

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС  
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»  
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

**КУРСОВА РОБОТА**

з дисципліни

Організація баз даних  
на тему «БД автотранспорту міста»

Студента III курсу групи КА-85  
Спеціальність 124 Системний аналіз  
Діхтяра Артема Вікторовича

Керівник: Афанасьєва І.В.

Національна оцінка: \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка ECTS: \_\_\_\_\_

Члени комісії

|          |  |
|----------|--|
| _____    | _____  |
| (підпис) | (вчене звання, науковий ступінь, прізвище та ініціали) |
| _____    | _____  |
| (підпис) | (вчене звання, науковий ступінь, прізвище та ініціали) |
| _____    | _____  |
| (підпис) | (вчене звання, науковий ступінь, прізвище та ініціали) |

Київ – 2021 рік

## **АНОТАЦІЯ**

Даний документ є пояснювальною запискою до курсової роботи з дисципліни «Організація баз даних» на тему «БД Автопідприємства міста»

Курсова робота присвячена створенню та демонстрації БД автопідприємства міста. Робота була створена у середовищі SQL Workbench.

## **ANNOTATION**

This document is an explanatory note for the final project assignment of the “Organization of Databases” discipline on topic “ Auto-enterprises of the city Database”.

Project is devoted to create and demonstrate work of a copy of a database of Auto-enterprises of the city Database. The project was created in SQL Workbench.

## ЗМІСТ

|   |     |
|---|-----|
| ПЕРЕЛІК СКОРОЧЕНЬ .....   | 4   |
| ВСТУП.....  | 5   |
| РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ.....                                       | 7   |
| 1.1. Умова задачі .....   | 7   |
| 1.2. Обґрунтування вибору методів та інструментів реалізації .....    | 9   |
| РОЗДІЛ 2 АРХІТЕКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ БД .....            | 111 |
| 2.1. Аналіз функціонування та організаційні засади підприємства ..... | 111 |
| 2.2. Проектування структури бази даних .....                          | 111 |
| 2.3. Життєвий цикл БД.....  | 122 |
| РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ВЗАЄМОДІЇ З БД .....                   | 144 |
| 3.1. Інструкція користувача .....                                     | 144 |
| 3.2. Реалізація механізмів SQL. SQL-запити .....                      | 14  |
| 3.3. Вимоги до апаратних і програмних засобів .....                   | 27  |
| 3.4. Випробування розроблених програм .....                           | 27  |
| 3.5. Опис тестової бази даних.....                                    | 35  |
| ВИСНОВКИ .....  | 39  |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....                                   | 40  |
| ДОДАТОК А .....   | 42  |

## **ПЕРЕЛІК СКОРОЧЕНЬ**

БД – База Даних

ПЗ – Програмне забезпечення

РСУБД - Система Управління Реляційними Базами Даних

СУБД - Система Управління Базами Даних

ID – Identifier

PK – Primary Key

SQL - Structured Query Language

## ВСТУП

На сьогоднішній день проектування баз даних набуло важливого значення для багатьох організацій, які для підвищення продуктивності своєї роботи використовують комп'ютерні технології. Базы даних стали основою інформаційних систем, а їх використання становиться невіддільною частиною функціонування будь-яких підприємств.

Метою створення БД є вдосконалення процесу управління діяльністю підприємства шляхом побудови єдиного програмно-технічного комплексу на основі електронного документообігу. У базу даних програми внесені дані про службовців різних спеціалізацій, про марки машин, що утримуються на підприємстві і дані про маршрути перевезення. Програма дає можливість додавати, зберігати й видаляти дані про службовців, наявні одиниці автотранспорту, пробіг та ремонти.

Завданням курсової роботи є проектування бази даних та підготовка усіх необхідних запитів для роботи з нею. Практичним завданням є відточення навичок SQL-програмування, проектування баз даних, дослідження принципів розробки реляційних баз даних на прикладі проектування та створення «БД Автопідприємства», набуття навичок створення запитів у СУБД MySQL Workbench.

В ході виконання роботи було проведено аналізу предметної області, згенеровано таблиці «БД Автопідприємства» та побудовані запити в даній БД.

При виконанні роботи було використано таке програмне забезпечення: операційна система Windows 8.1, СУБД MySQL Workbench 6.3 CE, веб-браузер Google Chrome для пошуку інформації на веб-сайтах, текстовий редактор Microsoft Word 2016 – для оформлення пояснювальної записки.

Пояснювальна записка складається зі вступу, трьох розділів, висновків, додатків та списку використаних джерел.

При написанні роботи використані підручники, навчальні посібники, роботи українських та закордонних науковців, фахівців-програмістів, чинне законодавство України, державні та закордонні стандарти в галузі інформаційних технологій.

## РОЗДІЛ 1

### ПОСТАНОВКА ЗАДАЧІ

#### 1.1. Умова 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'єднуються 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 заміненних 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 автомашини.
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.

## **1.2. Обґрунтування вибору методів та інструментів реалізації**

Першим кроком у реалізації будь-якого програмного продукту є вибір мови та середовища розробки.

Звичайно, оскільки важливість баз даних стрімко зростає, РСУБД набирають популярність. Серед них найпопулярнішими є MySQL та SQL Server, які виконують однакову функцію, хоча й мають різні області використання. Тому перш ніж почати розробку програмного продукту необхідно визначитися з СУБД, у якій буде реалізовано проект.

Як зазначає авторка статті «Реляційні СУБД – порівняння MySQL і SQL сервера», існує декілька ключових відмінностей між цими двома системами які в даному програмному продукті грають важливу роль:

- Синтаксис. За допомогою інформаційного ресурсу «Порівняння різних реалізацій SQL» можна порівняти наявні функції в обох СУБД та побачити яким чином вони реалізуються. Набувши навичок роботи з MySQL протягом семестру цей набір таблиць допоможе освоїтися у роботі з SQL Server.

- Скасування запитів. Не всім відомо що MySQL не дозволяє скасовувати запити посеред їх виконання. Це означає що як тільки команда була запущена на виконання, залишається тільки сподіватися що можливі збитки є оборотними. В

даній ситуації SQL Server без лишніх труднощів дозволив би зупинити запит посеред його виконання. В моєму випадку зважаючи на апаратне забезпечення яке я використовую, ця різниця є досить важливою.

- **Вартість.** MySQL має перевагу в тому що ця система розповсюджується безкоштовно. Хоча ціна для юридичних осіб на MS SQL Server виявляється неприємною для більшої частини організацій, Microsoft пропонує безкоштовну версію продукту.

Володимир Драч у своїй статті також перераховує декілька недоліків обох СУБД. Так наприклад, MySQL «пропонує велику кількість функцій», «прекрасно документована» та «може працювати з іншими БД, включаючи DB2 та Oracle», проте в ній «відсутня вбудована підтримка XML або OLAP», «для безкоштовної версії доступна тільки платна підтримка» і «доведеться витратити багато часу і зусиль, щоб змусити MySQL виконувати нескладні задачі, хоча інші системи роблять це автоматично». З іншої сторони Драч зазначає, що MS SQL Server «простий у використанні», його «поточна версія працює швидко і стабільно», «движок надає можливість регулювати й відстежувати рівні продуктивності, які допомагають знизити використання ресурсів».

Таким чином, дана БД реалізована в СУБД MySQL за допомогою середовища розробки MySQL Workbench 6.3, яке в свою чергу надає засоби для налаштування, спостереження й адміністрування баз даних і за допомогою якого можна розгортати, відстежувати та оновлювати компоненти рівня даних, що використовуються вашими додатками, а також створювати запити і скрипти.

## **РОЗДІЛ 2**

### **АРХІТЕКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ БД**

#### **2.1. Аналіз функціонування та організаційні засади підприємства**

Для реалізації даного програмного продукту необхідно спроектувати БД автопідприємства міста. Для цього проаналізуємо умову та вимоги завдання і на їх основі створимо таблиці.

До основних функціональних завдань підприємства відносяться: пасажирські та вантажні перевезення, облік персоналу(водіїв, слюсарів, зварників, тощо), контроль маршрутів, ремонт автотранспорту.

З умови видно, що для даної БД необхідно створити таблиці для Автомобілів, Працівників, Вантажоперевезень, Пробігу, Маршрутів, Гаражних господарств.

#### **2.2. Проектування структури бази даних**

В результаті аналізу функціонування та організаційні засади підприємства, можна зробити висновок про таблиці, що нам знадобляться.

Діаграма БД зображена на рисунку 2.1.

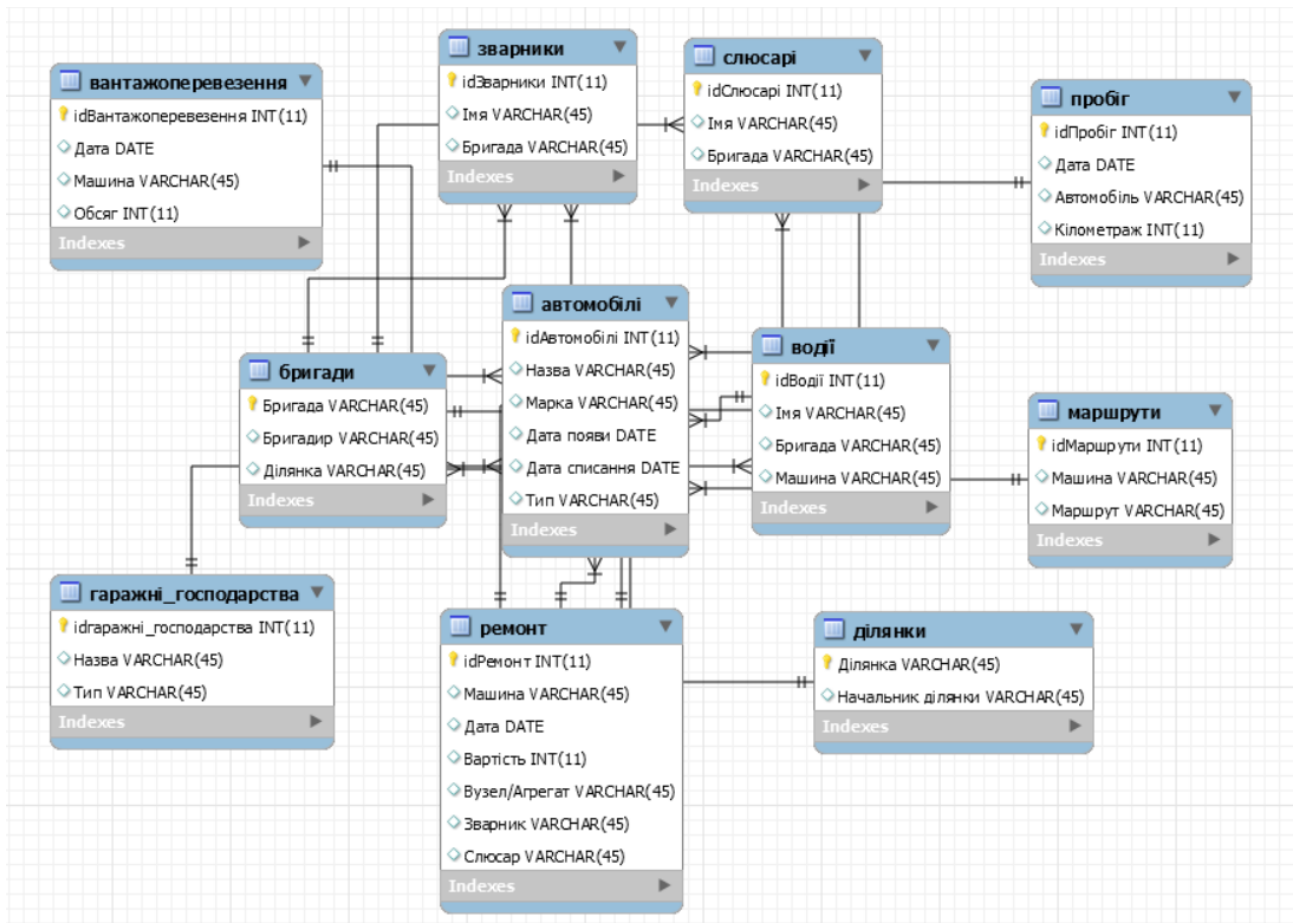


Рис. 2.1 Діаграма БД Автопідприємство

## 2.3. Життєвий цикл БД

### 1. Попереднє планування

На даному етапі була сформована модель структури бази даних, визначені сутності, проаналізовано зв'язки між сутностями на основі вимог до даної БД.

### 2. Перевірка здійсненності

Для розробки даної системи необхідно мати персональний комп'ютер, на якому інстальовано інструменти для розробки бази даних мовою програмування MySQL. Розробник має бути знайомий з даною мовою програмування. Наведені вимоги виконано, отже завдання є технічно здійсненним.

### 3. Визначення вимог

Вимоги до даної БД були сформовані замовником, зокрема вказана структура організації, види інформаційних запитів, які повинні бути передбачені у даній системі.

### 4. Проектування

Для реалізації технічного завдання обрано реляційну модель бази даних. В якості середовища розробки обрано MySQL Workbench. Проаналізовано сутності, визначені атрибути, ключові поля, типи зв'язків, побудовано EER-діаграму. Протестовано роботу системи на тестових даних.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ВЗАЄМОДІЇ З БД

#### 3.1. Інструкція користувача

Для того щоб користувач мав можливість редагувати та додавати дані в БД йому потрібно мати будь-який редактор MySQL. В даній інструкції вважаємо, що цим редактором є MySQL Workbench 6.3.

Перш за все необхідно створити сервер або під'єднатися до існуючого. Далі треба створити саму БД на цьому сервері. Для цього створюємо новий запит і копіюємо код з Додатку А під назвою “CREATE\_BD”. Натискаємо F5, або кнопку “Execute”, що запустити виконання програми. Для заповнення цієї БД далі аналогічно копіюємо частину коду з додатку під назвою “INSERT\_BD”. Запустимо виконання. Для виконання запитів необхідно аналогічно спіювати та запустити відповідну частину коду під назвою “REQUESTS\_BD”. Вже працюючи в MySQL, для виконати окремий шматок коду досить виділити його та запустити. Фільтри та сортування можна редагувати, додаючи або прибираючи відповідні умови та замінюючи відповідні значення.

Детальніше розглянемо кожний запит окремо у підрозділах 3.2.1 – 3.2.13.

#### 3.2. Реалізація механізмів SQL. SQL-запити

В результаті роботи було отримано 11 таблиць, які можна розділити на декілька підтипів за функціоналом. Основними таблицями вважатимемо таблиці *Автомобілі*, одна з таблиць працівників, наприклад *Водії*, оскільки з ними зв'язана більшість таблиць. Опис кожної з цих таблиць представлено у таблицях 3.1 та 3.2

Таблиця 3.1 Опис таблиці *Автомобілі*

|    | Назва поля    | Тип         | Опис  |
|----|---------------|-------------|---|
| PK | ID            | int         | унікальний ідентифікатор                                  |
|    | Назва         | varchar(45) | назва машини  |
|    | Марка         | varchar(45) | марка машини  |
|    | Дата появи    | date        | дата появи машини у підприємстві                          |
|    | Дата списання | date        | дата списання автомобіля(null, якщо авомобіль не списано) |
|    | Тип           | varchar(45) | Тип машини від його функцій(вантажівка, автобус, тощо).   |

Таблиця 3.2 Опис таблиці *Водії*

|    | Назва поля | Тип         | Опис                              |
|----|------------|-------------|-----------------------------------|
| PK | ID         | int         | унікальний ідентифікатор          |
|    | Ім'я       | varchar(45) | ім'я службовця                    |
|    | Бригада    | varchar(45) | назва бригади                     |
|    | Машина     | varchar(45) | машина, за якою закріплений водій |

Таким чином у таблиці *Автомобілі* зберігається основна інформація про кожну індивідуальну машину, а в таблиці *Водії* – про кожного водія.

Опишемо інші таблиці, що необхідно створити для виконання завдань, споміжні таблиці, додамо до них опис.

Бригади – таблиця для обліку бригад, у які об'єднуються працівники.

Опис таблиці наведено у таблиці 3.3

Таблиця 3.3 Опис таблиці *Бригади*

|    | Назва поля | Тип         | Опис                                  |
|----|------------|-------------|---------------------------------------|
| PK | Бригада    | varchar(45) | унікальна назва бригади               |
|    | Бригадир   | varchar(45) | Ім'я головного по бригаді – бригадира |
|    | Ділянка    | varchar(45) | ділянка, до якої належить бригада     |

Ділянка - це найвища управлінська одиниця в структурі підприємства. В ділянку входять кілька бригад.

## Опис Ділянок у таблиці 3.4

Таблиця 3.4 Опис таблиці *Ділянки*

|    | Назва поля        | Тип         | Опис                      |
|----|-------------------|-------------|---------------------------|
| PK | Ділянка           | varchar(45) | унікальна назва ділянки   |
|    | Начальник ділянки | varchar(45) | ім'я головного по ділянці |

Ведеться облік вантажоперевезень, що виконуються вантажівками для кожного дня та окремої машини.

## Опис таблиці Вантажоперевезення знаходиться у таблиці 3.5

Таблиця 3.5 Опис таблиці *Вантажоперевезення*

|    | Назва поля | Тип         | Опис   |
|----|------------|-------------|--|
| PK | ID         | int         | ідентифікатор вантажоперевезення                     |
|    | Дата       | date        | дата, коли було здійснено вантажоперевезення         |
|    | Машина     | varchar(45) | назва машини, якою було виконано вантажоперевезення  |
|    | Обсяг      | int         | число вантажу у тонах, що перевезла машина у цю дату |

Для машин ведеться облік пробігу. Пробіг записується окремо для кожної машини в кожен окремий день

## Інформація про Пробіг у таблиці 3.6

Таблиця 3.6 Опис таблиці *Пробіг*

|    | Назва поля | Тип         | Опис                           |
|----|------------|-------------|--------------------------------|
| PK | ID         | int         | ідентифікатор пробігу          |
|    | Дата       | date        | дата запису пробігу автомобіля |
|    | Автомобіль | varchar(45) | машина, що здійснила поїздки   |

Окрім водіїв у штаті містяться робітники, що займаються ремонтними роботами. Це слюсарі та зварники. Опис цих таблиць знаходиться у таблицях 3.7 та 3.8.



Таблиця 3.7 Опис таблиці *Слюсарі*

|    | Назва поля | Тип         | Опис  |
|----|------------|-------------|---|
| PK | ID         | int         | ідентифікатор працівника                    |
|    | Ім'я       | varchar(45) | ім'я слюсаря                                |
|    | Бригада    | varchar(45) | бригада, в яку входить конкретний працівник |

Таблиця 3.8 Опис таблиці *Зварники*

|    | Назва поля | Тип         | Опис  |
|----|------------|-------------|---|
| PK | ID         | int         | ідентифікатор працівника                    |
|    | Ім'я       | varchar(45) | ім'я зварника                               |
|    | Бригада    | varchar(45) | бригада, в яку входить конкретний працівник |

Зварники та слюсарі займаються ремонтом автомобілей, заміною чи відновленням певних вузлів чи агрегатів. Також ведеться облік марок авто, що ремонтуються, адже часто однакові деталі однакових марок машин можна без проблем встановити різним автомобілям. Ведеться облік вартості ремонту та запчастин, що ремонтуються.

Опис таблиці *Ремонт* знаходиться у таблиці 3.9.

Таблиця 3.9 Опис таблиці *Ремонт*

|    | Назва поля    | Тип         | Опис                              |
|----|---------------|-------------|-----------------------------------|
| PK | ID            | int         | ідентифікатор ремонту             |
|    | Машина        | varchar(45) | назва машини, що ремонтується     |
|    | Дата          | date        | дата ремонту                      |
|    | Вартість      | int         | вартість ремонту                  |
|    | Вузел/Агрегат | varchar(45) | назва запчастини, що ремонтується |
|    | Зварник       | varchar(45) | ім'я зварника, що проводив ремонт |
|    | Слюсар        | varchar(45) | ім'я слюсаря, що проводив ремонт  |

Вся техніка міститься в гаражних господарствах. Інформація про гаражні господарства знаходиться в таблиці 3.10

Таблиця 3.10 Опис таблиці *гаражні\_господарства*

|    | Назва поля | Тип         | Опис                                 |
|----|------------|-------------|--------------------------------------|
| PK | ID         | int         | ідентифікатор гаражного господарства |
|    | Назва      | varchar(45) | назва гаражного господарства         |

|  |     |             |  |
|--|-----|-------------|--|
|  | Тип | varchar(45) | Тип автомобілей, що знаходяться в цьому гаражному господарстві |
|--|-----|-------------|--|

Пасажирський автотранспорт (автобуси, маршрутні таксі) перевозить пасажирів за визначеними маршрутами, за кожним з них закріплені окремі одиниці автотранспорту.

Опис таблиці *Маршрути* міститься в таблиці 3.11.

Таблиця 3.11 Опис таблиці *Маршрути*

|    | Назва поля | Тип         | Опис  |
|----|------------|-------------|---|
| PK | ID         | int         | ідентифікатор                               |
|    | Машина     | varchar(45) | назва машини, що їздить за певним маршрутом |
|    | Маршрут    | varchar(45) | назва маршруту                              |

Повний код запитів можна побачити у Додатку А. Тому розглянемо кожний запит конкретніше.

### 3.2.1. Запит 1

Завдання: 1. Отримати дані про автопарк підприємства.

Щоб отримати дані про автопарк, запустимо процедуру task1

```
delimiter $$
-- drop procedure if exists task1;
create procedure task1()
begin
    select Назва from Автомобілі;
end $$
delimiter $$
```

### 3.2.2. Запит 2

Завдання: Отримати перелік і загальне число водіїв по підприємству, або для зазначеної автомашини.

Щоб отримати перелік і загальне число водіїв по підприємству, запустимо процедуру task2\_1

```
delimiter $$
-- drop procedure if exists task2_1;
```

```
create procedure task2_1()
begin
    select Імя from водії;
    select count(*) from водії;
end$$
DELIMITER $$
```

Щоб отримати перелік і загальне число водіїв, відповідальних за вказану машину, запусимо процедуру task2\_2

```
delimiter $$
-- drop procedure if exists task2_2;
create procedure task2_2(IN name_car VARCHAR(45))
begin
    select Імя as result
    from водії
    where Машина = name_car;
    select count(Імя)
    from водії
    where Машина = name_car;
end$$
DELIMITER $$
```

### 3.2.3. Занит 3

Завдання: Отримати розподіл водіїв по автомобілях.

Запусимо процедуру task3

```
delimiter $$
-- drop procedure if exists task3;
create procedure task3()
begin
    select Імя, Машина from водії;
end$$
DELIMITER $$
```

### 3.2.4. Занит 4

Завдання: Отримати дані про розподіл пасажирського автотранспорту по маршрутах.

Запусимо процедури task4

```
delimiter $$
-- drop procedure if exists task4;
create procedure task4()
begin
    select Машина, Маршрут from маршрути;
end$$
DELIMITER $$
```

### 3.2.5. Занит 5

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

Для відомостей про пробіг автівок певної категорії запусимо процедуру task5\_1

```
delimiter $$
-- drop procedure if exists task5_1;
create procedure task5_1(IN type_car VARCHAR(45))
begin
    select пробіг.Дата, пробіг.Кілометраж, автомобілі.Назва, автомобілі.Тип
    from пробіг
    inner join автомобілі
    on автомобілі.Назва = пробіг.Автомобіль
    where Тип = type_car;
end$$
DELIMITER $$
```

Для відомостей про пробіг конкретної машини за вказаний період запусимо процедуру task5\_2

```
delimiter $$
-- drop procedure if exists task5_2;
create procedure task5_2(IN car VARCHAR(45), IN date1 DATE, IN date2 DATE)
begin
    select пробіг.Дата, пробіг.Кілометраж, автомобілі.Назва, автомобілі.Тип
    from пробіг
    inner join автомобілі
    on автомобілі.Назва = пробіг.Автомобіль
    where пробіг.Дата between date1 and date2 AND пробіг.Автомобіль = car;
end$$
DELIMITER $$
```

### 3.2.6. Заняття 6

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

Щоб отримати дані про число ремонтів та їх вартості для авто певної категорії запусимо task6\_1

```
delimiter $$
-- drop procedure if exists task6_1;
create procedure task6_1(IN type_car VARCHAR(45))
begin
    select count(idРемонт), sum(Вартість)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where Тип = type_car;
end$$
```

```
DELIMITER $$
```

Щоб отримати дані про число ремонтів та їх вартості для авто певної марки запустимо task6\_2

```
delimiter $$
-- drop procedure if exists task6_2;
create procedure task6_2(IN marka_car VARCHAR(45))
begin
    select count(idРемонт), sum(Вартість)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where Марка = marka_car;
end$$
DELIMITER $$
```

Щоб отримати дані про число ремонтів та їх вартості для певного авто за вказаний період запустимо task6\_3

```
delimiter $$
-- drop procedure if exists task6_3;
create procedure task6_3(IN car VARCHAR(45), IN date1 DATE, IN date2 DATE)
begin
    select count(idРемонт), sum(Вартість)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where ремонт.Дата between date1 and date2 AND Машина = car;
end$$
DELIMITER $$
```

### 3.2.7. Завдання 7

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

Запустимо task7

```
delimiter $$
-- drop procedure if exists task7;
create procedure task7(IN name_person varchar(45))
begin
    if (exists(select імя from водії where імя = name_person)) then begin
        select name_person
        union
        select Бригадир from бригади
        inner join водії on водії.Бригада = бригади.Бригада where водії.Імя =
name_person
        union
        select `Начальник ділянки` from ділянки
```

```

        inner join бригади on бригади.Ділянка = ділянки.Ділянка where бригади.Бригадир
= (
        select Бригадир from бригади
        inner join водії on водії.Бригада = бригади.Бригада where водії.Імя =
name_person)
        union
        select "Водій";
    end;

    elseif (exists(select імя from Зварники where імя = name_person)) then begin
        select name_person
        union
        select Бригадир from бригади
        inner join Зварники on Зварники.Бригада = бригади.Бригада where Зварники.Імя =
name_person
        union
        select `Начальник ділянки` from ділянки
        inner join бригади on бригади.Ділянка = ділянки.Ділянка where бригади.Бригадир
= (
        select Бригадир from бригади
        inner join Зварники on Зварники.Бригада = бригади.Бригада where Зварники.Імя =
name_person)
        union
        select "Зварювальник";
    end;

    elseif (exists(select імя from Слюсарі where імя = name_person)) then begin
        select name_person
        union
        select Бригадир from бригади
        inner join Слюсарі on Слюсарі.Бригада = бригади.Бригада where Слюсарі.Імя =
name_person
        union
        select `Начальник ділянки` from ділянки
        inner join бригади on бригади.Ділянка = ділянки.Ділянка where бригади.Бригадир
= (
        select Бригадир from бригади
        inner join Слюсарі on Слюсарі.Бригада = бригади.Бригада where Слюсарі.Імя =
name_person)
        union
        select "Слюсар";
    end;

    elseif (exists(select Бригадир from Бригади where Бригадир = name_person)) then begin
        select name_person
        union
        select `Начальник ділянки` from ділянки
        inner join Бригади on Бригади.ділянка = ділянки.ділянка where Бригади.Бригадир
= name_person
        union
        select "Бригадир";
    end;

    elseif (exists(select ділянка from Ділянки where `Начальник ділянки` = name_person))
then begin
        select name_person
        union
        select "Начальник ділянки";
    end;

    else select "some problems";
end if;
end$$

```

```
DELIMITER $$
```

### 3.2.8. Завдання 8

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

Щоб дізнатись про наявність гаражного господарства в цілому запусимо task8\_1

```
delimiter $$
-- drop procedure if exists task8_1;
create procedure task8_1()
begin
    select if((select count(*) from гаражні_господарства)!=0, 'гаражні господарства є',
'гаражних господарств немає');
end$$
DELIMITER $$
```

Щоб дізнатись про наявність гаражного господарства для конкретної категорії запусимо task8\_2

```
delimiter $$
-- drop procedure if exists task8_2;
create procedure task8_2(IN type_car varchar(45))
begin
    select if((select count(*) from гаражні_господарства where Тип = type_car)!=0,
'гаражне господарство такого типу є', 'гаражного господарства такого типу немає');
end$$
DELIMITER $$
```

### 3.2.9. Завдання 9

Завдання: Отримати дані про розподіл автотранспорту на підприємстві.

Запусимо task9

```
delimiter $$
-- drop procedure if exists task9;
create procedure task9()
begin
    select автомобілі.Назва, автомобілі.Тип, гаражні_господарства.Назва as `Гаражне
господарство` from автомобілі
    inner join гаражні_господарства on гаражні_господарства.Тип = автомобілі.Тип;
end$$
DELIMITER $$
```

### 3.2.10. Завдання 10

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

Щоб отримати перелік і загальне число тварин, яким необхідний певний тип кормів з відповідними фільтрами запустимо

```
delimiter $$
-- drop procedure if exists task10;
create procedure task10(IN car VARCHAR(45), IN date1 DATE, IN date2 DATE)
begin
    select * from вантажоперевезення
    where Дата between date1 and date2 AND Машина = car;
end$$
DELIMITER $$
```

### 3.2.11. Завдання 11

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

Щоб отримати дані про число використаних для ремонту вказаних вузлів і агрегатів для транспорту певної категорії запустимо task11\_1

```
delimiter $$
-- drop procedure if exists task11_1;
create procedure task11_1(IN element_car VARCHAR(45), IN type_car varchar(45))
begin
    select count(idРемонт)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where `ремонт`.`Вузел/Агрегат` = element_car AND Тип = type_car;
end$$
DELIMITER $$
```

Щоб отримати дані про число використаних для ремонту вказаних вузлів і агрегатів для транспорту окремої марки запустимо task11\_2

```
delimiter $$
-- drop procedure if exists task11_2;
create procedure task11_2(IN element_car VARCHAR(45), IN marka_car varchar(45))
begin
    select count(idРемонт)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where `ремонт`.`Вузел/Агрегат` = element_car AND Марка = marka_car;
end$$
DELIMITER $$
```

Щоб отримати дані про число використаних для ремонту вказаних вузлів і агрегатів для конкретної машини за вказаний період запустимо task11\_3



```

delimiter $$
-- drop procedure if exists task11_3;
create procedure task11_3(IN element_car varchar(45), IN car VARCHAR(45), IN date1 DATE, IN
date2 DATE)
begin
    select count(idРемонт)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where Дата between date1 and date2 AND Машина = car AND `ремонт`.`Вузел/Агрегат` =
element_car;
end$$
DELIMITER $$

```

### 3.2.12. Завдання 12

Завдання: Отримати відомості про отримання та списання автотехніки за вказаний період.

Запустимо task12

```

delimiter $$
-- drop procedure if exists task12;
create procedure task12(IN date1 DATE, IN date2 DATE)
begin
    select Назва, `Дата появи`, `Дата списання` from автомобілі
    where `Дата появи` between date1 and date2 OR `Дата списання` between date1 and date2;
end$$
DELIMITER $$

```

### 3.2.13. Завдання 13

Завдання: Отримати склад підлеглих зазначеного бригадира, майстра тощо.

Запустимо task13

```

delimiter $$
-- drop procedure if exists task13;
create procedure task13(IN person_name varchar(45))
begin
    if (exists(select Бригадир from Бригади where Бригадир = person_name)) then begin
        select "Бригадир"
        union
        select Ім'я from водії
        inner join бригади on водії.Бригада = бригади.Бригада where Бригадир =
person_name
        union
        select Ім'я from зварники
        inner join бригади on зварники.Бригада = бригади.Бригада where Бригадир =
person_name
        union
        select Ім'я from слюсарі
        inner join бригади on слюсарі.Бригада = бригади.Бригада where Бригадир =
person_name;
    end if;
end;

```

```

        end;

        elseif (exists(select `Начальник ділянки` from ділянки where `Начальник ділянки` =
person_name)) then begin
            select "Начальник ділянки"
        union
            select Бригадир from Бригади
            inner join ділянки on бригади.Ділянка = ділянки.Ділянка where `Начальник
ділянки` = person_name;
        end;

        else select "some problems";
    end if;
end$$
DELIMITER $$

```

### 3.2.14. Запит 14

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

Щоб отримати дані про роботи, виконані зазначеним фахівцем (зварником, слюсарем тощо) за означений період в цілому запустимо task14\_1

```

delimiter $$
-- drop procedure if exists task14_1;
create procedure task14_1(IN name_worker VARCHAR(45), IN car varchar(45))
begin
    select Дата, Машина, `Вузел/Агрегат` from ремонт
    where (Зварник = name_worker and Машина = car) or (Слюсар = name_worker and Машина =
car);
end$$
DELIMITER $$

```

Щоб отримати дані про роботи, виконані зазначеним фахівцем (зварником, слюсарем тощо) по конкретній автомашині запустимо task14\_2

```

delimiter $$
-- drop procedure if exists task14_2;
create procedure task14_2(IN name_person VARCHAR(45), IN date1 DATE, IN date2 DATE)
begin
    select Дата, Машина, `Вузел/Агрегат` from ремонт
    where Дата between date1 and date2 AND (Зварник = name_person or Слюсар = name_person);
end$$
DELIMITER $$

```

### 3.3. Вимоги до апаратних і програмних засобів

Даний програмний продукт був протестований на комп'ютері з процесором Intel Core i7-4500U CPU @ 1.80GHz за допомогою ПЗ MySQL Workbench 6.3. Мінімальні вимоги для апаратного забезпечення: 512 MB пам'яті, x64 процесор: 1.4GHz типу AMD Opteron, AMD Athlon 64, Intel Xeon з підтримкою Intel EM64T або Intel Pentium IV з підтримкою EM64T

### 3.4. Випробування розроблених програм

В даному підрозділі продемонстровано роботу запитів на конкретних випадках.

#### 3.4.1 Запит 1

Отримати дані про автопарк підприємства.

call task1();

| Назва                  |
|------------------------|
| ЛАЗ-697                |
| Рута 22                |
| Рута 23                |
| Стрий Авто А075        |
| Стрий Авто А0756       |
| ЛАЗ-699                |
| ГалАЗ-3208             |
| Рута 37                |
| ГалАЗ-3209             |
| ЛАЗ-5255               |
| Богдан-3355            |
| ГАЗ-51                 |
| ГАЗ-68                 |
| ЗІЛ-138                |
| ЗІЛ-4331               |
| Богдан А091            |
| БАЗ-2215               |
| ГАЗ-3221               |
| ЧАЗ-983                |
| Mercedes-Benz Vito 638 |
| Mercedes-Benz Vito 639 |

### 3.4.2 Запит 2

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

call task2\_1()

|                     |
|---------------------|
| Імя                 |
| Елисеїв Інокентій   |
| Вакулин Владислав   |
| Шорин Богдан        |
| Крушевський Аркадій |
| Духовський Вадим    |
| Шелай Валентин      |
| Каленов Вадим       |
| Авербах Ярослав     |
| Наркевич Данило     |
| Богаєвський Павло   |
| Іванов Микита       |
| Білоус Стас         |
| count(*)            |
| 12                  |

call task2\_2('Рута 37');

|                   |
|-------------------|
| result            |
| Елисеїв Інокентій |
| Вакулин Владислав |
| count(Імя)        |
| 2                 |

### 3.4.3. Запит 3

Отримати розподіл водіїв по автомобілях.

call task3();

| Імя                 | Машина               |
|---------------------|----------------------|
| Єлисеїв Інокентій   | Рута 37              |
| Вакулин Владислав   | Рута 37              |
| Шорин Богдан        | ЛАЗ-699              |
| Крушевський Аркадій | ЛАЗ-5255             |
| Духовський Вадим    | Стрий Авто А0756     |
| Шелай Валентин      | ЗІЛ-138              |
| Каленов Вадим       | Богдан А091          |
| Авербах Ярослав     | ГАЗ-68               |
| Наркевич Данило     | ГалАЗ-3208           |
| Богаєвський Павло   | ЗІЛ-4331             |
| Іванов Микита       | Mercedes-Benz Vit... |
| Білоус Стас         | ГАЗ-3221             |

### 3.4.4. Запит 4

Отримати дані про розподіл пасажирського автотранспорту по маршрутах.

call task4();

| Машина                 | Маршрут |
|------------------------|---------|
| ЛАЗ-699                | М-100   |
| Рута 37                | М-101   |
| Стрий Авто А0756       | М-102   |
| Mercedes-Benz Vito 638 | М-103   |
| ГАЗ-3221               | М-103   |

### 3.4.5. Завдання 5

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

```
call task5_1('Автобус');
```

| Дата       | Кілометраж | Назва            | Тип     |
|------------|------------|------------------|---------|
| 2019-07-08 | 180        | Стрий Авто А0756 | Автобус |
| 2021-05-05 | 90         | ЛАЗ-699          | Автобус |
| 2019-06-06 | 100        | Рута 37          | Автобус |
| 2019-06-07 | 120        | Рута 37          | Автобус |
| 2019-07-07 | 150        | Рута 37          | Автобус |
| 2020-07-07 | 180        | Рута 37          | Автобус |
| 2017-06-06 | 80         | ЛАЗ-5255         | Автобус |
| 2017-07-07 | 100        | ЛАЗ-5255         | Автобус |
| 2018-06-06 | 120        | ЛАЗ-5255         | Автобус |

```
call task5_2('Рута 37', '2019-07-06', '2020-08-06');
```

| Дата       | Кілометраж | Назва   | Тип     |
|------------|------------|---------|---------|
| 2019-07-07 | 150        | Рута 37 | Автобус |
| 2020-07-07 | 180        | Рута 37 | Автобус |

### 3.4.6. Завдання 6

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

```
call task6_1('Автобус');
```

| count(idРемонт) | sum(Вартість) |
|-----------------|---------------|
| 2               | 15000         |

```
call task6_2('Богдан');
```

| count(idРемонт) | sum(Вартість) |
|-----------------|---------------|
| 2               | 20000         |

```
call task6_3('Богдан-3355', '2016-07-06', '2020-08-06');
```

| count(idРемонт) | sum(Вартість) |
|-----------------|---------------|
| 1               | 10000         |

#### 3.4.7. Запит 7

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

```
call task7('Шорин Богдан');
```

| name_person          |
|----------------------|
| Шорин Богдан         |
| Китаєв Олег          |
| Радолинський Ярослав |
| Водій                |

#### 3.4.8. Запит 8

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

```
call task8_1();
```

| if((select count(*) from |
|--------------------------|
| гаражні господарства є   |

```
call task8_2('Таксі');
```

| if((select count(*) from гаражні_господарства |
|---|
| гаражного господарства такого типу немає      |

### 3.4.9. Завдання 9

Отримати дані про розподіл автотранспорту на підприємстві.

call task9();

| Назва                  | Тип             | Гаражне господарство |
|------------------------|-----------------|----------------------|
| ЛАЗ-697                | Автобус         | Г-1                  |
| Рута 22                | Автобус         | Г-1                  |
| Рута 23                | Автобус         | Г-1                  |
| Стрий Авто А075        | Автобус         | Г-1                  |
| Стрий Авто А0756       | Автобус         | Г-1                  |
| ЛАЗ-699                | Автобус         | Г-1                  |
| ГалАЗ-3208             | Автобус         | Г-1                  |
| Рута 37                | Автобус         | Г-1                  |
| ГалАЗ-3209             | Автобус         | Г-1                  |
| ЛАЗ-5255               | Автобус         | Г-1                  |
| Богдан-3355            | Вантажівка      | Г-3                  |
| ГАЗ-51                 | Вантажівка      | Г-3                  |
| ГАЗ-68                 | Вантажівка      | Г-3                  |
| ЗІЛ-138                | Вантажівка      | Г-3                  |
| ЗІЛ-4331               | Вантажівка      | Г-3                  |
| Богдан А091            | Вантажівка      | Г-3                  |
| БАЗ-2215               | Маршрутне таксі | Г-2                  |
| ГАЗ-3221               | Маршрутне таксі | Г-2                  |
| ЧАЗ-983                | Маршрутне таксі | Г-2                  |
| Mercedes-Benz Vito 638 | Маршрутне таксі | Г-2                  |
| Mercedes-Benz Vito 639 | Маршрутне таксі | Г-2                  |

### 3.4.10. Завдання 10

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

call task10('ЗІЛ-138', '2017-07-06', '2020-08-06');

| idВантажоперевезення | Дата       | Машина  | Обсяг |
|----------------------|------------|---------|-------|
| 5                    | 2018-06-06 | ЗІЛ-138 | 250   |
| 9                    | 2019-10-10 | ЗІЛ-138 | 250   |



### 3.4.11. Занум 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.

```
call task11_1('шасі', 'Автобус');
```

| count(idРемонт) |
|-----------------|
| 1               |

```
call task11_2('шасі', 'Рута');
```

| count(idРемонт) |
|-----------------|
| 1               |

```
call task11_3('двигун', 'Богдан-3355', '2016-07-06', '2020-08-06');
```

| count(idРемонт) |
|-----------------|
| 1               |

### 3.2.12. Занум 12

Отримати відомості про отримання та списання автотехніки за вказаний період.

```
call task12('2017-07-23', '2018-07-23');
```

| Назва       | Дата появи | Дата списання |
|-------------|------------|---------------|
| Рута 23     | 2015-08-23 | 2018-07-23    |
| ЛАЗ-699     | 2017-07-23 | NULL          |
| ГалАЗ-3208  | 2017-07-23 | NULL          |
| Рута 37     | 2018-07-23 | NULL          |
| Богдан-3355 | 2015-07-23 | 2018-07-23    |
| ГАЗ-51      | 2015-07-23 | 2018-07-23    |
| БАЗ-2215    | 2015-07-23 | 2018-06-23    |

### 3.2.13. Запит 13

Отримати склад підлеглих зазначеного бригадира, майстра тощо.

```
call task13('Китаев Олег');
```

|                   |
|-------------------|
| Бригадир          |
| Бригадир          |
| Єлисеїв Інокентій |
| Вакулин Владислав |
| Шорин Богдан      |
| Ачкасов Валерій   |
| Анохин Іван       |

### 3.2.14. Запит 14

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

```
call task14_1('Ачкасов Валерій', 'Богдан-3355');
```

| Дата       | Машина      | Вузел/Агрегат |
|------------|-------------|---------------|
| 2017-06-06 | Богдан-3355 | двигун        |

```
call task14_2('Анохин Іван', '2016-07-06', '2020-08-06');
```

| Дата       | Машина      | Вузел/Агрегат |
|------------|-------------|---------------|
| 2017-06-06 | Богдан-3355 | двигун        |

### 3.5. Опис тестової бази даних

На наступних рисунках наведено таблиці із бази даних

| idАвтомобілі | Назва                  | Марка         | Дата появи | Дата списання | Тип             |
|--------------|------------------------|---------------|------------|---------------|-----------------|
| 200          | ЛАЗ-697                | ЛАЗ           | 2015-07-23 | 2018-08-23    | Автобус         |
| 201          | Рута 22                | Рута          | 2015-07-23 | 2018-08-23    | Автобус         |
| 202          | Рута 23                | Рута          | 2015-08-23 | 2018-07-23    | Автобус         |
| 203          | Стрий Авто А075        | Стрий Авто    | 2016-07-23 | NULL          | Автобус         |
| 204          | Стрий Авто А0756       | Стрий Авто    | 2016-07-23 | NULL          | Автобус         |
| 205          | ЛАЗ-699                | ЛАЗ           | 2017-07-23 | NULL          | Автобус         |
| 206          | ГалАЗ-3208             | ГалАЗ         | 2017-07-23 | NULL          | Автобус         |
| 207          | Рута 37                | Рута          | 2018-07-23 | NULL          | Автобус         |
| 208          | ГалАЗ-3209             | ГалАЗ         | 2018-08-23 | NULL          | Автобус         |
| 209          | ЛАЗ-5255               | ЛАЗ           | 2018-08-23 | NULL          | Автобус         |
| 210          | Богдан-3355            | Богдан        | 2015-07-23 | 2018-07-23    | Вантажівка      |
| 211          | ГАЗ-51                 | ГАЗ           | 2015-07-23 | 2018-07-23    | Вантажівка      |
| 212          | ГАЗ-68                 | ГАЗ           | 2015-07-23 | NULL          | Вантажівка      |
| 213          | ЗІЛ-138                | ЗІЛ           | 2016-07-23 | NULL          | Вантажівка      |
| 214          | ЗІЛ-4331               | ЗІЛ           | 2016-07-23 | NULL          | Вантажівка      |
| 215          | Богдан А091            | Богдан        | 2016-07-23 | NULL          | Вантажівка      |
| 216          | БАЗ-2215               | БАЗ           | 2015-07-23 | 2018-06-23    | Маршрутне таксі |
| 217          | ГАЗ-3221               | ГАЗ           | 2015-07-23 | NULL          | Маршрутне таксі |
| 218          | ЧАЗ-983                | ЧАЗ           | 2015-07-23 | NULL          | Маршрутне таксі |
| 219          | Mercedes-Benz Vito 638 | Mercedes-Benz | 2016-07-23 | NULL          | Маршрутне таксі |
| 220          | Mercedes-Benz Vito 639 | Mercedes-Benz | 2016-07-23 | NULL          | Маршрутне таксі |
| NULL         | NULL                   | NULL          | NULL       | NULL          | NULL            |

Рисунок 3.1 Таблиця *Автомобілі*

| Бригада | Бригадир        | Ділянка |
|---------|-----------------|---------|
| Б-100   | Китаєв Олег     | Д-100   |
| Б-101   | Мельников Павел | Д-100   |
| Б-102   | Балкін Олег     | Д-101   |
| Б-103   | Епанчин Максим  | Д-101   |
| NULL    | NULL            | NULL    |

Рисунок 3.2 Таблиця *Бригади*

| idВантажоперевезення | Дата       | Машина      | Обсяг |
|----------------------|------------|-------------|-------|
| 1                    | 2017-06-06 | ЛАЗ-5255    | 100   |
| 2                    | 2017-06-06 | ЗІЛ-138     | 150   |
| 3                    | 2017-07-07 | ЛАЗ-5255    | 130   |
| 4                    | 2018-06-06 | ЛАЗ-5255    | 200   |
| 5                    | 2018-06-06 | ЗІЛ-138     | 250   |
| 6                    | 2019-06-06 | Богдан А091 | 250   |
| 7                    | 2019-07-07 | Богдан А091 | 350   |
| 8                    | 2019-08-08 | Богдан А091 | 125   |
| 9                    | 2019-10-10 | ЗІЛ-138     | 250   |
| 10                   | 2020-07-07 | ГАЗ-68      | 350   |
| NULL                 | NULL       | NULL        | NULL  |

Рисунок 3.3 Таблиця *Вантажоперевезення*

| idВодії | Імя                 | Бригада | Машина                 |
|---------|---------------------|---------|------------------------|
| 1000    | Єлисеїв Інокентій   | Б-100   | Рута 37                |
| 1001    | Вакулин Владислав   | Б-100   | Рута 37                |
| 1002    | Шорин Богдан        | Б-100   | ЛАЗ-699                |
| 1003    | Крушевський Аркадій | Б-101   | ЛАЗ-5255               |
| 1004    | Духовський Вадим    | Б-101   | Стрий Авто А0756       |
| 1005    | Шелай Валентин      | Б-101   | ЗІЛ-138                |
| 1006    | Каленов Вадим       | Б-102   | Богдан А091            |
| 1007    | Авербах Ярослав     | Б-102   | ГАЗ-68                 |
| 1008    | Наркевич Данило     | Б-102   | ГалАЗ-3208             |
| 1009    | Богаєвський Павло   | Б-103   | ЗІЛ-4331               |
| 1010    | Іванов Микита       | Б-103   | Mercedes-Benz Vito 638 |
| 1011    | Білоус Стас         | Б-103   | ГАЗ-3221               |
| NULL    | NULL                | NULL    | NULL                   |

Рисунок 3.4 Таблиця *Водії*

| idгаражні_господарства | Назва | Тип             |
|------------------------|-------|-----------------|
| 1                      | Г-1   | Автобус         |
| 2                      | Г-2   | Маршрутне таксі |
| 3                      | Г-3   | Вантажівка      |
| NULL                   | NULL  | NULL            |

Рисунок 3.5 Таблиця *Гаражні господарства*

| Ділянка | Начальник ділянки    |
|---------|----------------------|
| Д-100   | Радолинський Ярослав |
| Д-101   | Гридчин Данило       |
| NULL    | NULL                 |

Рисунок 3.6 Таблиця *Ділянки*

| idЗварники | Імя              | Бригада |
|------------|------------------|---------|
| 1          | Ачкасов Валерій  | Б-100   |
| 2          | Бедрин Руслан    | Б-101   |
| 3          | Андреянов Микита | Б-102   |
| 4          | Алабишев Ілля    | Б-103   |
| NULL       | NULL             | NULL    |

Рисунок 3.7 Таблиця *Зварники*

| idМаршрути | Машина                 | Маршрут |
|------------|------------------------|---------|
| 1          | ЛАЗ-699                | М-100   |
| 2          | Рута 37                | М-101   |
| 3          | Стрий Авто А0756       | М-102   |
| 4          | Mercedes-Benz Vito 638 | М-103   |
| 5          | ГАЗ-3221               | М-103   |
| NULL       | NULL                   | NULL    |

Рисунок 3.8 Таблиця *Маршрути*

| idПробіг | Дата       | Автомобіль             | Кілометраж |
|----------|------------|------------------------|------------|
| 1        | 2017-06-06 | ЛАЗ-5255               | 80         |
| 2        | 2017-06-06 | ЗІЛ-138                | 300        |
| 3        | 2017-07-07 | ЛАЗ-5255               | 100        |
| 4        | 2018-06-06 | ЛАЗ-5255               | 120        |
| 5        | 2018-06-06 | ЗІЛ-138                | 320        |
| 6        | 2019-06-06 | Рута 37                | 100        |
| 7        | 2019-06-06 | Богдан А091            | 70         |
| 8        | 2019-06-07 | Рута 37                | 120        |
| 9        | 2019-07-07 | Рута 37                | 150        |
| 10       | 2019-07-07 | Богдан А091            | 80         |
| 11       | 2019-07-08 | Стрий Авто А0756       | 180        |
| 12       | 2019-08-08 | Богдан А091            | 90         |
| 13       | 2019-10-10 | ЗІЛ-138                | 320        |
| 14       | 2020-07-07 | Рута 37                | 180        |
| 15       | 2020-04-04 | Mercedes-Benz Vito 638 | 25         |
| 16       | 2020-07-01 | ГАЗ-68                 | 350        |
| 17       | 2021-01-01 | ГАЗ-3221               | 70         |
| 18       | 2021-05-05 | ЛАЗ-699                | 90         |
| NULL     | NULL       | NULL                   | NULL       |

Рисунок 3.9 Таблиця *Пробіг*

| idРемонт | Машина      | Дата       | Вартість | Вузел/Агрегат | Зварник          | Слюсар          |
|----------|-------------|------------|----------|---------------|------------------|-----------------|
| 1        | Богдан-3355 | 2017-06-06 | 10000    | двигун        | Ачкасов Валерій  | Анохин Іван     |
| 2        | Рута 22     | 2017-06-06 | 5000     | КП            | Бедрин Руслан    | Дедянов Ярослав |
| 3        | Богдан А091 | 2017-07-06 | 10000    | двигун        | Андреянов Микита | Вальцев Данило  |
| 4        | Рута 37     | 2018-07-06 | 10000    | шасі          | Алабишев Ілля    | Нелидов Роман   |
| NULL     | NULL        | NULL       | NULL     | NULL          | NULL             | NULL            |

Рисунок 3.10 Таблиця *Ремонт*

| idСлюсарі | Імя             | Бригада |
|-----------|-----------------|---------|
| 1         | Анохин Іван     | Б-100   |
| 2         | Дедянов Ярослав | Б-101   |
| 3         | Вальцев Данило  | Б-102   |
| 4         | Нелидов Роман   | Б-103   |
| NULL      | NULL            | NULL    |

Рисунок 3.10 Таблиця *Слюсарі*

## ВИСНОВКИ

В результаті виконання курсової роботи було розроблено Базу Даних Автопідприємства. В ході роботи над програмним продуктом «БД Автопідприємства» було закріплено знання отримані під час вивчення курсу «Організація баз даних та знань», а також набуто навичок роботи з СУБД MySQL та середовищем розробки MySQL Workbench 6.3.

Всі пункти технічного завдання виконані. Розроблений програмний продукт задовольняє всі пункти, які поставив замовник, хоча і потребує деяких уточнень через некоректно поставлені умови в деяких пунктах.

В першому розділі було описана умова задачі, а також розглядались використані інструменти розробки.

В другому розділі було описано та створено спроектовано ієрархію БД на основі досліджень праць українських та закордонних науковців й віхівців – програмістів, а також власних знань, здобутих протягом курсу.

В третьому розділі було надано опис використаних таблиць, складено керівництво користувача даного продукту, а також проведено тестування на імітаційних даних, в процесі якого було виявлено декілька недоліків.

Надалі розробка може бути удосконалена шляхом додавання тригерів та процедур, які значно полегшить навігацію та роботу з БД. Також значним вдосконаленням може бути імплементація зручного інтерфейсу, який спростить роботу з «БД Автопідприємства» для некваліфікованих користувачів.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Elena B.. Реляционные СУБД – сравнение MySQL и SQL сервер. – Hostinger, 2018 [Электронный ресурс] - <https://www.hostinger.com.ua/rukovodstva/reljacionnye-subd-sravnenie-mysql-i-sql-server/#MySQL-SQL>
2. Troels Arvin. Comparison of different SQL implementations [Электронный ресурс] - <http://troels.arvin.dk/db/rdbms/>
3. Драч. В. Сравнение современных СУБД – 2017 [Электронный ресурс] - <http://drach.pro/blog/hi-tech/item/145-db-comparison>
4. Гусев А.В., Дмитриев А.Г.. Microsoft SQL Server против MySQL в медицинских информационных системах – 2004 [Электронный ресурс] - <http://citforum.ru/database/articles/msmysql/>
5. Microsoft. Довідник по Transact-SQL (компонент Database Engine) [Электронный ресурс] - <https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15>
6. Docs.Data Довідник по SQL. [Электронный ресурс] - <https://docs.data.world/documentation/sql/concepts/basic/intro.html>
7. Гарсиа-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс.: Пер. с англ. - М.: Издательский дом "Вильямс", 2004. - 1088 с.
8. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.
9. Голицина О.Л., Максимов Н.В., Попов И.И. Базы данных: навчальний посібник. – М.: ФОРУМ: ИНФРА-М, 2005. 352с: ил. (Профессиональное образование).
10. Види програм і програмних документів: ДСТ 19.101-77 ЕСПД
11. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення: ДСТУ 3008-95 (ГОСТ 7.32-91). – Чинний від 2006-01-01. К.: Держспоживстандарт України, 2006. – 231 с. – (Національний стандарт України).



12. Схеми алгоритмів і програм. Правила виконання. ДСТ 19.002-80 ЕСПД.
13. Посилання на GitHub репозиторій. [Електронний ресурс] - <https://github.com/AfJhAD/KR.git>

**Лістинг ПЗ****“REQUESTS\_BD”**

```
use mydb;

-- 1
delimiter $$
-- drop procedure if exists task1;
create procedure task1()
begin
    select Назва from Автомобілі;
end $$
delimiter $$
call task1()$$

-- 2_1
delimiter $$
-- drop procedure if exists task2_1;
create procedure task2_1()
begin
    select Імя from водії;
    select count(*) from водії;
end$$
DELIMITER $$
call task2_1()$$

-- 2_2
delimiter $$
-- drop procedure if exists task2_2;
create procedure task2_2(IN name_car VARCHAR(45))
begin
    select Імя as result
    from водії
    where Машина = name_car;
    select count(Імя)
    from водії
    where Машина = name_car;
end$$
DELIMITER $$
call task2_2('Рута 37')$$

-- 3
delimiter $$
-- drop procedure if exists task3;
create procedure task3()
begin
    select Імя, Машина from водії;
end$$
DELIMITER $$
call task3$$

-- 4
delimiter $$
-- drop procedure if exists task4;
create procedure task4()
begin
    select Машина, Маршрут from маршрути;
end$$
DELIMITER $$
```

```

call task4$$

-- 5_1
delimiter $$
-- drop procedure if exists task5_1;
create procedure task5_1(IN type_car VARCHAR(45))
begin
    select пробіг.Дата, пробіг.Кілометраж, автомобілі.Назва, автомобілі.Тип
    from пробіг
    inner join автомобілі
    on автомобілі.Назва = пробіг.Автомобіль
    where Тип = type_car;
end$$
DELIMITER $$
call task5_1('Автобус')$$

-- 5_2
delimiter $$
-- drop procedure if exists task5_2;
create procedure task5_2(IN car VARCHAR(45), IN date1 DATE, IN date2 DATE)
begin
    select пробіг.Дата, пробіг.Кілометраж, автомобілі.Назва, автомобілі.Тип
    from пробіг
    inner join автомобілі
    on автомобілі.Назва = пробіг.Автомобіль
    where пробіг.Дата between date1 and date2 AND пробіг.Автомобіль = car;
end$$
DELIMITER $$
call task5_2('Рута 37', '2019-07-06', '2020-08-06')$$

-- 6_1
delimiter $$
-- drop procedure if exists task6_1;
create procedure task6_1(IN type_car VARCHAR(45))
begin
    select count(idРемонт), sum(Вартість)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where Тип = type_car;
end$$
DELIMITER $$
call task6_1('Автобус')$$

-- 6_2
delimiter $$
-- drop procedure if exists task6_2;
create procedure task6_2(IN marka_car VARCHAR(45))
begin
    select count(idРемонт), sum(Вартість)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where Марка = marka_car;
end$$
DELIMITER $$
call task6_2('Богдан')$$

-- 6_3
delimiter $$
-- drop procedure if exists task6_3;
create procedure task6_3(IN car VARCHAR(45), IN date1 DATE, IN date2 DATE)
begin

```

```

        select count(idРемонт), sum(Вартість)
        from ремонт
        inner join автомобілі
        on автомобілі.назва = ремонт.машина
        where ремонт.Дата between date1 and date2 AND Машина = car;
end$$
DELIMITER $$
call task6_3('Богдан-3355', '2016-07-06', '2020-08-06')$$

-- 7
delimiter $$
-- drop procedure if exists task7;
create procedure task7(IN name_person varchar(45))
begin
    if (exists(select імя from водії where імя = name_person)) then begin

        select name_person
        union
        select Бригадир from бригади
        inner join водії on водії.Бригада = бригади.Бригада where водії.Імя =
name_person
        union
        select `Начальник ділянки` from ділянки
        inner join бригади on бригади.Ділянка = ділянки.Ділянка where бригади.Бригадир
= (
        select Бригадир from бригади
        inner join водії on водії.Бригада = бригади.Бригада where водії.Імя =
name_person)
        union
        select "Водій";
    end;

    elseif (exists(select імя from Зварники where імя = name_person)) then begin
        select name_person
        union
        select Бригадир from бригади
        inner join Зварники on Зварники.Бригада = бригади.Бригада where Зварники.Імя =
name_person
        union
        select `Начальник ділянки` from ділянки
        inner join бригади on бригади.Ділянка = ділянки.Ділянка where бригади.Бригадир
= (
        select Бригадир from бригади
        inner join Зварники on Зварники.Бригада = бригади.Бригада where Зварники.Імя =
name_person)
        union
        select "Зварювальник";
    end;

    elseif (exists(select імя from Слюсарі where імя = name_person)) then begin
        select name_person
        union
        select Бригадир from бригади
        inner join Слюсарі on Слюсарі.Бригада = бригади.Бригада where Слюсарі.Імя =
name_person
        union
        select `Начальник ділянки` from ділянки
        inner join бригади on бригади.Ділянка = ділянки.Ділянка where бригади.Бригадир
= (
        select Бригадир from бригади
        inner join Слюсарі on Слюсарі.Бригада = бригади.Бригада where Слюсарі.Імя =
name_person)
        union

```

```

        select "Слюсар";
    end;

    elseif (exists(select Бригадир from Бригади where Бригадир = name_person)) then begin
        select name_person
        union
        select `Начальник ділянки` from ділянки
        inner join Бригади on Бригади.ділянка = ділянки.ділянка where Бригади.Бригадир
= name_person
        union
        select "Бригадир";
    end;

    elseif (exists(select ділянка from Ділянки where `Начальник ділянки` = name_person))
then begin
        select name_person
        union
        select "Начальник ділянки";
    end;

    else select "some problems";
end if;
end$$
DELIMITER $$
call task7('Шорин Богдан')$$

-- 8_1
delimiter $$
-- drop procedure if exists task8_1;
create procedure task8_1()
begin
    select if((select count(*) from гаражні_господарства)!=0, 'гаражні господарства є',
'гаражних господарств немає');
end$$
DELIMITER $$
call task8_1()$$

-- 8_2
delimiter $$
-- drop procedure if exists task8_2;
create procedure task8_2(IN type_car varchar(45))
begin
    select if((select count(*) from гаражні_господарства where Тип = type_car)!=0,
'гаражне господарство такого типу є', 'гаражного господарства такого типу немає');
end$$
DELIMITER $$
call task8_2('Таксі')$$

-- 9
delimiter $$
-- drop procedure if exists task9;
create procedure task9()
begin
    select автомобілі.Назва, автомобілі.Тип, гаражні_господарства.Назва as `Гаражне
господарство` from автомобілі
    inner join гаражні_господарства on гаражні_господарства.Тип = автомобілі.Тип;
end$$
DELIMITER $$
call task9()$$

-- 10
delimiter $$
-- drop procedure if exists task10;

```

```

create procedure task10(IN car VARCHAR(45), IN date1 DATE, IN date2 DATE)
begin
    select * from вантажоперевезення
    where Дата between date1 and date2 AND Машина = car;
end$$
DELIMITER $$
call task10('ЗІЛ-138', '2017-07-06', '2020-08-06')$$

-- 11_1
delimiter $$
-- drop procedure if exists task11_1;
create procedure task11_1(IN element_car VARCHAR(45), IN type_car varchar(45))
begin
    select count(idРемонт)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where `ремонт`.`Вузел/Агрегат` = element_car AND Тип = type_car;
end$$
DELIMITER $$
call task11_1('шасі', 'Автобус')$$

-- 11_2
delimiter $$
-- drop procedure if exists task11_2;
create procedure task11_2(IN element_car VARCHAR(45), IN marka_car varchar(45))
begin
    select count(idРемонт)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where `ремонт`.`Вузел/Агрегат` = element_car AND Марка = marka_car;
end$$
DELIMITER $$
call task11_2('шасі', 'Рута')$$

-- 11_3
delimiter $$
-- drop procedure if exists task11_3;
create procedure task11_3(IN element_car varchar(45), IN car VARCHAR(45), IN date1 DATE, IN
date2 DATE)
begin
    select count(idРемонт)
    from ремонт
    inner join автомобілі
    on автомобілі.назва = ремонт.машина
    where Дата between date1 and date2 AND Машина = car AND `ремонт`.`Вузел/Агрегат` =
element_car;
end$$
DELIMITER $$
call task11_3('двигун', 'Богдан-3355', '2016-07-06', '2020-08-06')$$

-- 12
delimiter $$
-- drop procedure if exists task12;
create procedure task12(IN date1 DATE, IN date2 DATE)
begin
    select Назва, `Дата появи`, `Дата списання` from автомобілі
    where `Дата появи` between date1 and date2 OR `Дата списання` between date1 and date2;
end$$
DELIMITER $$
call task12('2017-07-23', '2017-08-23')$$

```

```

-- 13_1
delimiter $$
-- drop procedure if exists task13;
create procedure task13(IN person_name varchar(45))
begin
    if (exists(select Бригадир from Бригади where Бригадир = person_name)) then begin
        select "Бригадир"
        union
        select Імя from водії
        inner join бригади on водії.Бригада = бригади.Бригада where Бригадир =
person_name
        union
        select Імя from зварники
        inner join бригади on зварники.Бригада = бригади.Бригада where Бригадир =
person_name
        union
        select Імя from слюсарі
        inner join бригади on слюсарі.Бригада = бригади.Бригада where Бригадир =
person_name;
    end;

    elseif (exists(select `Начальник ділянки` from ділянки where `Начальник ділянки` =
person_name)) then begin
        select "Начальник ділянки"
        union
        select Бригадир from Бригади
        inner join ділянки on бригади.Ділянка = ділянки.Ділянка where `Начальник
ділянки` = person_name;
    end;

    else select "some problems";
    end if;
end$$
DELIMITER $$
call task13('Китаєв Олег')$$

-- 14_1
delimiter $$
-- drop procedure if exists task14_1;
create procedure task14_1(IN name_worker VARCHAR(45), IN car varchar(45))
begin
    select Дата, Машина, `Вузел/Агрегат` from ремонт
    where (Зварник = name_worker and Машина = car) or (Слюсар = name_worker and Машина =
car) ;
end$$
DELIMITER $$
call task14_1('Ачкасов Валерій', 'Богдан-3355')$$

-- 14_2
delimiter $$
-- drop procedure if exists task14_2;
create procedure task14_2(IN name_person VARCHAR(45), IN date1 DATE, IN date2 DATE)
begin
    select Дата, Машина, `Вузел/Агрегат` from ремонт
    where Дата between date1 and date2 AND (Зварник = name_person or Слюсар = name_person);
end$$
DELIMITER $$
call task14_2('Анохин Іван', '2016-07-06', '2020-08-06')$$

```

## “CREATE\_BD”

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@ @UNIQUE_CHECKS,
UNIQUE_CHECKS=0;
```

```
SET @OLD_FOREIGN_KEY_CHECKS=@ @FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
```

```
SET @OLD_SQL_MODE=@ @SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
```

-----

-- Schema mydb

-----

```
DROP SCHEMA IF EXISTS `mydb` ;
```

-----

-- Schema mydb

-----

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

-----

-- Table `mydb`.`гаражні\_господарства`

-----

```
DROP TABLE IF EXISTS `mydb`.`гаражні_господарства` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`гаражні_господарства` (
  `idгаражні_господарства` INT NOT NULL,
  `Назва` VARCHAR(45) NOT NULL,
```



```

`Тип` VARCHAR(45) NOT NULL,
PRIMARY KEY (`Тип`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `mydb`.`Автомобілі`
-----

```

```

DROP TABLE IF EXISTS `mydb`.`Автомобілі` ;

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Автомобілі` (
  `idАвтомобілі` INT NOT NULL,
  `Назва` VARCHAR(45) NOT NULL,
  `Марка` VARCHAR(45) NOT NULL,
  `Дата появи` DATE NOT NULL,
  `Дата списання` DATE NULL,
  `Тип` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Назва`),
  CONSTRAINT `г1`
  FOREIGN KEY (`Тип`)
  REFERENCES `mydb`.`гаражні_господарства` (`Тип`)
  ON DELETE CASCADE
  ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```

CREATE INDEX `г1_idx` ON `mydb`.`Автомобілі` (`Тип` ASC);

```

```
-----  
-- Table `mydb`.`ділянки`  
-----  
  
DROP TABLE IF EXISTS `mydb`.`ділянки` ;  
  
CREATE TABLE IF NOT EXISTS `mydb`.`ділянки` (  
  `Ділянка` VARCHAR(45) NOT NULL,  
  `Начальник ділянки` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Ділянка`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;  
  
-----  
-- Table `mydb`.`бригади`  
-----  
  
DROP TABLE IF EXISTS `mydb`.`бригади` ;  
  
CREATE TABLE IF NOT EXISTS `mydb`.`бригади` (  
  `Бригада` VARCHAR(45) NOT NULL,  
  `Бригадир` VARCHAR(45) NOT NULL,  
  `Ділянка` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Бригада`),  
  CONSTRAINT `д1`  
  FOREIGN KEY (`Ділянка`)  
  REFERENCES `mydb`.`ділянки` (`Ділянка`)
```

```

    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

CREATE INDEX `д1_idx` ON `mydb`.`бригади` (`Ділянка` ASC);

-----
-- Table `mydb`.`вантажоперевезення`
-----

DROP TABLE IF EXISTS `mydb`.`вантажоперевезення` ;

CREATE TABLE IF NOT EXISTS `mydb`.`вантажоперевезення` (
  `idВантажоперевезення` INT NOT NULL,
  `Дата` DATE NOT NULL,
  `Машина` VARCHAR(45) NOT NULL,
  `Обсяг` INT NOT NULL,
  PRIMARY KEY (`idВантажоперевезення`),
  CONSTRAINT `a4`
    FOREIGN KEY (`Машина`)
    REFERENCES `mydb`.`Автомобілі` (`Назва`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

CREATE INDEX `a4_idx` ON `mydb`.`вантажоперевезення` (`Машина` ASC);

```

```

-----
-- Table `mydb`.`водії`
-----

DROP TABLE IF EXISTS `mydb`.`водії` ;

CREATE TABLE IF NOT EXISTS `mydb`.`водії` (
  `idВодії` INT NOT NULL,
  `Імя` VARCHAR(45) NOT NULL,
  `Бригада` VARCHAR(45) NOT NULL,
  `Машина` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idВодії`),
  CONSTRAINT `a1`
    FOREIGN KEY (`Машина`)
    REFERENCES `mydb`.`Автомобілі` (`Назва`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `б1`
    FOREIGN KEY (`Бригада`)
    REFERENCES `mydb`.`бригади` (`Бригада`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

CREATE INDEX `a1_idx` ON `mydb`.`водії` (`Машина` ASC);

```

```
CREATE INDEX `61_idx` ON `mydb`.`водії` (`Бригада` ASC);
```

```
-----  
-- Table `mydb`.`зварники`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`зварники` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`зварники` (
```

```
  `idЗварники` INT NOT NULL,
```

```
  `Імя` VARCHAR(45) NOT NULL,
```

```
  `Бригада` VARCHAR(45) NOT NULL,
```

```
  PRIMARY KEY (`Імя`),
```

```
  CONSTRAINT `62`
```

```
    FOREIGN KEY (`Бригада`)
```

```
    REFERENCES `mydb`.`бригади` (`Бригада`)
```

```
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE)
```

```
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8;
```

```
CREATE INDEX `62_idx` ON `mydb`.`зварники` (`Бригада` ASC);
```

```
-----  
-- Table `mydb`.`маршрути`  
-----
```

```
DROP TABLE IF EXISTS `mydb`.`маршрути` ;
```

```

CREATE TABLE IF NOT EXISTS `mydb`.`маршрути` (
  `idМаршрути` INT NOT NULL,
  `Машина` VARCHAR(45) NOT NULL,
  `Маршрут` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idМаршрути`),
  CONSTRAINT `a2`
  FOREIGN KEY (`Машина`)
  REFERENCES `mydb`.`Автомобілі` (`Назва`)
  ON DELETE CASCADE
  ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

CREATE INDEX `a2_idx` ON `mydb`.`маршрути` (`Машина` ASC);

-----
-- Table `mydb`.`пробіг`
-----

DROP TABLE IF EXISTS `mydb`.`пробіг` ;

CREATE TABLE IF NOT EXISTS `mydb`.`пробіг` (
  `idПробіг` INT NOT NULL,
  `Дата` DATE NOT NULL,
  `Автомобіль` VARCHAR(45) NOT NULL,
  `Кілометраж` INT NOT NULL,
  PRIMARY KEY (`idПробіг`),

```

```

CONSTRAINT `a3`
  FOREIGN KEY (`Автомобіль`)
  REFERENCES `mydb`.`Автомобілі` (`Назва`)
  ON DELETE CASCADE
  ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

CREATE INDEX `a3_idx` ON `mydb`.`пробіг` (`Автомобіль` ASC);

-----
-- Table `mydb`.`слюсарі`
-----

DROP TABLE IF EXISTS `mydb`.`слюсарі` ;

CREATE TABLE IF NOT EXISTS `mydb`.`слюсарі` (
  `idСлюсарі` INT NOT NULL,
  `Імя` VARCHAR(45) NOT NULL,
  `Бригада` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Імя`),
  CONSTRAINT `б3`
    FOREIGN KEY (`Бригада`)
    REFERENCES `mydb`.`бригади` (`Бригада`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

```

```
CREATE INDEX `63_idx` ON `mydb`.`слюсарі` (`Бригада` ASC);
```

```
-----
```

```
-- Table `mydb`.`ремонт`
```

```
-----
```

```
DROP TABLE IF EXISTS `mydb`.`ремонт` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`ремонт` (
```

```
  `idРемонт` INT NOT NULL,
```

```
  `Машина` VARCHAR(45) NOT NULL,
```

```
  `Дата` DATE NOT NULL,
```

```
  `Вартість` INT NOT NULL,
```

```
  `Вузел/Агрегат` VARCHAR(45) NOT NULL,
```

```
  `Зварник` VARCHAR(45) NOT NULL,
```

```
  `Слюсар` VARCHAR(45) NOT NULL,
```

```
  PRIMARY KEY (`idРемонт`),
```

```
  CONSTRAINT `a5`
```

```
    FOREIGN KEY (`Машина`)
```

```
    REFERENCES `mydb`.`Автомобілі` (`Назва`)
```

```
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE,
```

```
  CONSTRAINT `c1`
```

```
    FOREIGN KEY (`Слюсар`)
```

```
    REFERENCES `mydb`.`слюсарі` (`Імя`)
```

```
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE,
```



```
CONSTRAINT `з1`  
  FOREIGN KEY (`Зварник`)  
  REFERENCES `mydb`.`зварники` (`Імя`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;  
  
CREATE INDEX `a5_idx` ON `mydb`.`ремонт` (`Машина` ASC);  
  
CREATE INDEX `c1_idx` ON `mydb`.`ремонт` (`Слюсар` ASC);  
  
CREATE INDEX `з1_idx` ON `mydb`.`ремонт` (`Зварник` ASC);
```

**"INSERT\_BD"**

```
SET SQL_MODE=@OLD_SQL_MODE;
```

```
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
```

```
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

```
-----  
-- Data for table `mydb`.`гаражні_господарства`  
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`гаражні_господарства` (`idгаражні_господарства`,  
`Назва`, `Тип`) VALUES (1, 'Г-1', 'Автобус');
```

```
INSERT INTO `mydb`.`гаражні_господарства` (`idгаражні_господарства`,  
`Назва`, `Тип`) VALUES (2, 'Г-2', 'Маршрутне таксі');
```

```
INSERT INTO `mydb`.`гаражні_господарства` (`idгаражні_господарства`,  
`Назва`, `Тип`) VALUES (3, 'Г-3', 'Вантажівка');
```

```
COMMIT;
```

```
-----  
-- Data for table `mydb`.`Автомобілі`  
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата  
появи`, `Дата списання`, `Тип`) VALUES (200, 'ЛАЗ-697', 'ЛАЗ', '2015-07-23',  
'2018-08-23', 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (201, 'Рута 22', 'Рута', '2015-07-23', '2018-08-23', 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (202, 'Рута 23', 'Рута', '2015-08-23', '2018-07-23', 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (203, 'Стрий Авто А075', 'Стрий Авто', '2016-07-23', NULL, 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (204, 'Стрий Авто А0756', 'Стрий Авто', '2016-07-23', NULL, 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (205, 'ЛАЗ-699', 'ЛАЗ', '2017-07-23', NULL, 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (206, 'ГалАЗ-3208', 'ГалАЗ', '2017-07-23', NULL, 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (207, 'Рута 37', 'Рута', '2018-07-23', NULL, 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (208, 'ГалАЗ-3209', 'ГалАЗ', '2018-08-23', NULL, 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (209, 'ЛАЗ-5255', 'ЛАЗ', '2018-08-23', NULL, 'Автобус');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (210, 'Богдан-3355', 'Богдан', '2015-07-23', '2018-07-23', 'Вантажівка');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата появи`, `Дата списання`, `Тип`) VALUES (211, 'ГАЗ-51', 'ГАЗ', '2015-07-23', '2018-07-23', 'Вантажівка');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (212, 'ГАЗ-68', 'ГАЗ', '2015-07-23',
NULL, 'Вантажівка');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (213, 'ЗІЛ-138', 'ЗІЛ', '2016-07-23',
NULL, 'Вантажівка');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (214, 'ЗІЛ-4331', 'ЗІЛ', '2016-07-23',
NULL, 'Вантажівка');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (215, 'Богдан А091', 'Богдан', '2016-07-
23', NULL, 'Вантажівка');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (216, 'БАЗ-2215', 'БАЗ', '2015-07-23',
'2018-06-23', 'Маршрутне таксі');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (217, 'ГАЗ-3221', 'ГАЗ', '2015-07-23',
NULL, 'Маршрутне таксі');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (218, 'ЧАЗ-983', 'ЧАЗ', '2015-07-23',
NULL, 'Маршрутне таксі');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (219, 'Mercedes-Benz Vito 638',
'Mercedes-Benz', '2016-07-23', NULL, 'Маршрутне таксі');
```

```
INSERT INTO `mydb`.`Автомобілі` (`idАвтомобілі`, `Назва`, `Марка`, `Дата
появи`, `Дата списання`, `Тип`) VALUES (220, 'Mercedes-Benz Vito 639',
'Mercedes-Benz', '2016-07-23', NULL, 'Маршрутне таксі');
```

```
COMMIT;
```

```
-----
```

```
-- Data for table `mydb`.`ділянки`
```

```
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`ділянки` (`Ділянка`, `Начальник ділянки`) VALUES ('Д-100', 'Радолинський Ярослав');
```

```
INSERT INTO `mydb`.`ділянки` (`Ділянка`, `Начальник ділянки`) VALUES ('Д-101', 'Тридчин Данило');
```

```
COMMIT;
```

```
-----  
-- Data for table `mydb`.`бригади`  
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`бригади` (`Бригада`, `Бригадир`, `Ділянка`) VALUES ('Б-100', 'Китаєв Олег', 'Д-100');
```

```
INSERT INTO `mydb`.`бригади` (`Бригада`, `Бригадир`, `Ділянка`) VALUES ('Б-101', 'Мельников Павел', 'Д-100');
```

```
INSERT INTO `mydb`.`бригади` (`Бригада`, `Бригадир`, `Ділянка`) VALUES ('Б-102', 'Балкін Олег', 'Д-101');
```

```
INSERT INTO `mydb`.`бригади` (`Бригада`, `Бригадир`, `Ділянка`) VALUES ('Б-103', 'Епанчин Максим', 'Д-101');
```

```
COMMIT;
```

```
-----  
-- Data for table `mydb`.`вантажоперевезення`  
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (1, '2017-06-06', 'ЛАЗ-5255', 100);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (2, '2017-06-06', 'ЗІЛ-138', 150);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (3, '2017-07-07', 'ЛАЗ-5255', 130);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (4, '2018-06-06', 'ЛАЗ-5255', 200);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (5, '2018-06-06', 'ЗІЛ-138', 250);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (6, '2019-06-06', 'Богдан А091', 250);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (7, '2019-07-07', 'Богдан А091', 350);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (8, '2019-08-08', 'Богдан А091', 125);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (9, '2019-10-10', 'ЗІЛ-138', 250);
```

```
INSERT INTO `mydb`.`вантажоперевезення` (`idВантажоперевезення`, `Дата`,  
`Машина`, `Обсяг`) VALUES (10, '2020-07-07', 'ГАЗ-68', 350);
```

```
COMMIT;
```

```
-----  
-- Data for table `mydb`.`водії`  
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1000, 'Єлисеїв Інокентій', 'Б-100', 'Рута 37');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1001, 'Вакулин Владислав', 'Б-100', 'Рута 37');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1002, 'Шорин Богдан', 'Б-100', 'ЛАЗ-699');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1003, 'Крушевський Аркадій', 'Б-101', 'ЛАЗ-5255');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1004, 'Духовський Вадим', 'Б-101', 'Стрий Авто А0756');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1005, 'Шелай Валентин', 'Б-101', 'ЗІЛ-138');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1006, 'Каленов Вадим', 'Б-102', 'Богдан А091');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1007, 'Авербах Ярослав', 'Б-102', 'ГАЗ-68');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1008, 'Наркевич Данило', 'Б-102', 'ГалАЗ-3208');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1009, 'Богаєвський Павло', 'Б-103', 'ЗІЛ-4331');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1010, 'Іванов Микита', 'Б-103', 'Mercedes-Benz Vito 638');
```

```
INSERT INTO `mydb`.`водії` (`idВодії`, `Імя`, `Бригада`, `Машина`) VALUES
(1011, 'Білоус Стас', 'Б-103', 'ГАЗ-3221');
```

```
COMMIT;
```

```
-----
-- Data for table `mydb`.`зварники`
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;

INSERT INTO `mydb`.`зварники` (`idЗварники`, `Імя`, `Бригада`) VALUES (1,
'Ачкасов Валерій', 'Б-100');

INSERT INTO `mydb`.`зварники` (`idЗварники`, `Імя`, `Бригада`) VALUES (2,
'Бедрин Руслан', 'Б-101');

INSERT INTO `mydb`.`зварники` (`idЗварники`, `Імя`, `Бригада`) VALUES (3,
'Андреянов Микита', 'Б-102');

INSERT INTO `mydb`.`зварники` (`idЗварники`, `Імя`, `Бригада`) VALUES (4,
'Алабишев Ілля', 'Б-103');
```

```
COMMIT;
```

```
-----
-- Data for table `mydb`.`маршрути`
-----
```

```
START TRANSACTION;

USE `mydb`;

INSERT INTO `mydb`.`маршрути` (`idМаршрути`, `Машина`, `Маршрут`)
VALUES (1, 'ЛІАЗ-699', 'М-100');

INSERT INTO `mydb`.`маршрути` (`idМаршрути`, `Машина`, `Маршрут`)
VALUES (2, 'Рута 37', 'М-101');

INSERT INTO `mydb`.`маршрути` (`idМаршрути`, `Машина`, `Маршрут`)
VALUES (3, 'Стрий Авто А0756', 'М-102');

INSERT INTO `mydb`.`маршрути` (`idМаршрути`, `Машина`, `Маршрут`)
VALUES (4, 'Mercedes-Benz Vito 638', 'М-103');

INSERT INTO `mydb`.`маршрути` (`idМаршрути`, `Машина`, `Маршрут`)
VALUES (5, 'ГАЗ-3221', 'М-103');

COMMIT;
```



```
-----
-- Data for table `mydb`.`пробіг`
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (1, '2017-06-06', 'ЛІАЗ-5255', 80);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (2, '2017-06-06', 'ЗІЛ-138', 300);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (3, '2017-07-07', 'ЛІАЗ-5255', 100);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (4, '2018-06-06', 'ЛІАЗ-5255', 120);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (5, '2018-06-06', 'ЗІЛ-138', 320);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (6, '2019-06-06', 'Рута 37', 100);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (7, '2019-06-06', 'Богдан А091', 70);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (8, '2019-06-07', 'Рута 37', 120);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (9, '2019-07-07', 'Рута 37', 150);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (10, '2019-07-07', 'Богдан А091', 80);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (11, '2019-07-08', 'Стрий Авто А0756', 180);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (12, '2019-08-08', 'Богдан А091', 90);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (13, '2019-10-10', 'ЗІЛ-138', 320);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (14, '2020-07-07', 'Рута 37', 180);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (15, '2020-04-04', 'Mercedes-Benz Vito 638', 25);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (16, '2020-07-01', 'ТАЗ-68', 350);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (17, '2021-01-01', 'ТАЗ-3221', 70);
```

```
INSERT INTO `mydb`.`пробіг` (`idПробіг`, `Дата`, `Автомобіль`, `Кілометраж`)
VALUES (18, '2021-05-05', 'ЛАЗ-699', 90);
```

```
COMMIT;
```

```
-----
-- Data for table `mydb`.`слюсарі`
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`слюсарі` (`idСлюсарі`, `Імя`, `Бригада`) VALUES (1,
'Анохин Іван', 'Б-100');
```

```
INSERT INTO `mydb`.`слюсарі` (`idСлюсарі`, `Імя`, `Бригада`) VALUES (2,
'Десянов Ярослав', 'Б-101');
```

```
INSERT INTO `mydb`.`слюсарі` (`idСлюсарі`, `Імя`, `Бригада`) VALUES (3,
'Вальцев Данило', 'Б-102');
```

```
INSERT INTO `mydb`.`слюсарі` (`idСлюсарі`, `Імя`, `Бригада`) VALUES (4,
'Нелидов Роман', 'Б-103');
```

```
COMMIT;
```

```
-----  
-- Data for table `mydb`.`ремонт`  
-----
```

```
START TRANSACTION;
```

```
USE `mydb`;
```

```
INSERT INTO `mydb`.`ремонт` (`idРемонт`, `Машина`, `Дата`, `Вартість`,  
`Вузел/Агрегат`, `Зварник`, `Слюсар`) VALUES (1, 'Богдан-3355', '2017-06-06',  
10000, 'двигун', 'Ачкасов Валерій', 'Анохин Іван');
```

```
INSERT INTO `mydb`.`ремонт` (`idРемонт`, `Машина`, `Дата`, `Вартість`,  
`Вузел/Агрегат`, `Зварник`, `Слюсар`) VALUES (2, 'Рута 22', '2017-06-06', 5000,  
'КІТ', 'Бедрин Руслан', 'Дедянов Ярослав');
```

```
INSERT INTO `mydb`.`ремонт` (`idРемонт`, `Машина`, `Дата`, `Вартість`,  
`Вузел/Агрегат`, `Зварник`, `Слюсар`) VALUES (3, 'Богдан А091', '2017-07-06',  
10000, 'двигун', 'Андреянов Микита', 'Вальцев Данило');
```

```
INSERT INTO `mydb`.`ремонт` (`idРемонт`, `Машина`, `Дата`, `Вартість`,  
`Вузел/Агрегат`, `Зварник`, `Слюсар`) VALUES (4, 'Рута 37', '2018-07-06', 10000,  
'шасі', 'Алабишев Ілля', 'Нелидов Роман');
```

```
COMMIT;
```