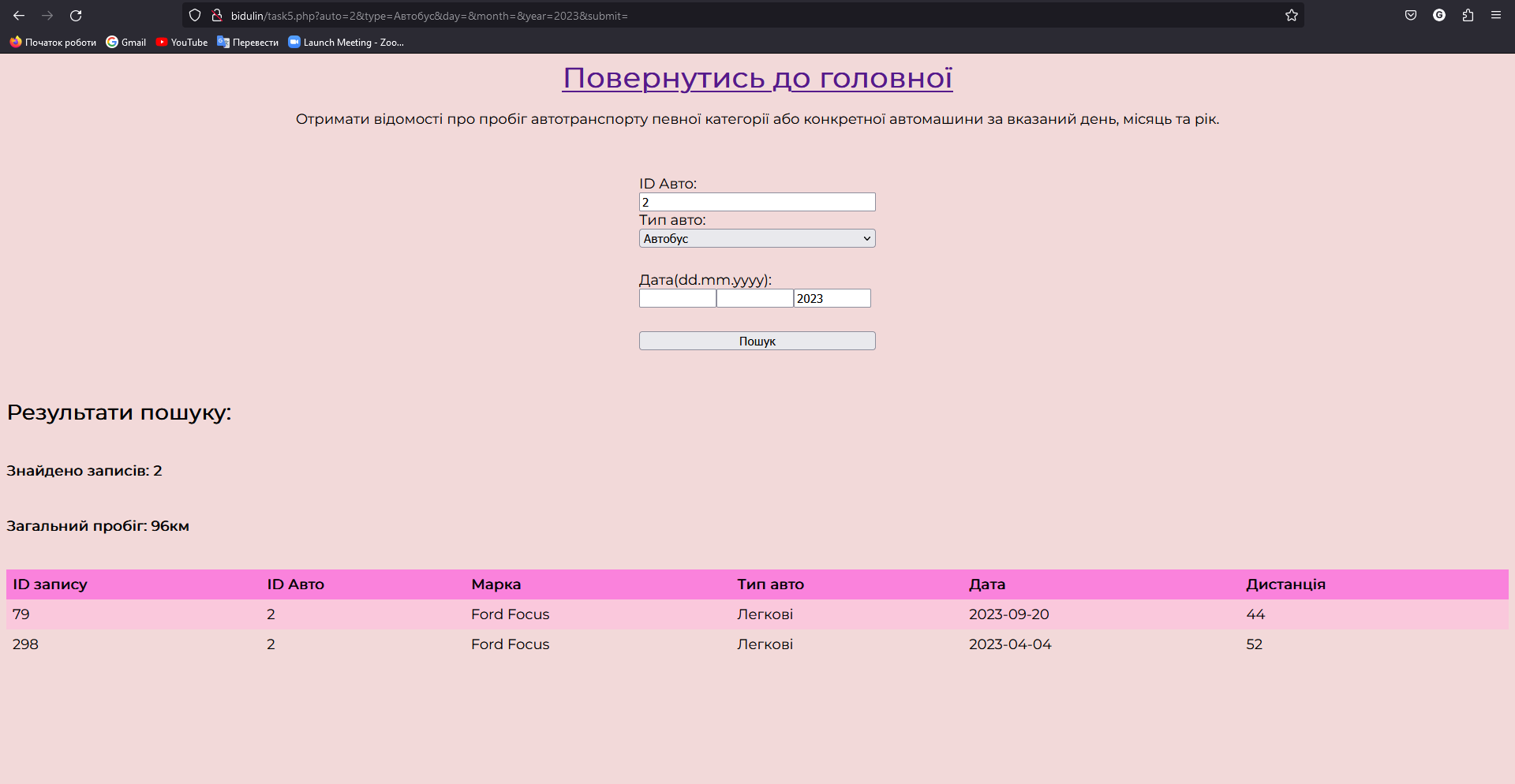


Рисунок 3.6 – Запит 5

1. Отримати дані про число ремонтів та їх вартості для автотранспорту певної категорії, окремої марки автотранспорту або вказаної автомашини за вказаний період.

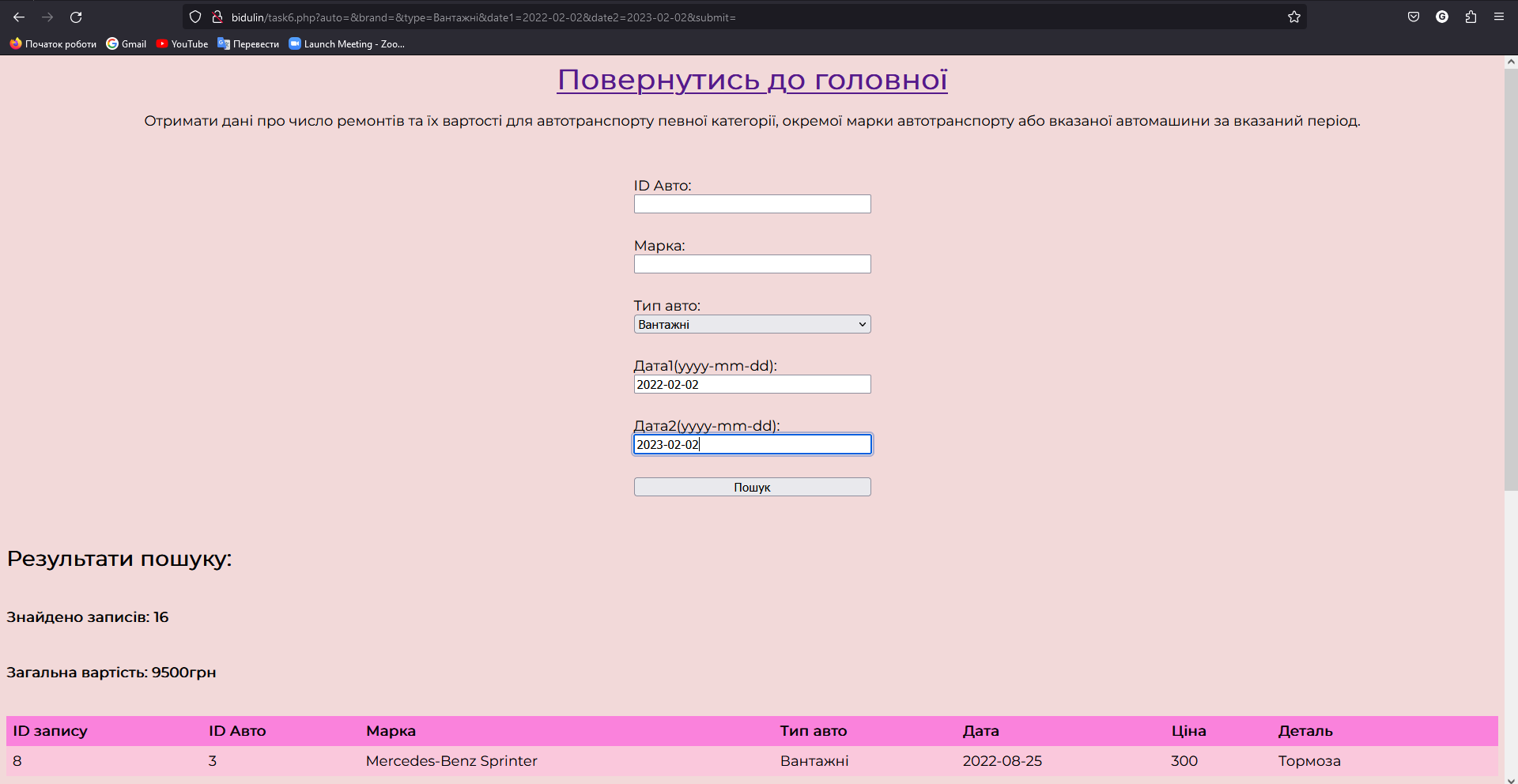


Рисунок 3.7 – Запит 6

1. Отримати дані про підпорядкованість персоналу: робітники-бригадири - майстри - начальники ділянок і цехів.

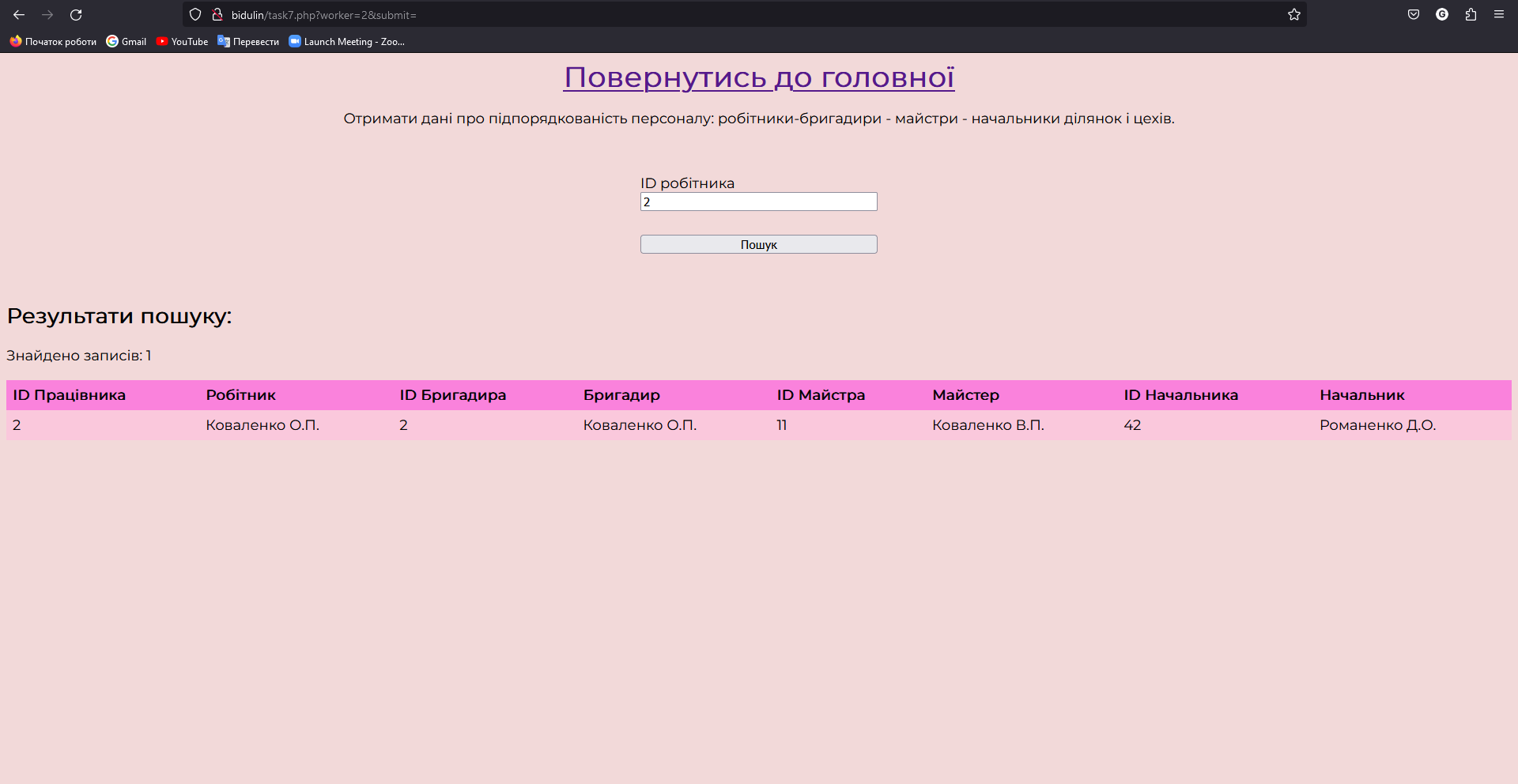


Рисунок 3.8 – Запит 7

1. Отримати відомості про наявність гаражного господарства в цілому і по кожній категорії транспорту.

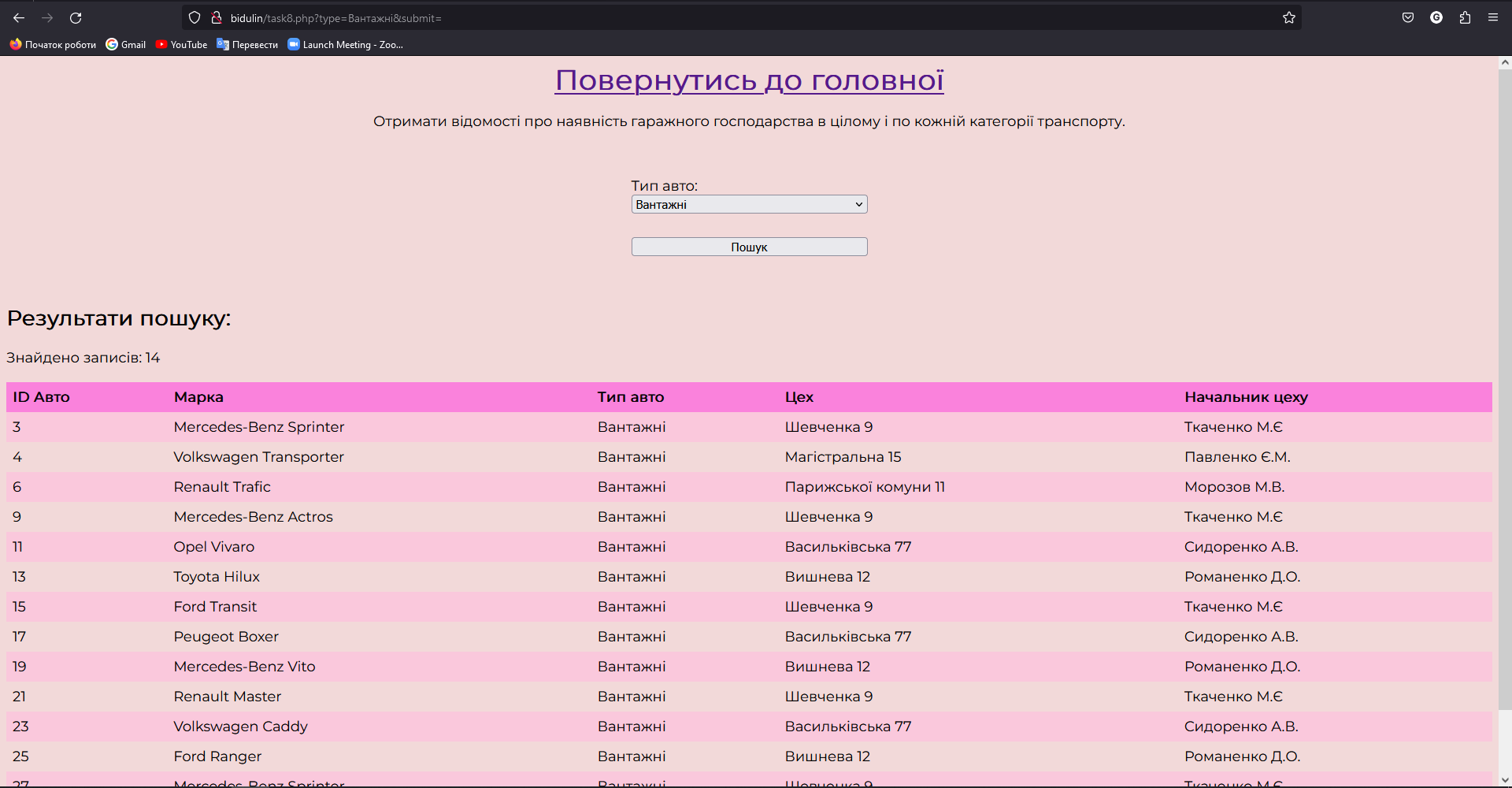


Рисунок 3.9 – Запит 8

1. Отримати дані про розподіл автотранспорту на підприємстві.

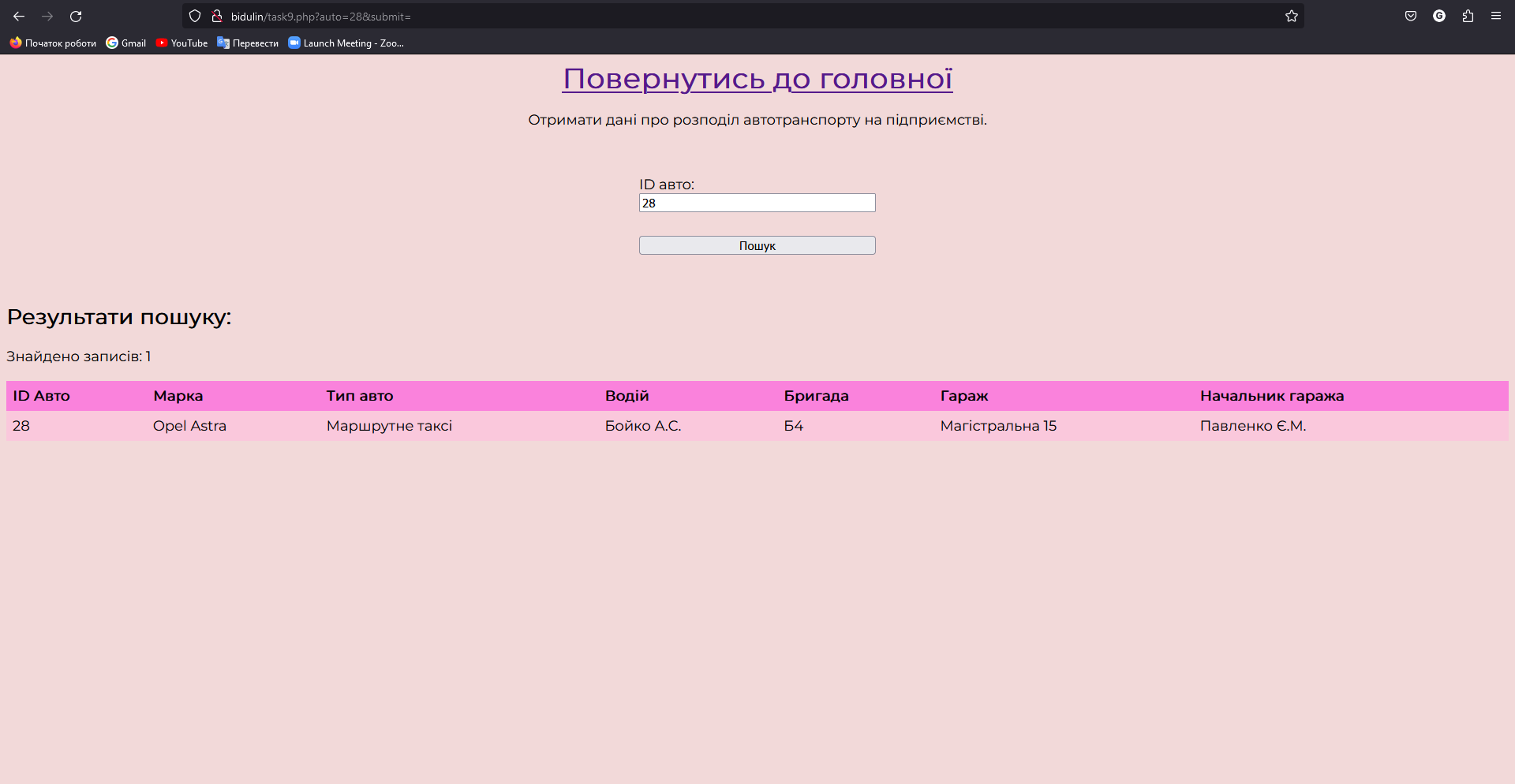


Рисунок 3.10 – Запит 9

1. Отримати відомості про вантажоперевезення, виконаних зазначеною автомашиною за встановлений період.

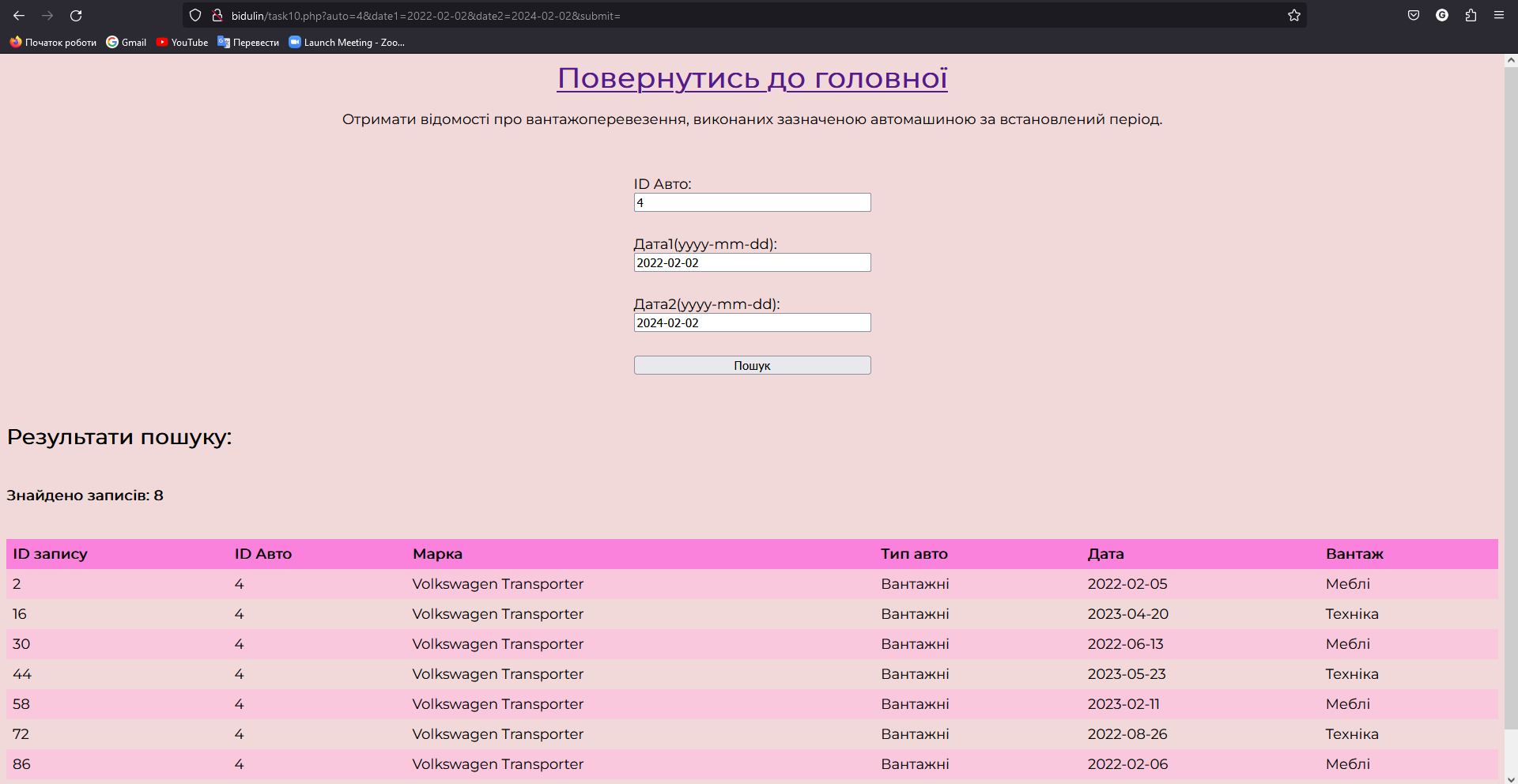
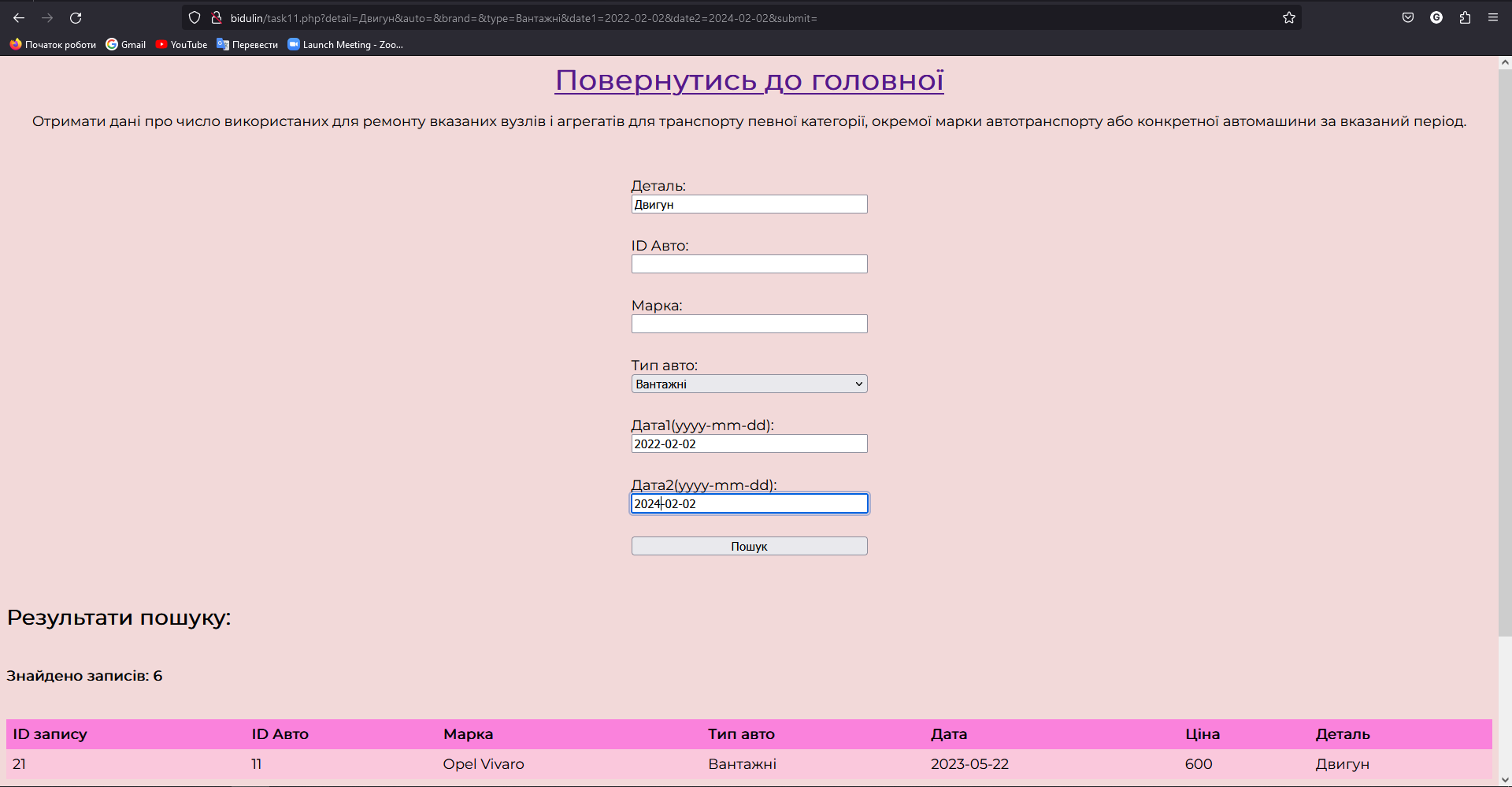


Рисунок 3.11 – Запит 10

1. Отримати дані про число використаних для ремонту вказаних вузлів і агрегатів для транспорту певної категорії, окремої марки автотранспорту або конкретної автомашини за вказаний період.

 Рисунок 3.12 – Запит 11

1. Отримати відомості про отримання та списання автотехніки за вказаний період.

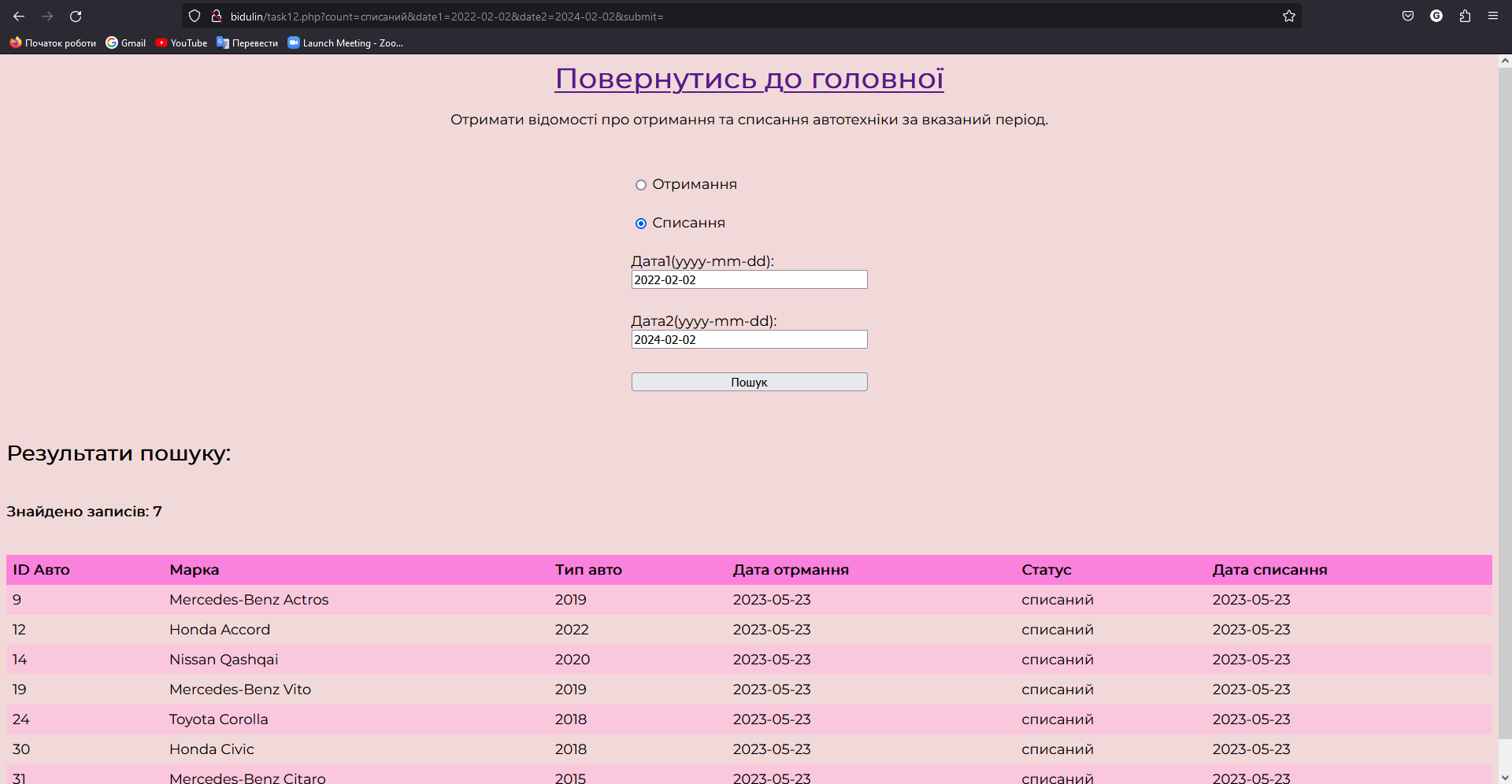


Рисунок 3.13 – Запит 12

1. Отримати склад підлеглих зазначеного бригадира, майстра тощо.

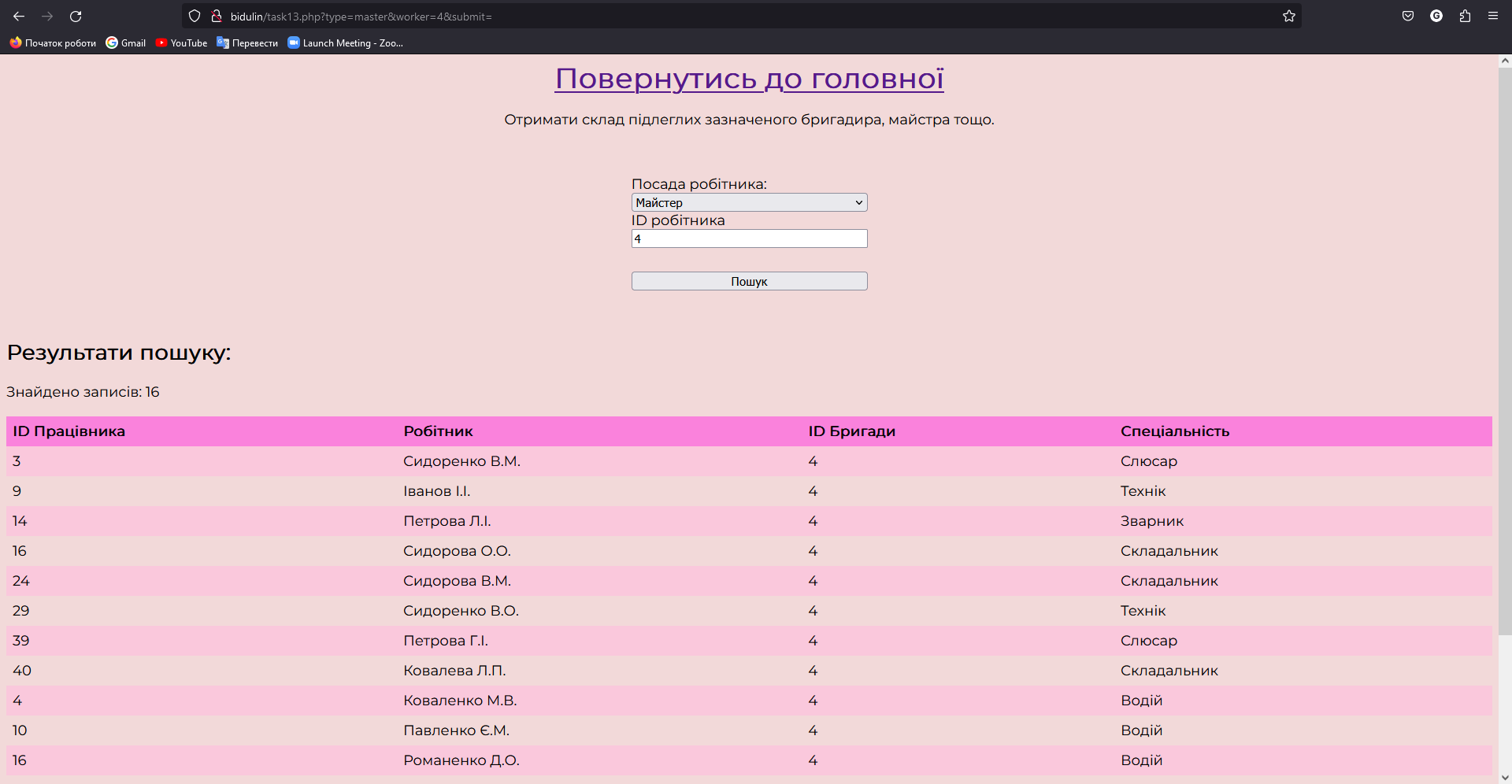


Рисунок 3.14 – Запит 13

1. Отримати дані про роботи, виконані зазначеним фахівцем (зварником, слюсарем тощо) за означений період в цілому і по конкретній автомашині.

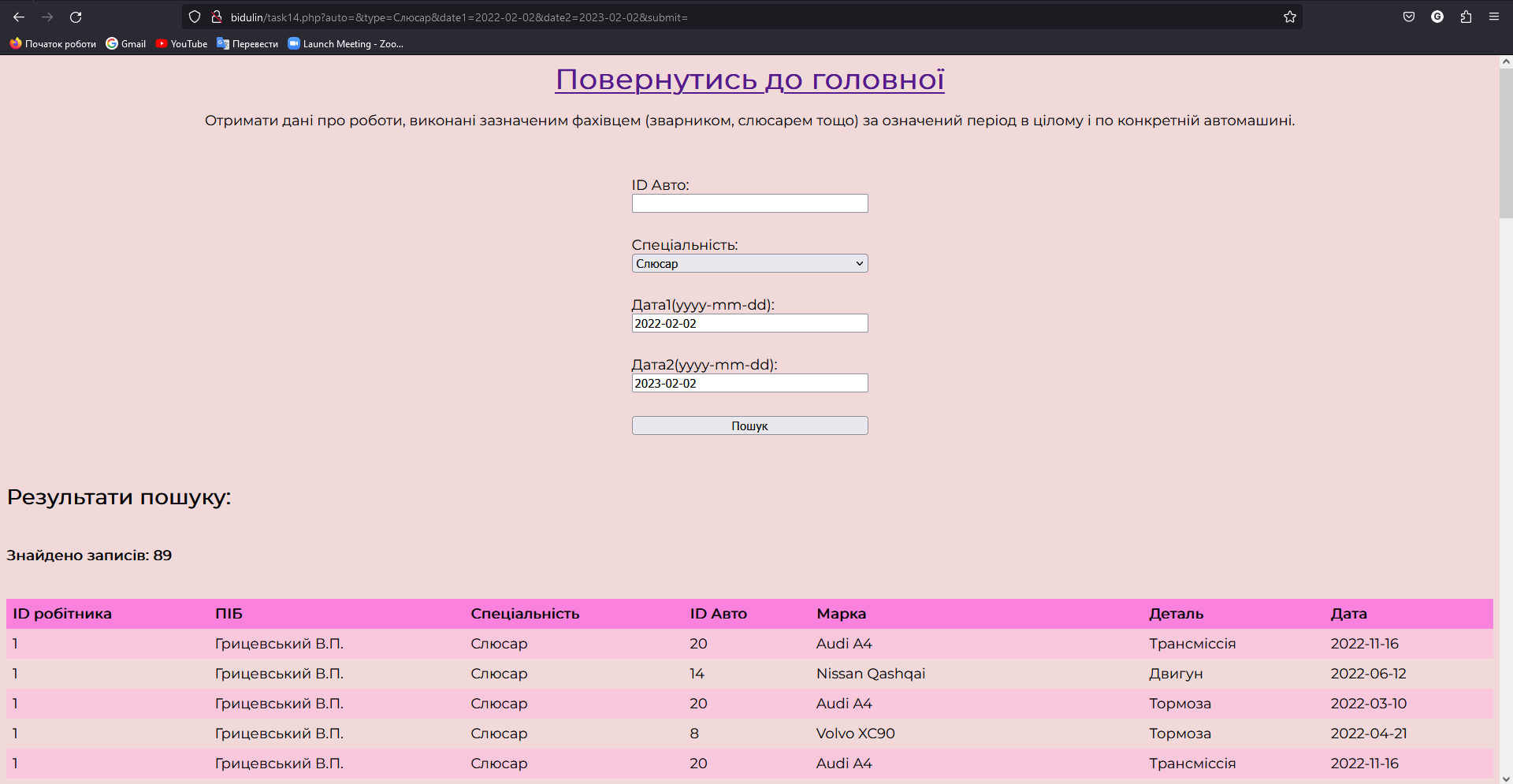


Рисунок 3.15 – Запит 14

**3.5 Опис тестової бази даних**

Тестова база даних складається зі згенерованих даних, тобто всі назви та імена вигадані. У ній знаходиться 35 авто, 46 робітників, 50 водіїв, 6 бригад та 6 гаражів, 9 маршрутів, 100 вантажних перевезень, 80 ремонтних робіт, 300 записів про пробіг.

**РОЗДІЛ 4**

**ВИСНОВОК**

В ході виконання курсової роботи було розроблено інформаційну систему для автопідприємства, робота якої була перевірена на тестових даних. Всі функціональні вимоги до системи були виконані, необхідна функціональність реалізована за рахунок використання запитів. До переваг розробленої системи можна віднести зручний веб-інтерфейс, який значно спрощує роботу з системою для користувачів, що не знайомі з мовою запитів MySQL та середовищем MySQL Workbench. Також, у системи доволі мало технічних вимог. Перспективами розвитку такої системи є додавання можливості адміністрування БД за допомогою веб-інтерфейсу, оскільки наразі в ньому немає підтримки додавання або видалення даних з таблиць, а також – підтримка віддаленого доступу до системи, оскільки зараз сервер та сама система мають знаходитися на ПК користувача.

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Stackoverflow – Where Developers Learn, Share, & Build Careers [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com/>

2. SQL Tutorial – GeeksForGeeks [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/sql-tutorial/>

3. MySQL Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>

4. Open Server Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://ospanel.io/docs/>

5. PHP Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/docs.php>

6. PHP MySQL Database [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/php/php_mysql_intro.asp>

**ДОДАТКИ**

**Додаток A. Створення та заповнення бази даних**

-- Adminer 4.8.1 MySQL 8.0.31 dump

SET NAMES utf8;

SET time\_zone = '+00:00';

SET foreign\_key\_checks = 0;

SET sql\_mode = 'NO\_AUTO\_VALUE\_ON\_ZERO';

SET NAMES utf8mb4;

CREATE DATABASE `auto` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;

USE `auto`;

DROP TABLE IF EXISTS `brigada`;

CREATE TABLE `brigada` (

`id` int NOT NULL,

`name` varchar(45) DEFAULT NULL,

`brigadir` int DEFAULT NULL,

`master` int DEFAULT NULL,

`garage` int DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `brigadir\_UNIQUE` (`brigadir`),

KEY `fk2\_commander\_idx` (`brigadir`),

KEY `fk2\_master\_idx` (`master`),

KEY `fk2\_director\_idx` (`garage`),

CONSTRAINT `fk2\_commander` FOREIGN KEY (`brigadir`) REFERENCES `staff` (`id`),

CONSTRAINT `fk2\_garage` FOREIGN KEY (`garage`) REFERENCES `garage` (`id`),

CONSTRAINT `fk2\_master` FOREIGN KEY (`master`) REFERENCES `staff` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `brigada` (`id`, `name`, `brigadir`, `master`, `garage`) VALUES

(1, 'Б1', 2, 11, 1),

(2, 'Б2', 1, 10, 2),

(3, 'Б3', 5, 18, 3),

(4, 'Б4', 3, 4, 4),

(5, 'Б5', 7, 13, 5),

(6, 'Б6', 6, 8, 6);

DELIMITER ;;

CREATE TRIGGER `BRIGADA\_BEFORE\_INSERT` BEFORE INSERT ON `brigada` FOR EACH ROW

BEGIN

DECLARE staff\_count INT;

DECLARE isAlredy INT;

SELECT COUNT(\*) INTO staff\_count

FROM staff

INNER JOIN brigada ON staff.brigada = brigada.id

WHERE (staff.id = NEW.brigadir OR staff.id = NEW.master) AND brigada.id = NEW.id;

IF staff\_count = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Некоректний командир, робітник не належить бригаді!';

END IF;

SELECT COUNT(\*) INTO isAlredy

FROM staff

INNER JOIN brigada ON staff.brigada = brigada.id

WHERE (staff.id = NEW.brigadir OR staff.id = NEW.master) AND (

brigada.brigadir = NEW.brigadir OR

brigada.master = NEW.brigadir OR

brigada.brigadir = NEW.master OR

brigada.master = NEW.master

);

IF isAlready > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Некоректний командир, робітник є командиром

іншої бригади!';

END IF;

END;;

CREATE TRIGGER `BRIGADA\_BEFORE\_UPDATE` BEFORE UPDATE ON `brigada` FOR EACH ROW

BEGIN

DECLARE staff\_count INT;

DECLARE isAlredy INT;

SELECT COUNT(\*) INTO staff\_count

FROM staff

INNER JOIN brigada ON staff.brigada = brigada.id

WHERE (staff.id = NEW.brigadir OR staff.id = NEW.master) AND brigada.id = NEW.id;

IF staff\_count = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Некоректний командир, робітник не належить бригаді!';

END IF;

SELECT COUNT(\*) INTO isAlredy

FROM staff

INNER JOIN brigada ON staff.brigada = brigada.id

WHERE (staff.id = NEW.brigadir OR staff.id = NEW.master) AND (

brigada.brigadir = NEW.brigadir OR

brigada.master = NEW.brigadir OR

brigada.brigadir = NEW.master OR

brigada.master = NEW.master

);

IF isAlready > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Некоректний командир, робітник є командиром

іншої бригади!';

END IF;

END;;

DELIMITER ;

DROP TABLE IF EXISTS `driver`;

CREATE TABLE `driver` (

`id` int NOT NULL,

`name` varchar(45) DEFAULT NULL,

`auto` int DEFAULT NULL,

`brigada` int DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk\_brigada\_idx` (`brigada`),

KEY `fk\_auto1\_idx` (`auto`),

CONSTRAINT `fk\_auto1` FOREIGN KEY (`auto`) REFERENCES `transport` (`id`),

CONSTRAINT `fk\_brigada` FOREIGN KEY (`brigada`) REFERENCES `brigada` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `driver` (`id`, `name`, `auto`, `brigada`) VALUES

(1, 'Іванов О.П.', 1, 1),

(2, 'Петров І.М.', 2, 2),

(3, 'Сидоренко А.В.', 3, 3),

(4, 'Коваленко М.В.', 4, 4),

(5, 'Мельник О.І.', 5, 5),

(6, 'Шевченко М.О.', 6, 6),

(7, 'Ковальова С.М.', 7, 1),

(8, 'Бойко А.С.', 8, 2),

(9, 'Ковальчук В.О.', 9, 3),

(10, 'Павленко Є.М.', 10, 4),

(11, 'Григоренко В.Д.', 11, 5),

(12, 'Ткаченко М.Є.', 12, 6),

(13, 'Кравченко Ю.В.', 13, 1),

(14, 'Білоус А.А.', 14, 2),

(15, 'Олійник О.Є.', 15, 3),

(16, 'Романенко Д.О.', 16, 4),

(17, 'Захарчук К.А.', 17, 5),

(18, 'Марченко А.С.', 18, 6),

(19, 'Клименко А.Є.', 19, 1),

(20, 'Морозов М.В.', 20, 2),

(21, 'Іванова О.П.', 21, 3),

(22, 'Петрова І.М.', 22, 4),

(23, 'Сидоренко А.В.', 23, 5),

(24, 'Коваленко М.О.', 24, 6),

(25, 'Мельник І.І.', 25, 1),

(26, 'Шевченко М.О.', 26, 2),

(27, 'Ковальова С.М.', 27, 3),

(28, 'Бойко А.С.', 28, 4),

(29, 'Ковальчук В.О.', 29, 5),

(30, 'Павленко Є.М.', 30, 6),

(31, 'Григоренко В.Д.', 31, 1),

(32, 'Ткаченко М.Є.', 32, 2),

(33, 'Кравченко Ю.В.', 33, 3),

(34, 'Білоус А.А.', 34, 4),

(35, 'Олійник О.Є.', 35, 5),

(36, 'Романенко Д.О.', 1, 6),

(37, 'Захарчук К.А.', 2, 1),

(38, 'Марченко А.С.', 3, 2),

(39, 'Клименко А.Є.', 4, 3),

(40, 'Морозов М.В.', 5, 4),

(41, 'Іванова О.П.', 6, 5),

(42, 'Петрова І.М.', 7, 6),

(43, 'Сидоренко А.В.', 8, 1),

(44, 'Коваленко М.О.', 9, 2),

(45, 'Мельник І.І.', 10, 3),

(46, 'Шевченко М.О.', 11, 4),

(47, 'Ковальова С.М.', 12, 5),

(48, 'Бойко А.С.', 13, 6),

(49, 'Ковальчук В.О.', 14, 1),

(50, 'Павленко Є.М.', 15, 2);

DROP TABLE IF EXISTS `garage`;

CREATE TABLE `garage` (

`id` int NOT NULL,

`address` varchar(45) DEFAULT NULL,

`director` int DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk3\_director\_idx` (`director`),

CONSTRAINT `fk\_director` FOREIGN KEY (`director`) REFERENCES `staff` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `garage` (`id`, `address`, `director`) VALUES

(1, 'Вишнева 12', 42),

(2, 'Грушевського 45', 43),

(3, 'Шевченка 9', 44),

(4, 'Магістральна 15', 45),

(5, 'Васильківська 77', 46),

(6, 'Парижської комуни 11', 41);

DROP TABLE IF EXISTS `race`;

CREATE TABLE `race` (

`id` int NOT NULL,

`auto` int DEFAULT NULL,

`date` date DEFAULT NULL,

`distance` int DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk\_auto2\_idx` (`auto`),

CONSTRAINT `fk\_auto2` FOREIGN KEY (`auto`) REFERENCES `transport` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `race` (`id`, `auto`, `date`, `distance`) VALUES

(1, 25, '2022-07-21', 3),

(2, 16, '2022-04-27', 34),

(3, 10, '2022-12-16', 16),

(4, 21, '2023-03-02', 67),

(5, 30, '2022-08-09', 72),

(6, 30, '2022-02-01', 67),

(7, 8, '2022-03-16', 79),

(8, 23, '2024-02-04', 46),

(9, 23, '2023-12-06', 15),

(10, 10, '2024-04-21', 100),

(11, 3, '2022-11-26', 68),

(12, 9, '2022-05-27', 13),

(13, 5, '2022-07-25', 82),

(14, 13, '2022-10-25', 64),

(15, 6, '2024-03-11', 9),

(16, 25, '2022-06-26', 93),

(17, 2, '2022-12-25', 94),

(18, 16, '2023-01-16', 84),

(19, 31, '2023-12-05', 45),

(20, 29, '2023-11-07', 40),

(21, 24, '2022-05-19', 78),

(22, 15, '2023-11-11', 63),

(23, 28, '2022-03-29', 12),

(24, 10, '2024-04-09', 99),

(25, 3, '2023-01-02', 88),

(26, 5, '2024-04-22', 48),

(27, 17, '2024-04-08', 33),

(28, 27, '2024-02-06', 9),

(29, 28, '2023-08-21', 7),

(30, 9, '2022-03-18', 68),

(31, 4, '2023-04-06', 31),

(32, 34, '2024-02-08', 52),

(33, 34, '2022-07-12', 24),

(34, 19, '2024-03-07', 100),

(35, 8, '2022-04-14', 94),

(36, 12, '2023-12-09', 9),

(37, 35, '2023-08-27', 50),

(38, 14, '2023-01-07', 99),

(39, 24, '2022-11-23', 87),

(40, 8, '2023-03-11', 83),

(41, 23, '2023-11-07', 92),

(42, 9, '2023-03-13', 76),

(43, 10, '2022-04-27', 83),

(44, 26, '2022-05-24', 64),

(45, 25, '2023-05-07', 74),

(46, 35, '2023-08-26', 53),

(47, 19, '2022-03-17', 85),

(48, 34, '2022-07-12', 26),

(49, 22, '2022-10-31', 87),

(50, 11, '2024-02-29', 63),

(51, 14, '2022-03-03', 20),

(52, 27, '2022-06-05', 64),

(53, 23, '2022-10-19', 73),

(54, 22, '2024-03-22', 80),

(55, 6, '2023-01-26', 75),

(56, 14, '2023-09-19', 42),

(57, 33, '2022-11-27', 12),

(58, 16, '2024-02-12', 9),

(59, 28, '2023-07-27', 94),

(60, 24, '2023-07-07', 12),

(61, 23, '2024-03-31', 74),

(62, 31, '2022-05-18', 17),

(63, 13, '2022-08-09', 22),

(64, 12, '2022-01-03', 3),

(65, 4, '2022-12-17', 74),

(66, 16, '2022-03-04', 100),

(67, 27, '2024-01-01', 90),

(68, 34, '2022-02-25', 48),

(69, 7, '2023-03-17', 97),

(70, 12, '2023-09-08', 57),

(71, 25, '2023-12-03', 93),

(72, 8, '2022-10-16', 99),

(73, 33, '2023-10-17', 93),

(74, 14, '2022-04-22', 50),

(75, 4, '2024-05-14', 68),

(76, 14, '2024-02-11', 29),

(77, 28, '2022-01-21', 79),

(78, 31, '2024-04-12', 19),

(79, 2, '2023-09-20', 44),

(80, 1, '2023-11-29', 94),

(81, 10, '2023-05-14', 4),

(82, 16, '2022-05-06', 38),

(83, 16, '2022-04-08', 22),

(84, 27, '2022-03-07', 14),

(85, 17, '2024-04-06', 32),

(86, 27, '2023-11-12', 67),

(87, 34, '2024-01-24', 41),

(88, 16, '2022-02-05', 85),

(89, 3, '2024-02-27', 26),

(90, 21, '2022-04-14', 87),

(91, 34, '2022-05-01', 87),

(92, 32, '2024-01-14', 61),

(93, 16, '2023-01-18', 85),

(94, 33, '2022-04-10', 78),

(95, 18, '2022-08-01', 69),

(96, 25, '2023-01-02', 2),

(97, 29, '2022-03-04', 89),

(98, 8, '2022-12-22', 40),

(99, 28, '2023-08-15', 7),

(100, 11, '2022-09-12', 57),

(101, 34, '2022-02-14', 41),

(102, 32, '2022-07-10', 43),

(103, 17, '2022-04-24', 21),

(104, 22, '2023-04-04', 75),

(105, 6, '2023-05-06', 33),

(106, 34, '2023-11-06', 2),

(107, 26, '2023-08-13', 16),

(108, 26, '2022-06-20', 79),

(109, 13, '2022-12-12', 93),

(110, 16, '2023-01-15', 85),

(111, 33, '2022-04-01', 74),

(112, 13, '2023-05-27', 86),

(113, 19, '2022-03-07', 79),

(114, 25, '2022-05-04', 62),

(115, 23, '2022-10-04', 70),

(116, 19, '2023-04-05', 7),

(117, 27, '2023-06-12', 74),

(118, 31, '2022-06-12', 30),

(119, 32, '2023-06-07', 32),

(120, 27, '2024-01-18', 2),

(121, 18, '2022-12-29', 61),

(122, 28, '2022-04-24', 28),

(123, 35, '2022-05-01', 71),

(124, 5, '2023-04-02', 23),

(125, 21, '2022-06-04', 17),

(126, 11, '2024-05-21', 10),

(127, 18, '2022-05-18', 33),

(128, 5, '2023-08-19', 3),

(129, 4, '2022-11-01', 49),

(130, 14, '2023-03-25', 39),

(131, 13, '2023-09-02', 39),

(132, 30, '2024-05-21', 51),

(133, 19, '2022-03-24', 92),

(134, 10, '2023-07-01', 32),

(135, 25, '2023-04-10', 58),

(136, 11, '2023-09-21', 74),

(137, 18, '2022-09-08', 95),

(138, 31, '2023-02-27', 83),

(139, 25, '2024-04-27', 79),

(140, 1, '2023-08-24', 42),

(141, 1, '2024-02-11', 34),

(142, 1, '2022-04-24', 57),

(143, 16, '2023-04-24', 40),

(144, 11, '2022-11-05', 86),

(145, 8, '2023-04-18', 5),

(146, 21, '2023-11-07', 14),

(147, 12, '2022-09-25', 52),

(148, 23, '2023-06-30', 24),

(149, 11, '2023-10-23', 92),

(150, 10, '2023-08-11', 53),

(151, 21, '2023-01-04', 32),

(152, 11, '2023-06-24', 16),

(153, 33, '2022-04-12', 84),

(154, 29, '2023-07-08', 69),

(155, 20, '2023-07-24', 64),

(156, 8, '2022-04-14', 99),

(157, 20, '2024-01-01', 52),

(158, 3, '2023-12-06', 83),

(159, 25, '2022-01-14', 99),

(160, 31, '2022-12-31', 47),

(161, 3, '2022-01-05', 79),

(162, 32, '2022-05-13', 6),

(163, 30, '2022-01-04', 52),

(164, 20, '2022-08-15', 62),

(165, 11, '2023-07-12', 31),

(166, 22, '2022-05-05', 88),

(167, 34, '2022-06-01', 99),

(168, 14, '2022-02-10', 4),

(169, 2, '2022-02-10', 15),

(170, 20, '2022-12-31', 39),

(171, 24, '2022-06-24', 99),

(172, 12, '2023-09-11', 55),

(173, 21, '2022-09-23', 77),

(174, 32, '2022-06-03', 21),

(175, 18, '2023-12-27', 70),

(176, 35, '2024-01-04', 25),

(177, 26, '2024-01-12', 10),

(178, 33, '2022-11-01', 97),

(179, 28, '2024-04-25', 53),

(180, 26, '2022-03-27', 29),

(181, 4, '2023-09-10', 22),

(182, 33, '2022-01-15', 30),

(183, 15, '2022-05-27', 62),

(184, 21, '2022-02-02', 46),

(185, 6, '2023-01-18', 72),

(186, 9, '2022-02-26', 61),

(187, 29, '2022-10-05', 12),

(188, 22, '2023-09-03', 68),

(189, 11, '2023-01-27', 36),

(190, 15, '2024-05-15', 74),

(191, 25, '2022-08-09', 20),

(192, 8, '2023-02-01', 66),

(193, 32, '2023-04-05', 95),

(194, 6, '2024-02-26', 7),

(195, 22, '2024-03-26', 80),

(196, 7, '2023-03-07', 94),

(197, 8, '2022-06-18', 37),

(198, 9, '2022-05-14', 2),

(199, 22, '2024-05-13', 14),

(200, 25, '2022-03-03', 28),

(201, 6, '2024-02-26', 9),

(202, 25, '2022-09-08', 32),

(203, 26, '2023-07-08', 3),

(204, 8, '2022-01-24', 48),

(205, 11, '2022-02-12', 36),

(206, 23, '2022-06-04', 94),

(207, 5, '2024-01-09', 85),

(208, 25, '2024-04-19', 71),

(209, 23, '2022-04-14', 66),

(210, 32, '2023-04-15', 100),

(211, 14, '2024-02-12', 31),

(212, 30, '2022-10-26', 17),

(213, 28, '2023-01-15', 83),

(214, 29, '2023-05-27', 50),

(215, 26, '2022-06-02', 68),

(216, 29, '2022-04-24', 17),

(217, 16, '2023-08-09', 6),

(218, 10, '2022-06-07', 10),

(219, 33, '2022-11-15', 4),

(220, 3, '2022-09-25', 29),

(221, 18, '2023-08-24', 92),

(222, 18, '2023-11-06', 36),

(223, 17, '2022-09-05', 1),

(224, 7, '2024-01-17', 76),

(225, 8, '2023-11-09', 26),

(226, 34, '2022-02-13', 36),

(227, 22, '2024-05-05', 9),

(228, 18, '2022-06-03', 43),

(229, 21, '2023-06-15', 33),

(230, 28, '2024-03-31', 36),

(231, 35, '2023-11-23', 5),

(232, 30, '2022-01-23', 64),

(233, 4, '2023-07-04', 83),

(234, 9, '2023-09-02', 80),

(235, 30, '2024-02-26', 95),

(236, 1, '2022-08-06', 20),

(237, 9, '2023-06-10', 30),

(238, 24, '2023-02-22', 38),

(239, 15, '2024-05-12', 69),

(240, 16, '2022-07-21', 79),

(241, 8, '2023-10-02', 2),

(242, 32, '2023-01-18', 50),

(243, 6, '2022-08-19', 88),

(244, 21, '2022-09-12', 71),

(245, 24, '2022-07-06', 7),

(246, 25, '2022-08-08', 19),

(247, 7, '2022-12-07', 38),

(248, 26, '2023-02-09', 17),

(249, 15, '2023-06-01', 72),

(250, 29, '2024-01-19', 90),

(251, 31, '2023-09-08', 91),

(252, 15, '2022-09-19', 30),

(253, 20, '2024-03-17', 95),

(254, 34, '2024-03-22', 80),

(255, 6, '2023-02-16', 85),

(256, 30, '2023-06-10', 53),

(257, 29, '2023-02-10', 91),

(258, 5, '2024-03-09', 21),

(259, 10, '2023-09-30', 86),

(260, 3, '2023-11-04', 66),

(261, 35, '2024-03-05', 63),

(262, 14, '2022-03-31', 33),

(263, 12, '2023-06-26', 15),

(264, 31, '2024-03-25', 5),

(265, 15, '2024-03-10', 36),

(266, 2, '2022-04-01', 42),

(267, 27, '2023-04-14', 43),

(268, 18, '2022-06-27', 54),

(269, 3, '2023-11-18', 70),

(270, 5, '2023-03-18', 18),

(271, 13, '2022-08-26', 29),

(272, 21, '2022-04-30', 90),

(273, 3, '2023-06-13', 86),

(274, 17, '2023-12-05', 61),

(275, 22, '2022-06-27', 21),

(276, 15, '2023-02-21', 14),

(277, 8, '2023-09-12', 89),

(278, 12, '2024-02-29', 59),

(279, 8, '2022-08-18', 71),

(280, 26, '2023-05-11', 64),

(281, 16, '2022-10-29', 39),

(282, 32, '2022-11-16', 12),

(283, 17, '2024-04-22', 46),

(284, 13, '2022-12-22', 99),

(285, 25, '2023-03-27', 51),

(286, 35, '2022-11-01', 83),

(287, 4, '2024-05-03', 62),

(288, 6, '2024-01-27', 91),

(289, 34, '2024-05-16', 14),

(290, 25, '2022-03-25', 38),

(291, 21, '2024-01-10', 46),

(292, 26, '2022-09-14', 28),

(293, 18, '2023-07-14', 73),

(294, 25, '2022-10-26', 61),

(295, 35, '2022-03-02', 43),

(296, 33, '2022-10-16', 89),

(297, 16, '2023-05-22', 56),

(298, 2, '2023-04-04', 52),

(299, 35, '2022-12-02', 98),

(300, 25, '2023-05-31', 86);

DROP TABLE IF EXISTS `repair`;

CREATE TABLE `repair` (

`id` int NOT NULL,

`auto` int DEFAULT NULL,

`date` date DEFAULT NULL,

`price` int DEFAULT NULL,

`detail` varchar(45) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk\_auto4\_idx` (`auto`),

CONSTRAINT `fk\_auto4` FOREIGN KEY (`auto`) REFERENCES `transport` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `repair` (`id`, `auto`, `date`, `price`, `detail`) VALUES

(1, 5, '2022-01-15', 800, 'Двигун'),

(2, 12, '2022-02-05', 1200, 'Генератор'),

(3, 20, '2022-03-10', 600, 'Тормоза'),

(4, 7, '2022-04-20', 400, 'Акумулятор'),

(5, 30, '2022-05-05', 900, 'Трансміссія'),

(6, 14, '2022-06-12', 500, 'Двигун'),

(7, 28, '2022-07-18', 700, 'Генератор'),

(8, 3, '2022-08-25', 300, 'Тормоза'),

(9, 9, '2022-09-30', 600, 'Акумулятор'),

(10, 25, '2022-10-10', 800, 'Трансміссія'),

(11, 1, '2022-11-15', 1000, 'Двигун'),

(12, 10, '2022-12-20', 400, 'Генератор'),

(13, 17, '2023-01-05', 600, 'Тормоза'),

(14, 33, '2023-02-10', 1200, 'Акумулятор'),

(15, 22, '2023-03-15', 700, 'Трансміссія'),

(16, 8, '2023-04-20', 400, 'Двигун'),

(17, 35, '2023-05-01', 900, 'Генератор'),

(18, 16, '2023-05-10', 500, 'Тормоза'),

(19, 29, '2023-05-15', 800, 'Акумулятор'),

(20, 4, '2023-05-20', 300, 'Трансміссія'),

(21, 11, '2023-05-22', 600, 'Двигун'),

(22, 21, '2023-05-23', 800, 'Генератор'),

(23, 2, '2023-05-23', 400, 'Тормоза'),

(24, 13, '2023-05-23', 500, 'Акумулятор'),

(25, 6, '2022-01-16', 700, 'Трансміссія'),

(26, 15, '2022-02-06', 1000, 'Двигун'),

(27, 23, '2022-03-11', 500, 'Генератор'),

(28, 8, '2022-04-21', 300, 'Тормоза'),

(29, 31, '2022-05-06', 800, 'Акумулятор'),

(30, 3, '2022-06-13', 400, 'Трансміссія'),

(31, 18, '2022-07-19', 600, 'Двигун'),

(32, 30, '2022-08-26', 300, 'Генератор'),

(33, 5, '2022-09-30', 500, 'Тормоза'),

(34, 12, '2022-10-11', 700, 'Акумулятор'),

(35, 20, '2022-11-16', 1000, 'Трансміссія'),

(36, 7, '2022-12-21', 400, 'Двигун'),

(37, 29, '2023-01-06', 500, 'Генератор'),

(38, 16, '2023-02-11', 700, 'Тормоза'),

(39, 35, '2023-03-16', 300, 'Акумулятор'),

(40, 4, '2023-04-21', 800, 'Трансміссія'),

(41, 11, '2023-05-22', 400, 'Двигун'),

(42, 21, '2023-05-23', 600, 'Генератор'),

(43, 2, '2023-05-23', 300, 'Тормоза'),

(44, 13, '2023-05-23', 500, 'Акумулятор'),

(45, 6, '2022-01-16', 700, 'Трансміссія'),

(46, 15, '2022-02-06', 1000, 'Двигун'),

(47, 23, '2022-03-11', 500, 'Генератор'),

(48, 8, '2022-04-21', 300, 'Тормоза'),

(49, 31, '2022-05-06', 800, 'Акумулятор'),

(50, 3, '2022-06-13', 400, 'Трансміссія'),

(51, 18, '2022-07-19', 600, 'Двигун'),

(52, 30, '2022-08-26', 300, 'Генератор'),

(53, 5, '2022-09-30', 500, 'Тормоза'),

(54, 12, '2022-10-11', 700, 'Акумулятор'),

(55, 20, '2022-11-16', 1000, 'Трансміссія'),

(56, 7, '2022-12-21', 400, 'Двигун'),

(57, 29, '2023-01-06', 500, 'Генератор'),

(58, 16, '2023-02-11', 700, 'Тормоза'),

(59, 35, '2023-03-16', 300, 'Акумулятор'),

(60, 4, '2023-04-21', 800, 'Трансміссія'),

(61, 11, '2023-05-22', 400, 'Двигун'),

(62, 21, '2023-05-23', 600, 'Генератор'),

(63, 2, '2023-05-23', 300, 'Тормоза'),

(64, 13, '2023-05-23', 500, 'Акумулятор'),

(65, 6, '2022-01-16', 700, 'Трансміссія'),

(66, 15, '2022-02-06', 1000, 'Двигун'),

(67, 23, '2022-03-11', 500, 'Генератор'),

(68, 8, '2022-04-21', 300, 'Тормоза'),

(69, 31, '2022-05-06', 800, 'Акумулятор'),

(70, 3, '2022-06-13', 400, 'Трансміссія'),

(71, 18, '2022-07-19', 600, 'Двигун'),

(72, 30, '2022-08-26', 300, 'Генератор'),

(73, 5, '2022-09-30', 500, 'Тормоза'),

(74, 12, '2022-10-11', 700, 'Акумулятор'),

(75, 20, '2022-11-16', 1000, 'Трансміссія'),

(76, 7, '2022-12-21', 400, 'Двигун'),

(77, 29, '2023-01-06', 500, 'Генератор'),

(78, 16, '2023-02-11', 700, 'Тормоза'),

(79, 35, '2023-03-16', 300, 'Акумулятор'),

(80, 4, '2023-04-21', 800, 'Трансміссія');

DROP TABLE IF EXISTS `route`;

CREATE TABLE `route` (

`id` int NOT NULL,

`name` varchar(45) DEFAULT NULL,

`auto` int DEFAULT NULL,

`passangers` int DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk\_auto5\_idx` (`auto`),

CONSTRAINT `fk\_auto5` FOREIGN KEY (`auto`) REFERENCES `transport` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `route` (`id`, `name`, `auto`, `passangers`) VALUES

(1, 'Олімпійська-Героїв України', 31, 50),

(2, 'Майдан Незалежності-Площа Льва Толстого', 33, 40),

(3, 'Хрещатик-Либідська', 34, 30),

(4, 'Вокзальна-Дарниця', 32, 25),

(5, 'Льва Толстого-Виставковий центр', 35, 50),

(6, 'Майдан Андріївський-Михайлівська', 8, 35),

(7, 'Контрактова площа-Поштова площа', 14, 40),

(8, 'Південний вокзал-Академмістечко', 28, 30),

(9, 'Льва Толстого-КПІ', 30, 45);

DELIMITER ;;

CREATE TRIGGER `ROUTE\_BEFORE\_INSERT` BEFORE INSERT ON `route` FOR EACH ROW

BEGIN

DECLARE tr\_type INT;

SELECT COUNT(\*) INTO tr\_type FROM transport

WHERE transport.id = NEW.auto AND (transport.category = 'Автобус' OR transport.category = 'Маршрутне таксі');

IF tr\_type = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Не пасажирський транспорт';

END IF;

END;;

CREATE TRIGGER `ROUTE\_BEFORE\_UPDATE` BEFORE UPDATE ON `route` FOR EACH ROW

BEGIN

DECLARE tr\_type INT;

SELECT COUNT(\*) INTO tr\_type FROM transport

WHERE transport.id = NEW.auto AND (transport.category = 'Автобус' OR transport.category = 'Маршрутне таксі');

IF tr\_type = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Не пасажирський транспорт';

END IF;

END;;

CREATE TRIGGER `route\_AFTER\_UPDATE` AFTER UPDATE ON `route` FOR EACH ROW

BEGIN

DECLARE tr\_type INT;

SELECT COUNT(\*) INTO tr\_type FROM transport

WHERE transport.id = NEW.auto AND (transport.category = 'Автобус' OR transport.category = 'Маршрутне таксі');

IF tr\_type = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Не пасажирський транспорт';

END IF;

END;;

DELIMITER ;

DROP TABLE IF EXISTS `staff`;

CREATE TABLE `staff` (

`id` int NOT NULL,

`name` varchar(45) DEFAULT NULL,

`speciality` varchar(45) DEFAULT NULL,

`brigada` int DEFAULT NULL,

`start\_date` date DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk\_brigada\_idx` (`brigada`),

CONSTRAINT `fk\_brigada11` FOREIGN KEY (`brigada`) REFERENCES `brigada` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `staff` (`id`, `name`, `speciality`, `brigada`, `start\_date`) VALUES

(1, 'Грицевський В.П.', 'Слюсар', 2, '2023-05-23'),

(2, 'Коваленко О.П.', 'Зварник', 1, NULL),

(3, 'Сидоренко В.М.', 'Слюсар', 4, '2023-05-23'),

(4, 'Василенко М.О.', 'Складальник', 4, NULL),

(5, 'Петрова Г.І.', 'Технік', 3, '2023-05-23'),

(6, 'Ковалева Л.П.', 'Зварник', 6, NULL),

(7, 'Сидорова В.М.', 'Слюсар', 5, '2023-05-23'),

(8, 'Васильчук О.О.', 'Складальник', 6, NULL),

(9, 'Іванов І.І.', 'Технік', 4, '2023-05-23'),

(10, 'Петренко О.В.', 'Зварник', 2, NULL),

(11, 'Коваленко В.П.', 'Слюсар', 1, '2023-05-23'),

(12, 'Сидоренко В.О.', 'Складальник', 1, NULL),

(13, 'Василенко О.М.', 'Технік', 5, '2023-05-23'),

(14, 'Петрова Л.І.', 'Зварник', 4, NULL),

(15, 'Ковалева В.М.', 'Слюсар', 6, '2023-05-23'),

(16, 'Сидорова О.О.', 'Складальник', 4, NULL),

(17, 'Іванов М.І.', 'Технік', 1, '2023-05-23'),

(18, 'Петренко І.В.', 'Зварник', 3, NULL),

(19, 'Коваленко О.П.', 'Слюсар', 3, '2023-05-23'),

(20, 'Сидоренко В.М.', 'Складальник', 5, NULL),

(21, 'Василенко М.О.', 'Технік', 2, '2023-05-23'),

(22, 'Петрова Г.І.', 'Зварник', 6, NULL),

(23, 'Ковалева Л.П.', 'Слюсар', 1, '2023-05-23'),

(24, 'Сидорова В.М.', 'Складальник', 4, NULL),

(25, 'Васильчук О.О.', 'Технік', 5, '2023-05-23'),

(26, 'Іванов І.І.', 'Зварник', 5, NULL),

(27, 'Петренко О.В.', 'Слюсар', 6, '2023-05-23'),

(28, 'Коваленко В.П.', 'Складальник', 2, NULL),

(29, 'Сидоренко В.О.', 'Технік', 4, '2023-05-23'),

(30, 'Василенко О.М.', 'Зварник', 6, NULL),

(31, 'Петрова Л.І.', 'Слюсар', 5, '2023-05-23'),

(32, 'Ковалева В.М.', 'Складальник', 2, NULL),

(33, 'Сидорова О.О.', 'Технік', 5, '2023-05-23'),

(34, 'Іванов М.І.', 'Зварник', 5, NULL),

(35, 'Петренко І.В.', 'Слюсар', 1, '2023-05-23'),

(36, 'Коваленко О.П.', 'Складальник', 6, NULL),

(37, 'Сидоренко В.М.', 'Технік', 2, '2023-05-23'),

(38, 'Василенко О.О.', 'Зварник', 2, NULL),

(39, 'Петрова Г.І.', 'Слюсар', 4, '2023-05-23'),

(40, 'Ковалева Л.П.', 'Складальник', 4, NULL),

(41, 'Морозов М.В.', 'Директор', NULL, NULL),

(42, 'Романенко Д.О.', 'Директор', NULL, NULL),

(43, 'Григоренко В.Д', 'Директор', NULL, NULL),

(44, 'Ткаченко М.Є', 'Директор', NULL, NULL),

(45, 'Павленко Є.М.', 'Директор', NULL, NULL),

(46, 'Сидоренко А.В.', 'Директор', NULL, NULL);

DELIMITER ;;

CREATE TRIGGER `STAFF\_BEFORE\_INSERT` BEFORE INSERT ON `staff` FOR EACH ROW

BEGIN

IF NEW.`speciality` != 'Слюсар' OR NEW.`speciality` != 'Технік' THEN

SET NEW.`start\_date` = NULL;

END IF;

END;;

CREATE TRIGGER `STAFF\_BEFORE\_UPDATE` BEFORE UPDATE ON `staff` FOR EACH ROW

BEGIN

IF NEW.`speciality` != 'Слюсар' OR NEW.`speciality` != 'Технік' THEN

SET NEW.`start\_date` = NULL;

END IF;

END;;

DELIMITER ;

DROP TABLE IF EXISTS `transport`;

CREATE TABLE `transport` (

`id` int NOT NULL AUTO\_INCREMENT,

`brand` varchar(45) DEFAULT NULL,

`year` int DEFAULT NULL,

`category` varchar(45) DEFAULT NULL,

`garage` int DEFAULT NULL,

`capacity` int DEFAULT NULL,

`carry` varchar(45) DEFAULT NULL,

`start\_date` date DEFAULT NULL,

`passed` varchar(45) DEFAULT NULL,

`finish\_date` date DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk\_garage\_idx` (`garage`),

CONSTRAINT `fk\_garage` FOREIGN KEY (`garage`) REFERENCES `garage` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `transport` (`id`, `brand`, `year`, `category`, `garage`, `capacity`, `carry`, `start\_date`, `passed`, `finish\_date`) VALUES

(1, 'Mercedes', 2003, 'Легковий', 1, NULL, NULL, '2023-05-23', 'активний', NULL),

(2, 'Ford Focus', 2019, 'Легкові', 2, NULL, NULL, '2023-05-23', 'активний', NULL),

(3, 'Mercedes-Benz Sprinter', 2020, 'Вантажні', 3, NULL, '2000', '2023-05-23', 'активний', NULL),

(4, 'Volkswagen Transporter', 2017, 'Вантажні', 4, NULL, '1500', '2023-05-23', 'активний', NULL),

(5, 'Hyundai Sonata', 2016, 'Таксі', 5, NULL, NULL, '2023-05-23', 'активний', NULL),

(6, 'Renault Trafic', 2021, 'Вантажні', 6, NULL, '1200', '2023-05-23', 'активний', NULL),

(7, 'Skoda Octavia', 2018, 'Легкові', 1, NULL, NULL, '2023-05-23', 'активний', NULL),

(8, 'Volvo XC90', 2020, 'Маршрутне таксі', 2, 7, NULL, '2023-05-23', 'активний', NULL),

(9, 'Mercedes-Benz Actros', 2019, 'Вантажні', 3, NULL, '2500', '2023-05-23', 'списаний', '2023-05-23'),

(10, 'BMW X5', 2017, 'Легкові', 4, NULL, NULL, '2023-05-23', 'активний', NULL),

(11, 'Opel Vivaro', 2018, 'Вантажні', 5, NULL, '1100', '2023-05-23', 'активний', NULL),

(12, 'Honda Accord', 2022, 'Таксі', 6, NULL, NULL, '2023-05-23', 'списаний', '2023-05-23'),

(13, 'Toyota Hilux', 2019, 'Вантажні', 1, NULL, '1800', '2023-05-23', 'активний', NULL),

(14, 'Nissan Qashqai', 2020, 'Маршрутне таксі', 2, 5, NULL, '2023-05-23', 'списаний', '2023-05-23'),

(15, 'Ford Transit', 2018, 'Вантажні', 3, NULL, '2000', '2023-05-23', 'активний', NULL),

(16, 'Volkswagen Golf', 2017, 'Таксі', 4, NULL, NULL, '2023-05-23', 'активний', NULL),

(17, 'Peugeot Boxer', 2021, 'Вантажні', 5, NULL, '1600', '2023-05-23', 'активний', NULL),

(18, 'Chevrolet Cruze', 2016, 'Легкові', 6, NULL, NULL, '2023-05-23', 'активний', NULL),

(19, 'Mercedes-Benz Vito', 2019, 'Вантажні', 1, NULL, '1300', '2023-05-23', 'списаний', '2023-05-23'),

(20, 'Audi A4', 2020, 'Легкові', 2, NULL, NULL, '2023-05-23', 'активний', NULL),

(21, 'Renault Master', 2018, 'Вантажні', 3, NULL, '2200', '2023-05-23', 'активний', NULL),

(22, 'Hyundai Tucson', 2017, 'Таксі', 4, NULL, NULL, '2023-05-23', 'активний', NULL),

(23, 'Volkswagen Caddy', 2021, 'Вантажні', 5, NULL, '1000', '2023-05-23', 'активний', NULL),

(24, 'Toyota Corolla', 2018, 'Легкові', 6, NULL, NULL, '2023-05-23', 'списаний', '2023-05-23'),

(25, 'Ford Ranger', 2020, 'Вантажні', 1, NULL, '1700', '2023-05-23', 'активний', NULL),

(26, 'BMW 3 Series', 2019, 'Таксі', 2, NULL, NULL, '2023-05-23', 'активний', NULL),

(27, 'Mercedes-Benz Sprinter', 2017, 'Вантажні', 3, NULL, '2500', '2023-05-23', 'активний', NULL),

(28, 'Opel Astra', 2016, 'Маршрутне таксі', 4, 5, NULL, '2023-05-23', 'активний', NULL),

(29, 'Renault Kangoo', 2021, 'Вантажні', 5, NULL, '1100', '2023-05-23', 'активний', NULL),

(30, 'Honda Civic', 2018, 'Маршрутне таксі', 6, 5, NULL, '2023-05-23', 'списаний', '2023-05-23'),

(31, 'Mercedes-Benz Citaro', 2015, 'Автобус', 1, 50, NULL, '2023-05-23', 'списаний', '2023-05-23'),

(32, 'MAN Lions City', 2018, 'Автобус', 2, 40, NULL, '2023-05-23', 'активний', NULL),

(33, 'Volvo 9700', 2017, 'Автобус', 3, 55, NULL, '2023-05-23', 'активний', NULL),

(34, 'Scania Interlink', 2016, 'Автобус', 4, 45, NULL, '2023-05-23', 'активний', NULL),

(35, 'Iveco Crossway', 2019, 'Автобус', 5, 35, NULL, '2023-05-23', 'активний', NULL);

DELIMITER ;;

CREATE TRIGGER `transport\_BEFORE\_INSERT2` BEFORE INSERT ON `transport` FOR EACH ROW

BEGIN

IF NEW.`category` = 'Вантажні' THEN

SET NEW.`capacity` = NULL;

ELSEIF NEW.`category` = 'Автобус' OR NEW.`category` = 'Маршрутне таксі' THEN

SET NEW.`carry` = NULL;

ELSE

SET NEW.`capacity` = NULL;

SET NEW.`carry` = NULL;

END IF;

END;;

CREATE TRIGGER `transport\_BEFORE\_INSERT` BEFORE INSERT ON `transport` FOR EACH ROW

BEGIN

IF NEW.passed = 'списаний' THEN

SET NEW.finish\_date = CURDATE();

else SET NEW.finish\_date = null;

END IF;

END;;

CREATE TRIGGER `transport\_BEFORE\_UPDATE` BEFORE UPDATE ON `transport` FOR EACH ROW

BEGIN

IF NEW.passed = 'списаний' THEN

SET NEW.finish\_date = CURDATE();

else SET NEW.finish\_date = null;

END IF;

END;;

CREATE TRIGGER `transport\_BEFORE\_UPDATE2` BEFORE UPDATE ON `transport` FOR EACH ROW

BEGIN

IF NEW.`category` = 'Вантажні' THEN

SET NEW.`capacity` = NULL;

ELSEIF NEW.`category` = 'Автобус' OR NEW.`category` = 'Маршрутне таксі' THEN

SET NEW.`carry` = NULL;

ELSE

SET NEW.`capacity` = NULL;

SET NEW.`carry` = NULL;

END IF;

END;;

DELIMITER ;

DROP TABLE IF EXISTS `vantaj`;

CREATE TABLE `vantaj` (

`id` int NOT NULL,

`auto` int DEFAULT NULL,

`type\_vantaj` varchar(45) DEFAULT NULL,

`date` date DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk\_auto3\_idx` (`auto`),

CONSTRAINT `fk\_auto3` FOREIGN KEY (`auto`) REFERENCES `transport` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

INSERT INTO `vantaj` (`id`, `auto`, `type\_vantaj`, `date`) VALUES

(1, 3, 'Будівельні матеріали', '2022-01-15'),

(2, 4, 'Меблі', '2022-02-05'),

(3, 6, 'Речі', '2022-03-10'),

(4, 9, 'Техніка', '2022-04-20'),

(5, 11, 'Будівельні матеріали', '2022-05-05'),

(6, 13, 'Меблі', '2022-06-12'),

(7, 15, 'Речі', '2022-07-18'),

(8, 17, 'Техніка', '2022-08-25'),

(9, 19, 'Будівельні матеріали', '2022-09-30'),

(10, 21, 'Меблі', '2022-10-10'),

(11, 23, 'Речі', '2022-11-15'),

(12, 25, 'Техніка', '2022-12-20'),

(13, 27, 'Будівельні матеріали', '2023-01-05'),

(14, 29, 'Меблі', '2023-02-10'),

(15, 3, 'Речі', '2023-03-15'),

(16, 4, 'Техніка', '2023-04-20'),

(17, 6, 'Будівельні матеріали', '2023-05-01'),

(18, 9, 'Меблі', '2023-05-10'),

(19, 11, 'Речі', '2023-05-15'),

(20, 13, 'Техніка', '2023-05-20'),

(21, 15, 'Будівельні матеріали', '2023-05-22'),

(22, 17, 'Меблі', '2023-05-23'),

(23, 19, 'Речі', '2023-05-23'),

(24, 21, 'Техніка', '2023-05-23'),

(25, 23, 'Будівельні матеріали', '2022-01-16'),

(26, 25, 'Меблі', '2022-02-06'),

(27, 27, 'Речі', '2022-03-11'),

(28, 29, 'Техніка', '2022-04-21'),

(29, 3, 'Будівельні матеріали', '2022-05-06'),

(30, 4, 'Меблі', '2022-06-13'),

(31, 6, 'Речі', '2022-07-19'),

(32, 9, 'Техніка', '2022-08-26'),

(33, 11, 'Будівельні матеріали', '2022-09-30'),

(34, 13, 'Меблі', '2022-10-11'),

(35, 15, 'Речі', '2022-11-16'),

(36, 17, 'Техніка', '2022-12-21'),

(37, 19, 'Будівельні матеріали', '2023-01-06'),

(38, 21, 'Меблі', '2023-02-11'),

(39, 23, 'Речі', '2023-03-16'),

(40, 25, 'Техніка', '2023-04-21'),

(41, 27, 'Будівельні матеріали', '2023-05-22'),

(42, 29, 'Меблі', '2023-05-23'),

(43, 3, 'Речі', '2023-05-23'),

(44, 4, 'Техніка', '2023-05-23'),

(45, 6, 'Будівельні матеріали', '2022-01-16'),

(46, 9, 'Меблі', '2022-02-06'),

(47, 11, 'Речі', '2022-03-11'),

(48, 13, 'Техніка', '2022-04-21'),

(49, 15, 'Будівельні матеріали', '2022-05-06'),

(50, 17, 'Меблі', '2022-06-13'),

(51, 19, 'Речі', '2022-07-19'),

(52, 21, 'Техніка', '2022-08-26'),

(53, 23, 'Будівельні матеріали', '2022-09-30'),

(54, 25, 'Меблі', '2022-10-11'),

(55, 27, 'Речі', '2022-11-16'),

(56, 29, 'Техніка', '2022-12-21'),

(57, 3, 'Будівельні матеріали', '2023-01-06'),

(58, 4, 'Меблі', '2023-02-11'),

(59, 6, 'Речі', '2023-03-16'),

(60, 9, 'Техніка', '2023-04-21'),

(61, 11, 'Будівельні матеріали', '2023-05-22'),

(62, 13, 'Меблі', '2023-05-23'),

(63, 15, 'Речі', '2023-05-23'),

(64, 17, 'Техніка', '2023-05-23'),

(65, 19, 'Будівельні матеріали', '2022-01-16'),

(66, 21, 'Меблі', '2022-02-06'),

(67, 23, 'Речі', '2022-03-11'),

(68, 25, 'Техніка', '2022-04-21'),

(69, 27, 'Будівельні матеріали', '2022-05-06'),

(70, 29, 'Меблі', '2022-06-13'),

(71, 3, 'Речі', '2022-07-19'),

(72, 4, 'Техніка', '2022-08-26'),

(73, 6, 'Будівельні матеріали', '2022-09-30'),

(74, 9, 'Меблі', '2022-10-11'),

(75, 11, 'Речі', '2022-11-16'),

(76, 13, 'Техніка', '2022-12-21'),

(77, 15, 'Будівельні матеріали', '2023-01-06'),

(78, 17, 'Меблі', '2023-02-11'),

(79, 19, 'Речі', '2023-03-16'),

(80, 21, 'Техніка', '2023-04-21'),

(81, 23, 'Будівельні матеріали', '2023-05-22'),

(82, 25, 'Меблі', '2023-05-23'),

(83, 27, 'Речі', '2023-05-23'),

(84, 29, 'Техніка', '2023-05-23'),

(85, 3, 'Будівельні матеріали', '2022-01-16'),

(86, 4, 'Меблі', '2022-02-06'),

(87, 6, 'Речі', '2022-03-11'),

(88, 9, 'Техніка', '2022-04-21'),

(89, 11, 'Будівельні матеріали', '2022-05-06'),

(90, 13, 'Меблі', '2022-06-13'),

(91, 15, 'Речі', '2022-07-19'),

(92, 17, 'Техніка', '2022-08-26'),

(93, 19, 'Будівельні матеріали', '2022-09-30'),

(94, 21, 'Меблі', '2022-10-11'),

(95, 23, 'Речі', '2022-11-16'),

(96, 25, 'Техніка', '2022-12-21'),

(97, 27, 'Будівельні матеріали', '2023-01-06'),

(98, 29, 'Меблі', '2023-02-11'),

(99, 3, 'Речі', '2023-03-16'),

(100, 4, 'Техніка', '2023-04-21');

DELIMITER ;;

CREATE TRIGGER `vantaj\_BEFORE\_INSERT` BEFORE INSERT ON `vantaj` FOR EACH ROW

BEGIN

DECLARE tr\_type INT;

SELECT COUNT(\*) INTO tr\_type FROM transport

WHERE transport.id = NEW.auto AND transport.category = 'Вантажні' ;

IF tr\_type = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Не вантажний транспорт';

END IF;

END;;

CREATE TRIGGER `vantaj\_BEFORE\_UPDATE` BEFORE UPDATE ON `vantaj` FOR EACH ROW

BEGIN

DECLARE tr\_type INT;

SELECT COUNT(\*) INTO tr\_type FROM transport

WHERE transport.id = NEW.auto AND transport.category = 'Вантажні' ;

IF tr\_type = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Не вантажний транспорт';

END IF;

END;;

DELIMITER ;

-- 2023-05-27 09:58:33

**Додаток Б. Реалізація SQL запитів у веб інтерфейсі**

**task1.php:**

<?php

      require\_once 'connection.php';

      $link = mysqli\_connect($host, $user, $password, $database)

        or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    // отримуємо значення полів форми

    $garage = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['garage']));

    if(empty($garage)){

    // виконуємо запит до бази даних, щоб знайти частини, які належать вказаній армії та бригаді

    $sql = "SELECT transport.id, transport.brand, transport.year,transport.category, garage.address, transport.start\_date, transport.passed,

    transport.finish\_date

    FROM transport

    JOIN garage ON transport.garage = garage.id";}

    else{

      $sql = "SELECT transport.id, transport.brand, transport.year,transport.category, garage.address, transport.start\_date, transport.passed,

      transport.finish\_date

      FROM transport

      JOIN garage ON transport.garage = garage.id

      WHERE transport.garage = $garage";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "Знайдено записів: " . $rows;

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID</th><th>Марка</th><th>Рік</th><th>Категорія</th><th>Гарадж</th><th>Дата придбання</th><th>Статус</th><th>Дата списання</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 8; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task2.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    // отримуємо значення полів форми

    $auto = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['auto']));

    if (empty($auto)) {

      // виконуємо запит до бази даних, щоб знайти частини, які належать вказаній армії та бригаді

      $sql = "SELECT driver.id, driver.name, transport.id, transport.brand

    FROM driver

    JOIN transport ON driver.auto = transport.id

    ORDER BY driver.id";

    } else {

      $sql = "SELECT driver.id, driver.name, transport.id, transport.brand

      FROM driver

      JOIN transport ON driver.auto = transport.id

      WHERE driver.auto = $auto

      ORDER BY driver.id";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "Знайдено записів: " . $rows;

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID Водія</th><th>Ім'я</th><th>ID Авто</th><th>Марка</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 4; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task3.php:**

<?php

      require\_once 'connection.php';

      $link = mysqli\_connect($host, $user, $password, $database)

        or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    // отримуємо значення полів форми

    $type = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['type']));

    $sql = "SELECT driver.id, driver.name, transport.id, transport.brand, transport.category

      FROM driver

      JOIN transport ON driver.auto = transport.id

      WHERE transport.category = '$type'

      ORDER BY driver.id";

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "Знайдено записів: " . $rows;

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID Водія</th><th>Ім'я</th><th>ID Авто</th><th>Марка</th><th>Тип авто</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 5; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task4.php:**

<?php

      require\_once 'connection.php';

      $link = mysqli\_connect($host, $user, $password, $database)

        or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    // отримуємо значення полів форми

    $type = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['type']));

    if(empty($type)){

      $sql = "SELECT route.id, route.name, route.passangers,transport.brand, transport.capacity

      FROM route

      JOIN transport ON route.auto = transport.id

      ORDER BY route.id";

    }

    else{

    $sql = "SELECT route.id, route.name, route.passangers,transport.brand, transport.capacity

    FROM route

    JOIN transport ON route.auto = transport.id

    WHERE transport.category = '$type'

    ORDER BY route.id";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "Знайдено записів: " . $rows;

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID</th><th>Назва</th><th>К-ть пасажирів</th><th>Марка</th><th>Місткість авто</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 5; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task5.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    $auto = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['auto']));

    $type = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['type']));

    $day = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['day']));

    $month = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['month']));

    $year = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['year']));

    if(empty($auto)){

      $sql = "SELECT r.id, t.id, t.brand,t.category, r.date, r.distance

      FROM race r

      JOIN transport t ON r.auto = t.id

      WHERE t.category = '$type'

        AND YEAR(r.date) = '$year'";

      if (!empty($month)) {

          $sql .= " AND MONTH(r.date) = '$month'";

      }

      elseif (!empty($day)) {

          $sql .= " AND DAY(r.date) = '$day'";

      }

      $sqlSum = "SELECT SUM(r.distance) AS total\_distance

      FROM race r

      JOIN transport t ON r.auto = t.id

      WHERE t.category = '$type'

        AND YEAR(r.date) = '$year'";

      if (!empty($month)) {

          $sql .= " AND MONTH(r.date) = '$month'";

      }

      elseif (!empty($day)) {

          $sql .= " AND DAY(r.date) = '$day'";

      }

    }

    else{

      $sql = "SELECT r.id, t.id, t.brand,t.category, r.date, r.distance

      FROM race r

      JOIN transport t ON r.auto = t.id

      WHERE t.id = $auto

        AND YEAR(r.date) = $year";

      if (!empty($month)) {

          $sql .= " AND MONTH(r.date) = $month";

      }

      if (!empty($day)) {

          $sql .= " AND DAY(r.date) = '$day'";

      }

      $sqlSum= "SELECT SUM(r.distance) AS total\_distance

      FROM race r

      JOIN transport t ON r.auto = t.id

      WHERE t.id = $auto

        AND YEAR(r.date) = $year";

      if (!empty($month)) {

          $sql .= " AND MONTH(r.date) = $month";

      }

      if (!empty($day)) {

          $sql .= " AND DAY(r.date) = '$day'";

      }

    }

    $result = mysqli\_query($link, $sql);

    $distance = mysqli\_query($link, $sqlSum);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "<h4>Знайдено записів: " . $rows . "</h4>";

    echo "<h4>Загальний пробіг: " . mysqli\_fetch\_row($distance)[0] . "км </h4>";

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID запису</th><th>ID Авто</th><th>Марка</th><th>Тип авто</th><th>Дата</th><th>Дистанція</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 6; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task6.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    $auto = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['auto']));

    $brand = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['brand']));

    $type = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['type']));

    $date1 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date1']));

    $date2 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date2']));

    if (!empty($auto)) {

      $sql = "SELECT r.id, t.id, t.brand,t.category, r.date, r.price, r.detail

      FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE t.id = $auto

        AND r.date BETWEEN '$date1' AND '$date2'";

      $sqlSum = "SELECT SUM(r.price) AS total

      FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE t.id = $auto

        AND r.date BETWEEN '$date1' AND '$date2'";

    } elseif (!empty($brand)) {

      $sql = "SELECT r.id, t.id, t.brand,t.category, r.date, r.price, r.detail

      FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE t.brand = '$brand'

        AND r.date BETWEEN '$date1' AND '$date2'";

      $sqlSum = "SELECT SUM(r.price) AS total

       FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE t.brand = '$brand'

        AND r.date BETWEEN '$date1' AND '$date2'";

    } else {

      $sql = "SELECT r.id, t.id, t.brand,t.category, r.date, r.price, r.detail

      FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE t.category = '$type'

        AND r.date BETWEEN '$date1' AND '$date2'";

      $sqlSum = "SELECT SUM(r.price) AS total

      FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE t.category = '$type'

        AND r.date BETWEEN '$date1' AND '$date2'";

    }

    $result = mysqli\_query($link, $sql);

    $price = mysqli\_query($link, $sqlSum);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "<h4>Знайдено записів: " . $rows . "</h4>";

    echo "<h4>Загальна вартість: " . mysqli\_fetch\_row($price)[0] . "грн </h4>";

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID запису</th><th>ID Авто</th><th>Марка</th><th>Тип авто</th><th>Дата</th><th>Ціна</th><th>Деталь</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 7; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task7.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    // отримуємо значення полів форми

    $worker = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['worker']));

    if (empty($worker)) {

      $sql = "SELECT s1.id, s1.name,s2.id, s2.name,s3.id, s3.name,s4.id, s4.name

      FROM staff s1

      LEFT JOIN brigada b1 ON b1.id = s1.brigada

      LEFT JOIN staff s2 ON s2.id = b1.brigadir

      LEFT JOIN staff s3 ON s3.id = b1.master

      LEFT JOIN garage b2 ON b2.id = b1.garage

      LEFT JOIN staff s4 ON s4.id = b2.director";

    } else {

      $sql = "SELECT s1.id, s1.name,s2.id, s2.name,s3.id, s3.name,s4.id, s4.name

      FROM staff s1

      LEFT JOIN brigada b1 ON b1.id = s1.brigada

      LEFT JOIN staff s2 ON s2.id = b1.brigadir

      LEFT JOIN staff s3 ON s3.id = b1.master

      LEFT JOIN garage b2 ON b2.id = b1.garage

      LEFT JOIN staff s4 ON s4.id = b2.director

      WHERE s1.id = $worker";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "Знайдено записів: " . $rows;

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID Працівника</th><th>Робітник</th><th>ID Бригадира</th><th>Бригадир</th><th>ID Майстра</th><th>Майстер</th><th>ID Начальника</th><th>Начальник</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 8; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task8.php:**

  <?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    // отримуємо значення полів форми

    $type = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['type']));

    if (empty($type)) {

      $sql = "SELECT transport.id, transport.brand,transport.category, garage.address, staff.name

      FROM transport

      JOIN garage ON transport.garage = garage.id

      JOIN staff ON garage.director = staff.id";

    } else {

      $sql = "SELECT transport.id, transport.brand,transport.category, garage.address, staff.name

      FROM transport

      JOIN garage ON transport.garage = garage.id

      JOIN staff ON garage.director = staff.id

      WHERE transport.category = '$type'";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "Знайдено записів: " . $rows;

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID Авто</th><th>Марка</th><th>Тип авто</th><th>Цех</th><th>Начальник цеху</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 5; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task9.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    // отримуємо значення полів форми

    $auto = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['auto']));

    if (empty($auto)) {

      $sql = "SELECT transport.id, transport.brand,transport.category ,driver.name, brigada.name, garage.address, staff.name

      FROM transport

      JOIN garage ON transport.garage = garage.id

      JOIN staff ON garage.director = staff.id

      JOIN driver ON transport.id = driver.auto

      JOIN brigada ON driver.brigada =  brigada.id";

    } else {

      $sql = "SELECT transport.id, transport.brand,transport.category ,driver.name, brigada.name, garage.address, staff.name

      FROM transport

      JOIN garage ON transport.garage = garage.id

      JOIN staff ON garage.director = staff.id

      JOIN driver ON transport.id = driver.auto

      JOIN brigada ON driver.brigada =  brigada.id

      WHERE transport.id = $auto";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "Знайдено записів: " . $rows;

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID Авто</th><th>Марка</th><th>Тип авто</th><th>Водій</th><th>Бригада</th><th>Гараж</th><th>Начальник гаража</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 7; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task10.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    $auto = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['auto']));

    $date1 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date1']));

    $date2 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date2']));

    $sql = "SELECT v.id, t.id, t.brand,t.category, v.date, v.type\_vantaj

      FROM vantaj v

      JOIN transport t ON v.auto = t.id

      WHERE t.id = $auto

        AND v.date BETWEEN '$date1' AND '$date2'";

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "<h4>Знайдено записів: " . $rows . "</h4>";

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID запису</th><th>ID Авто</th><th>Марка</th><th>Тип авто</th><th>Дата</th><th>Вантаж</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 6; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  } else {

    $sql = "SELECT transport.id, transport.brand, transport.category, transport.carry FROM transport

    WHERE transport.category = 'Вантажні'";

    $result = mysqli\_query($link, $sql);

    echo '<h2>Вантажні авто:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "<h4>Знайдено записів: " . $rows . "</h4>";

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID Авто</th><th>Марка</th><th>Тип авто</th><th>Вантажопідйомність</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 4; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

  }

  ?>

**task11.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    $detail = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['detail']));

    $auto = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['auto']));

    $brand = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['brand']));

    $type = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['type']));

    $date1 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date1']));

    $date2 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date2']));

    if (!empty($auto)) {

      $sql = "SELECT r.id, t.id, t.brand,t.category, r.date, r.price, r.detail

      FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE r.detail = '$detail' AND t.id = $auto

        AND (r.date BETWEEN '$date1' AND '$date2')";

    } elseif (!empty($brand)) {

      $sql = "SELECT r.id, t.id, t.brand,t.category, r.date, r.price, r.detail

      FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE r.detail = '$detail' AND t.brand = '$brand'

        AND (r.date BETWEEN '$date1' AND '$date2')";

    } else {

      $sql = "SELECT r.id, t.id, t.brand,t.category, r.date, r.price, r.detail

      FROM repair r

      JOIN transport t ON r.auto = t.id

      WHERE r.detail = '$detail' AND t.category = '$type'

        AND (r.date BETWEEN '$date1' AND '$date2')";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "<h4>Знайдено записів: " . $rows . "</h4>";

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID запису</th><th>ID Авто</th><th>Марка</th><th>Тип авто</th><th>Дата</th><th>Ціна</th><th>Деталь</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 7; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task12.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    $count = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['count']));

    $date1 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date1']));

    $date2 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date2']));

    if($count == "активний")

    $sql = "SELECT transport.id, transport.brand,transport.year, transport.start\_date, transport.passed, transport.finish\_date

      FROM transport

      WHERE transport.start\_date BETWEEN '$date1' AND '$date2'";

    elseif($count == "списаний"){

      $sql = "SELECT transport.id, transport.brand,transport.year, transport.start\_date, transport.passed, transport.finish\_date

      FROM transport

      WHERE transport.finish\_date BETWEEN '$date1' AND '$date2'";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "<h4>Знайдено записів: " . $rows . "</h4>";

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID Авто</th><th>Марка</th><th>Тип авто</th><th>Дата отрмання</th><th>Статус</th><th>Дата списання</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 6; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task13.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    // отримуємо значення полів форми

    $worker = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['worker']));

    $type = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['type']));

    if ($type == "brigadir") {

      $sql = "SELECT id, name, brigada, speciality

      FROM (

        SELECT s.id, s.name, s.brigada, s.speciality

        FROM staff s

        LEFT JOIN brigada b ON s.brigada = b.id

        WHERE b.brigadir = $worker  AND s.id <> $worker AND s.id <> b.master

        UNION

        SELECT d.id, d.name, d.brigada, 'Водій' AS speciality

        FROM driver d

        LEFT JOIN brigada b ON d.brigada = b.id

        WHERE b.brigadir = $worker

      ) AS merged\_staff;";

    } elseif($type == "master") {

      $sql = "SELECT id, name, brigada, speciality

      FROM (

        SELECT s.id, s.name, s.brigada, s.speciality

        FROM staff s

        LEFT JOIN brigada b ON s.brigada = b.id

        WHERE b.master = $worker AND s.id <> $worker

        UNION

        SELECT d.id, d.name, d.brigada, 'Водій' AS speciality

        FROM driver d

        LEFT JOIN brigada b ON d.brigada = b.id

        WHERE b.master = $worker

      ) AS merged\_staff;";

    }

    elseif($type == 'director') {

      $sql = "SELECT id, name, brigada, speciality

      FROM (

        SELECT s.id, s.name, s.brigada, s.speciality

        FROM staff s

        LEFT JOIN brigada b ON s.brigada = b.id

        LEFT JOIN garage g ON b.garage = g.id

        WHERE g.director = $worker

        UNION

        SELECT d.id, d.name, d.brigada, 'Водій' AS speciality

        FROM driver d

        LEFT JOIN brigada b ON d.brigada = b.id

        LEFT JOIN garage g ON b.garage = g.id

        WHERE g.director = $worker

      ) AS merged\_staff;";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "Знайдено записів: " . $rows;

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID Працівника</th><th>Робітник</th><th>ID Бригади</th><th>Спеціальність</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 4; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**task14.php:**

<?php

  require\_once 'connection.php';

  $link = mysqli\_connect($host, $user, $password, $database)

    or die("Помилка" . mysqli\_error($link));

  if (isset($\_GET['submit'])) {

    $auto = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['auto']));

    $type = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['type']));

    $date1 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date1']));

    $date2 = htmlentities(mysqli\_real\_escape\_string($link, $\_GET['date2']));

    if (!empty($auto)) {

      $sql = "Select s.id, s.name, s.speciality,t.id, t.brand, r.detail, r.date From staff s

      JOIN brigada b ON s.brigada = b.id

      JOIN garage g ON b.garage = g.id

      JOIN transport t ON t.garage = g.id

      JOIN repair r ON  t.id = r.auto

      WHERE s.speciality = '$type' AND t.id = $auto AND r.date BETWEEN '$date1' AND '$date2'

      ORDER BY s.id ASC";

    } else {

      $sql = "Select s.id, s.name, s.speciality,t.id, t.brand, r.detail, r.date From staff s

      JOIN brigada b ON s.brigada = b.id

      JOIN garage g ON b.garage = g.id

      JOIN transport t ON t.garage = g.id

      JOIN repair r ON  t.id = r.auto

      WHERE s.speciality = '$type' AND r.date BETWEEN '$date1' AND '$date2'

      ORDER BY s.id ASC";

    }

    $result = mysqli\_query($link, $sql);

    // виводимо результати запиту

    echo '<h2>Результати пошуку:</h2>';

    $rows = mysqli\_num\_rows($result);

    echo "<h4>Знайдено записів: " . $rows . "</h4>";

    if (mysqli\_num\_rows($result) > 0) {

      echo "<table>";

      echo "<tr><th>ID робітника</th><th>ПІБ</th><th>Спеціальність</th><th>ID Авто</th><th>Марка</th><th>Деталь</th><th>Дата</th>";

      for ($i = 0; $i < $rows; ++$i) {

        $row = mysqli\_fetch\_row($result);

        echo "<tr>";

        for ($j = 0; $j < 7; ++$j) echo "<td>$row[$j]</td>";

        echo "</tr>";

      }

      echo "</table>";

      mysqli\_free\_result($result);

    } else {

      echo "No results found.";

    }

    // закриваємо з'єднання з базою даних

    mysqli\_close($link);

  }

  ?>

**index.php:**

<!DOCTYPE *html*>

<html>

<head>

  <meta *charset*="utf-8">

  <link *rel*="preconnect" *href*="https://fonts.googleapis.com">

  <link *rel*="preconnect" *href*="https://fonts.gstatic.com" *crossorigin*>

  <link *href*="https://fonts.googleapis.com/css2?family=Montserrat&display=swap" *rel*="stylesheet">

  <style>

    body {

      background: rgb(242, 217, 217);

      font-family: 'Montserrat', sans-serif;

      display: flex;

      justify-content: center;

      flex-direction: column;

    }

    h1 {

      margin: 120px auto 30px;

      font-size: 48px;

    }

    h3 {

      margin: 0px auto 60px;

      font-size: 36px;

    }

    ul {

      margin: auto;

      display: flex;

      flex-direction: column;

      gap: 5px;

      list-style: none;

    }

    li {

      padding: 5px;

      transition: 0.3s;

    }

    li*:hover* {

      background-color: white;

      color: #000;

    }

    a {

      color: #000;

      text-decoration: none;

    }

  </style>

</head>

<body>

  <h1>Курсова робота з "Організації Баз Даних"</h1>

  <h3>Виконав студент групи КА-01 Бідулін Д.О.</h3>

  <ul>

    <a *href*="task1.php">

      <li>

        1. Отримати дані про автопарк підприємства.

      </li>

    </a>

    <a *href*="task2.php">

      <li>

        2. Отримати перелік і загальне число водіїв по підприємству, або для зазначеної автомашині.

      </li>

    </a>

    <a *href*="task3.php">

      <li>

        3. Отримати розподіл водіїв по автомобілях.

      </li>

    </a>

    <a *href*="task4.php">

      <li>

        4. Отримати дані про розподіл пасажирського автотранспорту по маршрутах.

      </li>

    </a>

    <a *href*="task5.php">

      <li>

        5. Отримати відомості про пробіг автотранспорту певної категорії або конкретної автомашини за вказаний день, місяць та рік.

      </li>

    </a>

    <a *href*="task6.php">

      <li>

        6. Отримати дані про число ремонтів та їх вартості для автотранспорту певної категорії, окремої марки автотранспорту або вказаної автомашини за вказаний період.

      </li>

    </a>

    <a *href*="task7.php">

      <li>

        7. Отримати дані про підпорядкованість персоналу: робітники-бригадири - майстри - начальники ділянок і цехів.

      </li>

    </a>

    <a *href*="task8.php">

      <li>

        8. Отримати відомості про наявність гаражного господарства в цілому і по кожній категорії транспорту.

      </li>

    </a>

    <a *href*="task9.php">

      <li>

        9. Отримати дані про розподіл автотранспорту на підприємстві.

      </li>

    </a>

    <a *href*="task10.php">

      <li>

        10. Отримати відомості про вантажоперевезення, виконаних зазначеною автомашиною за встановлений період.

      </li>

    </a>

    <a *href*="task11.php">

      <li>

        11. Отримати дані про число використаних для ремонту вказаних вузлів і агрегатів для транспорту певної категорії, окремої марки автотранспорту або конкретної автомашини за вказаний період.

      </li>

    </a>

    <a *href*="task12.php">

      <li>

        12. Отримати відомості про отримання та списання автотехніки за вказаний період.

      </li>

    </a>

    <a *href*="task13.php">

      <li>

        13. Отримати склад підлеглих зазначеного бригадира, майстра тощо.

      </li>

    </a>

    <a *href*="task14.php">

      <li>

        14. Отримати дані про роботи, виконані зазначеним фахівцем (зварником, слюсарем тощо) за означений період в цілому і по конкретній автомашині.

      </li>

    </a>

  </ul>

</body>

</html>