**Анотація**

Пояснювальна записка до курсової роботи: 58 сторінок, 136 рисунків, 12

таблиць, 4 посилання

Об’єкт дослідження: База даних маршрутів громадського транспорту міста.

Мета роботи: закріплення навичок з проектування, реалізації реляційних баз даних та їх використання для практичних задач.

Проведено аналіз предметного середовища, визначено сутності, атрибути та зв’язки між об’єктами. Побудована ER-модель предметного середовища, реляційна схема бази даних, виділено первинні та зовнішні ключі, визначено обмеження для підтримки цілісності даних. Розроблено скрипти для побудови спроектованої бази даних, імпортовано дані в неї, виконано різноманітні запити до неї.

**Зміст**

[ВСТУП 4](#_Toc154680377)

[1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА 5](#_Toc154680378)

[1.1 Опис об’єкту дослідження 5](#_Toc154680379)

[1.2 Опис вхідних та вихідних даних 6](#_Toc154680380)

[1.3 Основні бізнес процеси використання даних 6](#_Toc154680381)

[2 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ 7](#_Toc154680382)

[3 ПОСТАНОВКА ЗАВДАННЯ 9](#_Toc154680383)

[3.1 Мета 9](#_Toc154680384)

[3.2 Основні вимоги 9](#_Toc154680385)

[3.3 Завдання 10](#_Toc154680386)

[4 ПОБУДОВА ER-МОДЕЛІ 10](#_Toc154680387)

[4.1 Опис виділених інформаційних об’єктів 10](#_Toc154680388)

[4.2 Опис відношень 11](#_Toc154680389)

[4.3 Опис користувачів 13](#_Toc154680390)

[4.4 Опис бізнес-правил 14](#_Toc154680391)

[4.5 ER-діаграма 15](#_Toc154680392)

[5 РЕАЛІЗАЦІЯ БАЗИ ДАНИХ 16](#_Toc154680393)

[5.1 Обґрунтування вибору СУБД 16](#_Toc154680394)

[5.2 Опис структури у табличному вигляді 16](#_Toc154680395)

[5.3 SQL скрипт 23](#_Toc154680396)

[5.4 Схема бази даних 25](#_Toc154680397)

[6 РОБОТА З БАЗОЮ ДАНИХ 25](#_Toc154680398)

[6.1 Тексти вставки даних 25](#_Toc154680399)

[6.2 Тексти збережених процедур/функцій 29](#_Toc154680400)

[6.3 Тексти тригерів 36](#_Toc154680401)

[6.4 Тексти представлень 41](#_Toc154680402)

[6.5 SQL-запити 43](#_Toc154680403)

[6.6 Результати оптимізації 52](#_Toc154680404)

[6.7 Створення користувачів 54](#_Toc154680405)

[ВИСНОВКИ 56](#_Toc154680406)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 57](#_Toc154680407)

# ВСТУП

У сучасному інформаційному суспільстві, де міста постійно зростають та розвиваються, ефективна організація громадського транспорту стає важливим аспектом комфортного життя мешканців та гостей міст. Інтенсивний ритм життя вимагає швидких та точних рішень у сфері транспортної логістики. В цьому контексті, впровадження баз даних для управління маршрутами громадського транспорту стає необхідністю, оскільки вона сприяє оптимізації та підвищенню ефективності транспортної системи.

Актуальність теми полягає у вирішенні проблеми організації громадського транспорту через використання сучасних технологій управління інформацією. Застосування баз даних у цій галузі дозволяє не лише зберігати та організовувати великі обсяги даних, але й забезпечує зручний доступ до них для прийняття обґрунтованих управлінських рішень.

Метою даної курсової роботи є вивчення, аналіз та розробка бази даних для ефективного управління маршрутами громадського транспорту міста. Задачі включають в себе визначення потреб системи громадського транспорту, проектування структури бази даних, розробку самої бази даних, а також тестування та вдосконалення розробленої системи.

Призначення отриманої розробки полягає в покращенні управління громадським транспортом міста, а саме: зручного планування та оптимізації маршрутів, розрахунок часу та відстані поїздки, моніторинг роботи системи, надання актуальної інформації пасажирам і багато іншого.

Ця робота знаходить широкий застосунок у сферах транспортної індустрії та управління міськими послугами, оскільки впровадження баз даних для управління маршрутами громадського транспорту може мати широкий вплив на розвиток транспортної інфраструктури міста та сприяти покращенню якості життя його мешканців.

# ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА

## Опис об’єкту дослідження

Об’єктом дослідження даної курсової роботи є база даних, розроблена для управління маршрутами громадського транспорту міста. Цей інформаційний компонент є ключовим елементом системи управління транспортною логістикою та надає можливість оптимізації та контролю над рухом громадського транспорту в місті.

База даних складається з низки взаємозалежних таблиць, які охоплюють різні аспекти громадського транспорту, такі як маршрути, зупинки, типи зупинок, розклади, транспортні засоби, типи транспортних засобів, технічні обслуговування транспортних засобів, працівники та контракти працівників.

Загалом об’єкт дослідження може використовуватися для різних цілей, таких як: оптимізації існуючих маршрутів, шляхом визначення найбільш затребуваних, забезпечення працездатності та комфортного використання транспортної інфраструктури міста, шляхом перевірки стану транспортних засобів, що використовуються для обслуговування міста, а також їх депо, що використовуються для зберігання. Також важливим пунктом є надання актуальної інформації користувачам громадського транспорту міста, а саме інформація щодо графіків руху, затримки руху, початок та закінчення роботи кожного дня.

Дослідження об’єкту бази даних для управління маршрутами громадського транспорту дозволяє визначити ефективні способи впровадження та підтримки системи, спрямованої на поліпшення організації та функціонування громадського транспорту в місті.

## Опис вхідних та вихідних даних

Використовуючи дані та відомості про предметну область, були виділені такі вхідні дані:

- Інформація про типи транспортів

- Інформація про наявні транспорти в місті

- Інформація про працівників

- Інформація про контракти працівників

- Інформація про обслуговування транспорту

- Інформація про зупинки

- Інформація про маршрути

- Інформація про депо

- Інформація про графіки

Вихідними даними є результати обробки вхідних даних за допомогою

Запитів, процедур, функцій та представлень.

## Основні бізнес процеси використання даних

В контексті управління маршрутами громадського транспорту база даних використовується для підтримки ключових бізнес-процесів, спрямованих на оптимізацію роботи транспортної системи та підвищення якості обслуговування пасажирів. Основні бізнес-процеси включають:

* **Планування маршрутів і розкладів:**

Опис процесу: Система може використовуватись для планування оптимальних маршрутів і розкладів громадського транспорту міста, за допомогою даних про графіки роботи транспортних засобів, інформацію про зупинки, маршрути та транспортні засоби.

Мета: Зменшення часу очікування для пасажирів, оптимізація використання транспортних ресурсів.

* **Технічне обслуговування транспортних засобів:**

Опис процесу: Використання даних про технічний стан транспортних засобів для планування та виконання регулярного технічного обслуговування.

Мета: Забезпечення безперебійності роботи транспортної системи та підвищення терміну служби транспортних засобів.

* **Зручне використання бази даних для управління маршрутами громадського транспорту:**

Опис процесу: База даних розроблена таким чином, щоб бути гнучкою та легко адаптуватися до різних потреб та вимог системи громадського транспорту. Можливість налаштування параметрів, прав доступу та інших аспектів сприяє зручному використанню.

Мета: Сприяння інтуїтивної та зрозумілої роботи з базою даних, для полегшення роботи.

* **Оптимізація роботи:**

Опис процесу: Система забезпечує швидкий та ефективний доступ до потрібної інформації. Запити до бази даних оптимізовані для швидкого виконання, а результуючі дані надаються у зручному форматі для подальшого використання.

Мета: оптимізація рутинних процесів та пришвидшення доступу та роботи з базою даних.

# АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ

Для оптимізації управління маршрутами громадського транспорту та надання користувачам зручних сервісів використовується програмне рішення, спрямоване на автоматизацію функцій, пов'язаних із обслуговуванням маршрутів та пасажирів. Для аналізу програмних продуктів у цьому контексті розглядаються платформи, що дають актуальну інформацію про маршрути громадського транспорту в Києві. Ці платформи орієнтовані на взаємодію з користувачами та надання актуальної інформації щодо маршрутів і розкладів руху, що відповідає специфіці програмного середовища в даній сфері.

**Про компанію "easyWay":**

Компанія "easyWay" спеціалізується на розробці та підтримці інформаційних платформ для громадського транспорту. Метою компанії є полегшення мобільності та забезпечення доступу до актуальної інформації для пасажирів.

**Плюси сайту "eWay":**

1. Інтуїтивний та простий інтерфейс робить веб-сайт "eWay" дуже зручним у використанні для різних категорій користувачів.
2. Наявність інтерактивної карти дозволяє візуалізувати маршрути та зрозуміти їхнє розташування.
3. Сайт забезпечує актуальну інформацію про розклади та рух громадського транспорту.

**Мінуси сайту "eWay":**

1. Відсутність мобільного додатку може обмежити зручність користування для користувачів, які використовують мобільні пристрої.
2. Іноді можливі труднощі зі стабільністю роботи сайту, особливо в періоди пікового навантаження.

**Про компанію "Moovit":**

Компанія "Moovit" відома своєю спеціалізацією на розробці та управлінні інформаційними платформами для громадського транспорту. Її метою є забезпечення ефективності мобільності та забезпечення пасажирів актуальною інформацією для комфортного користування громадським транспортом.

**Плюси сайту "Moovit":**

1. Наявність інтерактивної карти та точної інформації про маршрути дозволяє пасажирам легко планувати свої подорожі.
2. "Moovit" пропонує багатофункціональний підхід, що включає в себе інформацію про розклади, оповіщення та інші корисні сервіси.
3. Наявність мобільного додатку робить взаємодію з платформою "Moovit" більш зручною та доступною для користувачів, що використовують мобільні пристрої.

**Мінуси сайту "Moovit":**

1. Як і в багатьох інших платформах, можливі труднощі зі стабільністю роботи, особливо в періоди пікового навантаження.
2. У порівнянні з конкурентами може відсутність окремих додаткових функцій або сервісів.

Отже "Easyway" та "Moovit" використовують базу даних для ефективного зберігання та швидкого доступу до інформації про маршрути громадського транспорту. Дані про розклади, графіки руху, маршрути, зупинки та інші важливі елементи транспортної системи зберігаються у структурованій базі даних. Це дозволяє забезпечити швидкий доступ до інформації, необхідної для користувачів, і підтримувати актуальність даних у режимі реального часу. Користувачі можуть швидко отримати інформацію про найближчі маршрути, зміни у розкладах та інші аспекти громадського транспорту.

**Джерела:** [EasyWay, Moovit](#_СПИСОК_ВИКОРИСТАНОЇ_ЛІТЕРАТУРИ).

# ПОСТАНОВКА ЗАВДАННЯ

## Мета

Мета курсової роботи полягає в розробці та вивченні бази даних для управління маршрутами громадського транспорту міста з метою оптимізації та покращення функціонування громадської транспортної системи, для комфортного та безперебійного використання пасажирами.

## Основні вимоги

Основні вимоги до розробки бази даних для управління маршрутами громадського транспорту включають ряд технічних та функціональних аспектів. Нижче наведено ключові вимоги, які слід врахувати при проектуванні та розробці системи:

* **Надійність та доступність:**

Система повинна бути зрозумілою та легкою в використанні. Важливо забезпечити, зрозумілий набір функціоналу для максимальної доступності користувачів.

* **Ефективність:**

Система повинна працювати ефективно, забезпечуючи швидкий доступ до даних та оптимальне використання ресурсів. Це особливо важливо при роботі в режимі реального часу, наприклад, для моніторингу руху транспортних засобів.

**Масштабованість:**

Система повинна бути готовою до масштабування для забезпечення нормальної роботи в разі збільшення обсягу даних чи збільшення кількості користувачів. Це може стосуватися як збільшення кількості транспортних засобів, так і пасажиропотоку.

* **Зручний інтерфейс користувача:**

Для адміністраторів та інших користувачів, які мають доступ до системи, важливо розробити зручний та інтуїтивно зрозумілий інтерфейс для роботи з базою даних.

## Завдання

Завдання курсової роботи полягає в вивчені теоретичних основ створення бази даних, вибір програмного забезпечення, збір інформації та опрацювання існуючих моделей, створення структури власної бази даних, заповнення бази даних власними даними, тестування та покращення роботи.

# ПОБУДОВА ER-МОДЕЛІ

## Опис виділених інформаційних об’єктів

Було виділено такі інформаційні об’єкти:

* **Працівники**

Працівники – це люди, які працюють на громадську транспортну інфраструктуру міста.

* **Контракти працівників**

Контракти працівників – це документи, які регулюють трудові відносини між працівником та роботодавцем або компанією.

* **Транспортні засоби**

Транспортні засоби - це машини, які використовуються для перевезення пасажирів громадським транспортом.

* **Типи транспортних засобів**

Типи транспортних засобів - це класифікація транспортних засобів за їхніми характеристиками та призначенням.

* **Технічні обслуговування**

Технічні обслуговування - це роботи, які проводяться для підтримки транспортних засобів у справному стані.

* **Графіки**

Графіки - це таблиці, які визначають час руху транспортних засобів.

* **Маршрути**

Маршрути - це послідовність зупинок, які обслуговує транспортний засіб.

* **Зупинки**

Зупинки - це місця, де транспортні засоби зупиняються для посадки та висадки пасажирів.

* **Типи зупинок**

Типи зупинок - це класифікація зупинок за їхніми характеристиками.

* **Депо**

Депо - це приміщення, де зберігаються та обслуговуються транспортні засоби.

## Опис відношень

Сутність Employees пов’язана один або багато до одного з Employee\_contracts, оскільки працівник може мати декілька контрактів, наприклад після завершення минулого одразу підписав наступний.

Сутність Employee\_contracts пов’язана один до одного або багатьох з Employees, оскільки контракт оформлюється лише для одного працівника.

Сутність Employee\_contracts пов’язана нуль або один до одного або багатьох з Vehicles, оскільки працівник може бути підв’язаним до конкретного транспортного засобу при підписання контракту.

Сутність Vehicles пов’язана один або багато до нуля або одного з Employee\_contracts, оскільки даний транспорті може обслуговувати одна або багато людей, прописаних у контракті.

Сутність Vehicles пов’язана нуль або багато до одного з Technical\_maintenance, оскільки у транспортного засобу може бути багато технічних обслуговувань, а може й зовсім не бути.

Сутність Vehicles пов’язана з Vehicles\_types один до нуля або багатьох, оскільки даний транспортний засіб може відповідати лише одному типу транспорту.

Сутність Vehicles пов’язана нуль або багато до одного з Route\_vehicles, оскільки це є проміжна таблиця, щоб уникнути зв’язок багато до багатьох.

Сутність Vehicles пов’язана один до нуля або багатьох з Depots, оскільки транспортний засіб має лише одне депо до якого він підв’язаний.

Сутність Depots пов’язана нуль або багато до одного з Vehicles, оскільки депо може мати як багато так і нуль транспортних засобів.

Сутність Vehicles\_types пов’язана нуль або багато до одного з Vehicles, оскільки даний тип транспортного засобу може мати декілька транспортних засобів, що підходять конкретному виду, а може й не мати зовсім.

Сутність Technical\_maintenance пов’язана один до нуля або багатьох з Vehicles, оскільки конкретне технічне обслуговування відбувається для конкретного транспортного засобу.

Сутність Routes пов’язана нуль або багато до одного з Route\_vehicles, оскільки це є проміжна таблиця, щоб уникнути зв’язок багато до багатьох.

Сутність Schedules пов’язана нуль або багато до одного з Route\_vehicles, оскільки маршрут один і той самий графік може мати декілька маршрутів, а може й не мати зовсім.

Сутність Routes пов’язана один до нуля або багатьох з Stops, оскільки маршрут може мати лише одну початкову та кінцеву зупинку, також сутність поєднана з Stops через окрему таблицю для уникнення зв’язку багато до багатьох, оскільки маршрут може мати декілька зупинок, і зупинка може мати декілька маршрутів, що проходять крізь неї.

Сутність Stops пов’язана один до один або багатьох з Stop\_types, оскільки зупинка може мати лише один тип.

Сутність Stops\_types пов’язана один або багато до одного з Stops, оскільки конкретний тип зупинки може мати як одна зупинка, так і багато зупинок.

## Опис користувачів

Система повинна бути зручною та достовірною у подачі даних, отже

виділимо наступні задачі та користувачів:

Адміністратор - повинен мати усі права на редагування та перегляд усіх

наявних таблиць.

Водій – повинен мати доступ переглядати таблиці про транспортні засоби, маршрути, зупинки та графіки, типи зупинок, типи транспортних засобів, працівників та їх контракти для отримання загальної інформації.

Інспектор – повинен мати доступ переглядати типи транспортних засобів, їх депо та мати можливість додавати, редагувати, переглядати технічні обслуговування та самі транспортні засоби для перевірки стану транспортних засобів та своєчасних проведень технічних обслуговувань.

Менеджер – повинен мати доступ переглядати транспортні засоби, їх типи та мати можливість додавати, редагувати, переглядати самих працівників та їх контракти для заключення договорів та опціонального присвоєння транспортних засобів працівникам.

## Опис бізнес-правил

Основні бізенс правила:

**Vehicles\_types:**

* + Назва типу транспорту повинна бути одна з - ('Metro', 'Bus', 'Tram', 'Trolleybus', 'Minibus', 'Express Tram')
  + Місткість повинна бути додатною

**Stops\_types:**

* + Назва типу зупинки повинна бути одна з - ('Metro Station', 'Express Tram Stop', 'Tram Stop', 'Public Transit')
  + кількість пасажирів повинна бути додатною
  + площа повинна бути додатною

**Employees:**

* + Дата народження повинна бути більше або 18 років назад
  + Номер телефону та пошта унікальні

**Depots:**

* + Місткість повинна бути додатною
  + Стан депо повинен бути одним з - ('Good', 'Average', 'Poor')

**Routes:**

* + Довжина маршруту повинна бути додатною

**Vehicles:**

* + Дата випуску повинна бути меншою або дорівнювати поточному року
  + Стан транспортного засобу повинен бути одним з - ('Brand New', 'Average', 'Need to be fixed')

**Tehnical\_maintenance:**

* + Тип обслуговування повинен бути одним з - ('Routine', 'Repair', 'Major')
  + Дата початку повинна бути раніше поточної дати
  + Дата завершення повинна бути пізніше дати початку
  + Ціна повинна бути додатною

**Employee\_contracts:**

* + Позиція повинна бути одна з - ('Manager', 'Engineer', 'Technician', 'Driver')
  + Зарплатня повинна бути додатною
  + Дата підписання контракту повинна бути раніше поточного дня
  + Кінцева дата придатності контракту повинна бути пізніше дати підписання
  + Кількість років досвіду повинна бути додатною

**Schedules:**

* + Час початку повинен бути раніше часу закінчення
  + Частота повинна бути додатною
  + Час зупинки повинен бути додатнім та не більше 10 хвилин
  + Дні роботи повинні бути одним з - ('Weekends', 'Week', 'Working days')

**Route\_stops:**

* + Тип зупинки в маршрути повинен бути одним з - ('Starting', 'Mid', 'Ending')

## ER-діаграма

Використовуючи програмне забезпечення Draw.io побудуємо ER-модель(Рис.4.1), що описує подане предметне середовище, виокремивши зв’язки між вищевказаними сутностями:

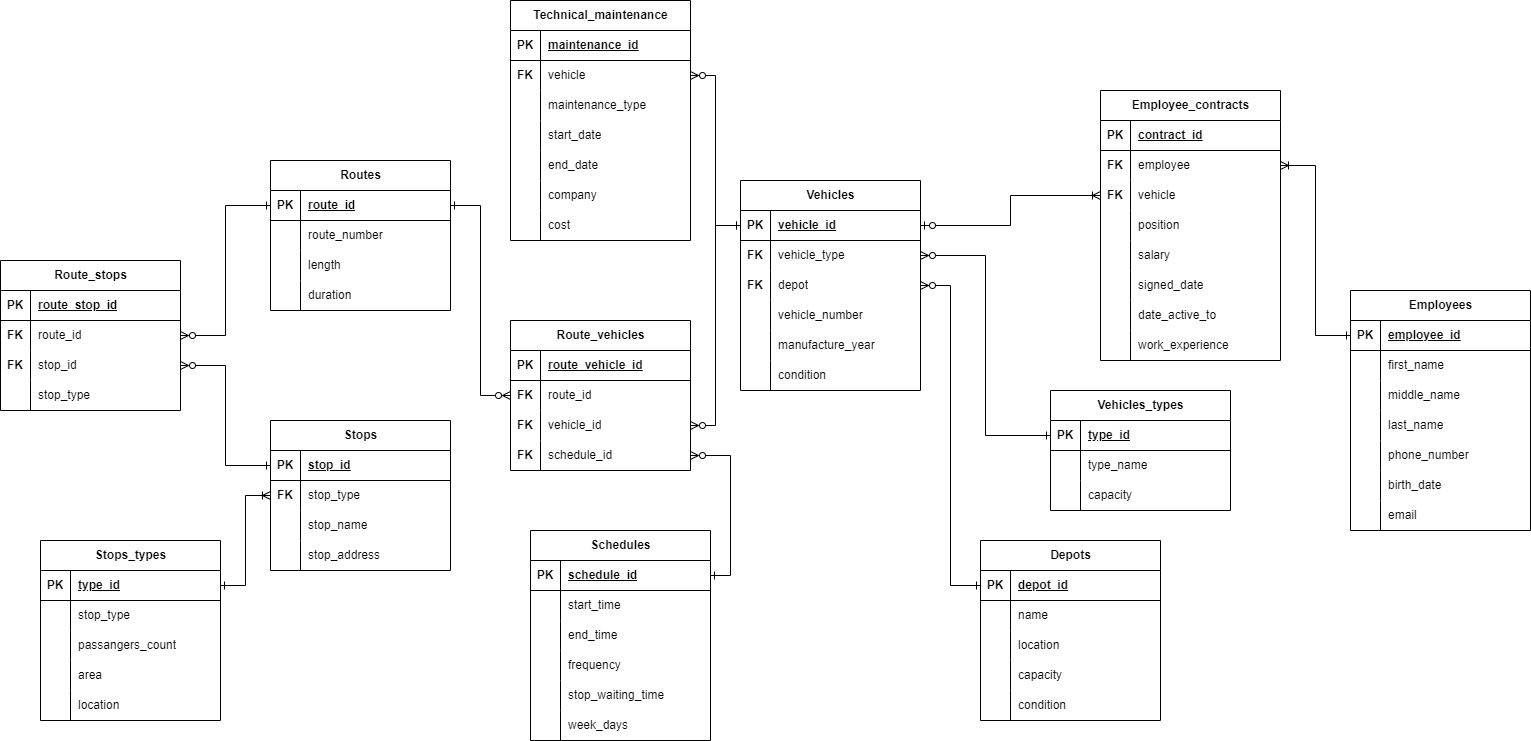


Рисунок 4.1 – Реляційна схема бази даних

# РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

## Обґрунтування вибору СУБД

Для розробки бази даних маршрутів громадського транспорту міста, я використовував СУБД – **MySql**

**MySQL** - це безкоштовна і відкрита система управління базами даних (СУБД) реляційного типу. Вона є однією з найпопулярніших СУБД у світі, що використовується для різних цілей, включаючи управління маршрутами громадського транспорту.

Для розробки бази даних маршрутів громадського транспорту **MySQL** є хорошим вибором з таких причин:

* Функціональність: MySQL має достатню функціональність для задоволення потреб курсової роботи. Наприклад, вона підтримує складні запити, які можуть знадобитися для аналізу даних про маршрути та зупинки.
* Простота використання: **MySQL** є відносно простою в освоєнні та використанні.
* Вартість: **MySQL** є безкоштовною, що є важливою перевагою для студентів.

## Опис структури у табличному вигляді

Таблиця “Employees” призначена для зберігання власних даних про працівників, котрі працюють на транспортну інфраструктуру міста. Структура таблиці наступна:

Таблиця 5.1 – Employees

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| employee\_id | int |  | PK |  | id робітника |
| first\_name | varhcar | 30 |  |  | ім’я робітника |
| middle\_name | varchar | 30 |  |  | по батькові |
| last\_name | varchar | 30 |  |  | прізвище |
| birth\_date | date |  |  |  | дата народження |

Продовження таблиці 5.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| phone\_number | varchar | 20 |  |  | номер телефону |
| email | varchar | 30 |  |  | електронна пошта |

Таблиця “ Employee\_contracts” призначена для зберігання інформації про усі існуючі контракти працівників, котрі працюють на транспортну інфраструктуру міста. Структура таблиці наступна:

Таблиця 5.2 – Employee contracts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| contract\_id | int |  | PK |  | id контракту |
| employee | int |  | FK |  | id робітника |
| vehicle | int |  | FK |  | id транспортного засобу |
| position | varchar | 30 |  | Manager, Engineer, Technician, Driver | посада |
| salary | decimal | 10, 2 |  |  | заробітна плата |
| signed\_date | date |  |  |  | дата підписання |
| date\_active\_to | date |  |  |  | кінцева дата валідності |
| work\_experience | tinyint unsigned |  |  |  | досвід роботи |

Таблиця “ Vehicles” призначена для зберігання інформації про усі громадські транспортні засоби міста. Структура таблиці наступна:

Таблиця 5.3 – Vehicles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| vehicle\_id | int |  | PK |  | id транспортного засобу |
| vehicle\_type | int |  | FK |  | id типу транспортного засобу |
| depot | int |  | FK |  | депо, в якому базується транспортний засіб |
| vehicle\_number | int |  |  |  | номер транспортного засобу |
| manufacture\_year | year |  |  |  | дата випуску |
| vehicle\_condition | varhcar | 17 |  | Brand New, Average, Need to be fixed | поточний стан транспортного засобу |

Таблиця “ Vehicle\_types” призначена для зберігання інформації про усі існуючі типи громадських транспортних засобів міста. Структура таблиці наступна:

Таблиця 5.4 – Vehicle types

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| type\_id | int |  | PK |  | id типу транспортного засобу |

Продовження таблиці 5.4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| type\_name | varchar | 50 |  | Metro, Bus, Tram, Trolleybus, Minibus, Express Tram | назва типу |
| capacity | int |  |  |  | місткість типу транспортного засобу |

Таблиця “ Tehnical\_maintenance” призначена для зберігання інформації про історію технічних обслуговувань громадських транспортів міста. Структура таблиці наступна:

Таблиця 5.5 – Tehnical maintenance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| maintenance\_id | int |  | PK |  | id технічного обслуговування |
| vehicle | int |  | FK |  | id транспортного засобу |
| maintenance\_type | varchar | 30 |  | Routine, Repair, Major | тип технічного обслуговування |
| start\_dtae | date |  |  |  | дата початку |
| end\_date | date |  |  |  | кінцева дата |
| company | varchar | 40 |  |  | назва компанії |
| cost | decimal | 10, 2 |  |  | ціна |

Таблиця “ Schedules” призначена для зберігання інформації, про графіки громадських транспортних засобів міста. Структура таблиці наступна:

Таблиця 5.6 – Schedules

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| schedule\_id | int |  | PK |  | id графіку |
| start\_time | time |  |  |  | початок роботи |
| end\_time | time |  |  |  | кінець роботи |
| frequency | time |  |  |  | частота роботи |
| stop\_waiting\_time | time |  |  |  | час виділений на зупинку |
| week\_days | varhcar | 20 |  | Weekends, Week, Working days | дні роботи графіку |

Таблиця “ Routes” призначена для зберігання інформації про існуючі маршрути громадських транспортів міста. Структура таблиці наступна:

Таблиця 5.7 – Routes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| route\_id | int |  | PK |  | id маршруту |
| route\_number | int |  |  |  | номер маршруту |
| length | int |  |  |  | довжина маршруту |
| duration | time |  |  |  | час проходження маршруту |

Таблиця “ Stops” призначена для зберігання інформації про існуючи зупинки громадського транспорту міста. Структура таблиці наступна:

Таблиця 5.8 – Stops

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| stop\_id | int |  | PK |  | id зупинки |
| stop\_type | int |  | FK |  | id типу зупинки |
| stop\_name | varchar | 30 |  |  | назва зупинки |
| stop\_address | varhcar | 30 |  |  | адреса зупинки |

Таблиця “ Stop\_types” призначена для зберігання інформації про усі існуючі типи зупинок громадського транспорту міста. Структура таблиці наступна:

Таблиця 5.9 – Stop types

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| type\_id | int |  | type\_id |  | id типу зупинки |
| type\_name | varchar | 50 | type\_name | Metro Station, Express Tram Stop, Tram stop, Public Transit | назва типу зупинки |
| passanger\_count | int |  | passanger\_count |  | максимальна кількість пасажирів |
| area | int |  | area |  | площа зупикни |
| location | varhcar | 50 | location |  | локація зупикни |

Таблиця “ Depots” призначена для зберігання інформації про усі депо, що обслуговують громадські транспорти міста. Структура таблиці наступна:

Таблиця 5.10 – Depots

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| depot\_id | int |  | PK |  | id депо |
| depot\_name | varchar | 50 |  |  | назва депо |
| location | varchar | 40 |  |  | адреса депо |
| capacity | int |  |  |  | максимальна кількість транспортних засобів в депо |
| depot\_condition | varhcar | 30 |  | Good, Average, Poor | поточний стан депо |

Таблиця “ Route\_stops” призначена для запобігання зв’язку багато до багатьох між таблицями “Routes” та “Stops”. Структура таблиці наступна:

Таблиця 5.11 – Route stops

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| route\_stop\_id | int |  | PK |  | id маршруту - зупикни |
| route\_id | int |  | FK |  | id маршруту |
| stop\_id | int |  | FK |  | id зупинки |
| stop\_type | varhcar | 30 |  | Starting, Mid, Ending | тип зупинки |

Таблиця “ Route\_vehicles” призначена для запобігання зв’язку багато до багатьох між таблицями “Routes” та “Vehicles”. Структура таблиці наступна:

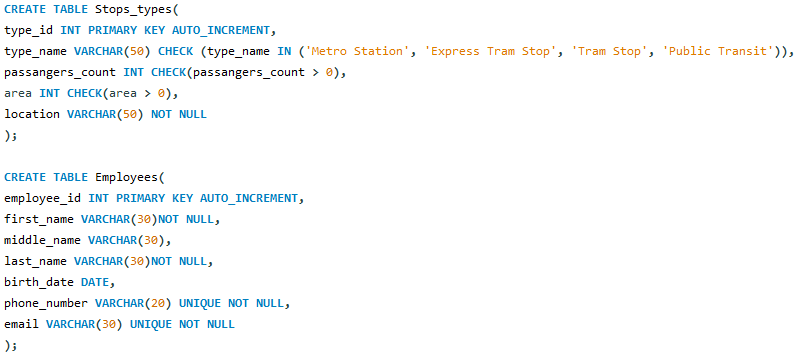
Таблиця 5.12 – Route vehicles

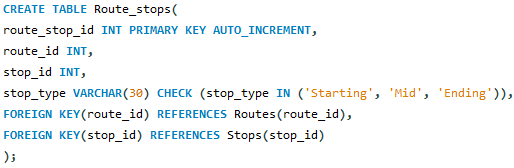
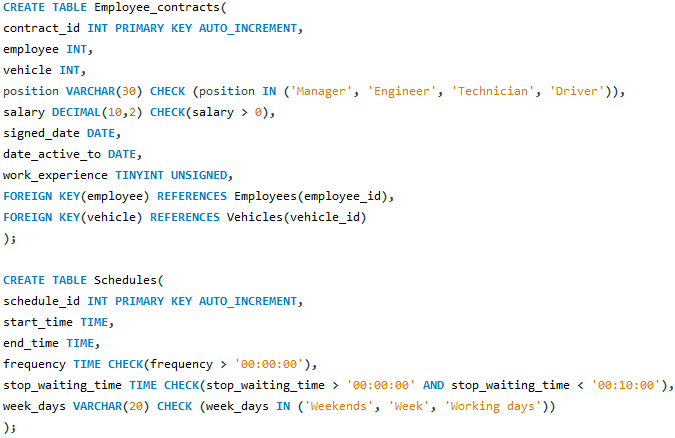
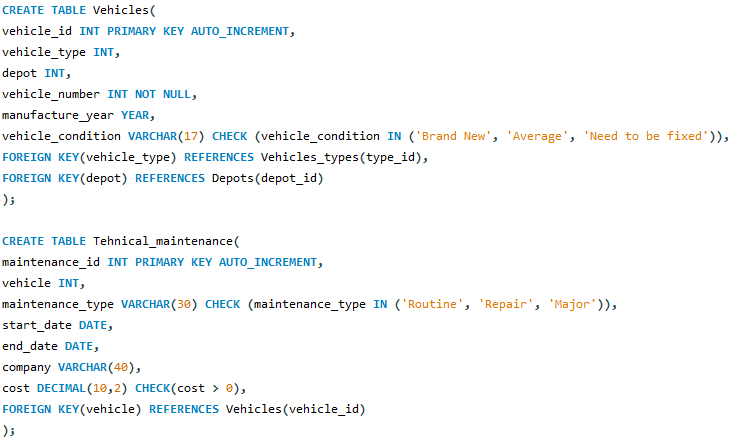
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Ім’я поля*** | ***Тип даних*** | ***Розмір*** | ***Ключ*** | ***Можливі значення*** | ***Опис*** |
| route\_vehicle\_id | int |  | PK |  | id маршруту – транспортного засобу |
| route\_id | int |  | FK |  | id маршруту |
| vehicle\_id | int |  | FK |  | idтранспортного засобу |
| schedule\_id | int |  | FK |  | id графіку |

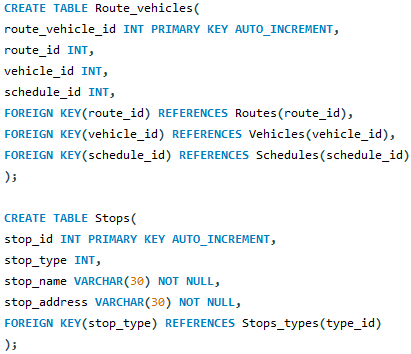
## SQL скрипт

Створення базу даних раніше визначеної структури, з врахованням бізнес правила для запобіганні некоректності данних:





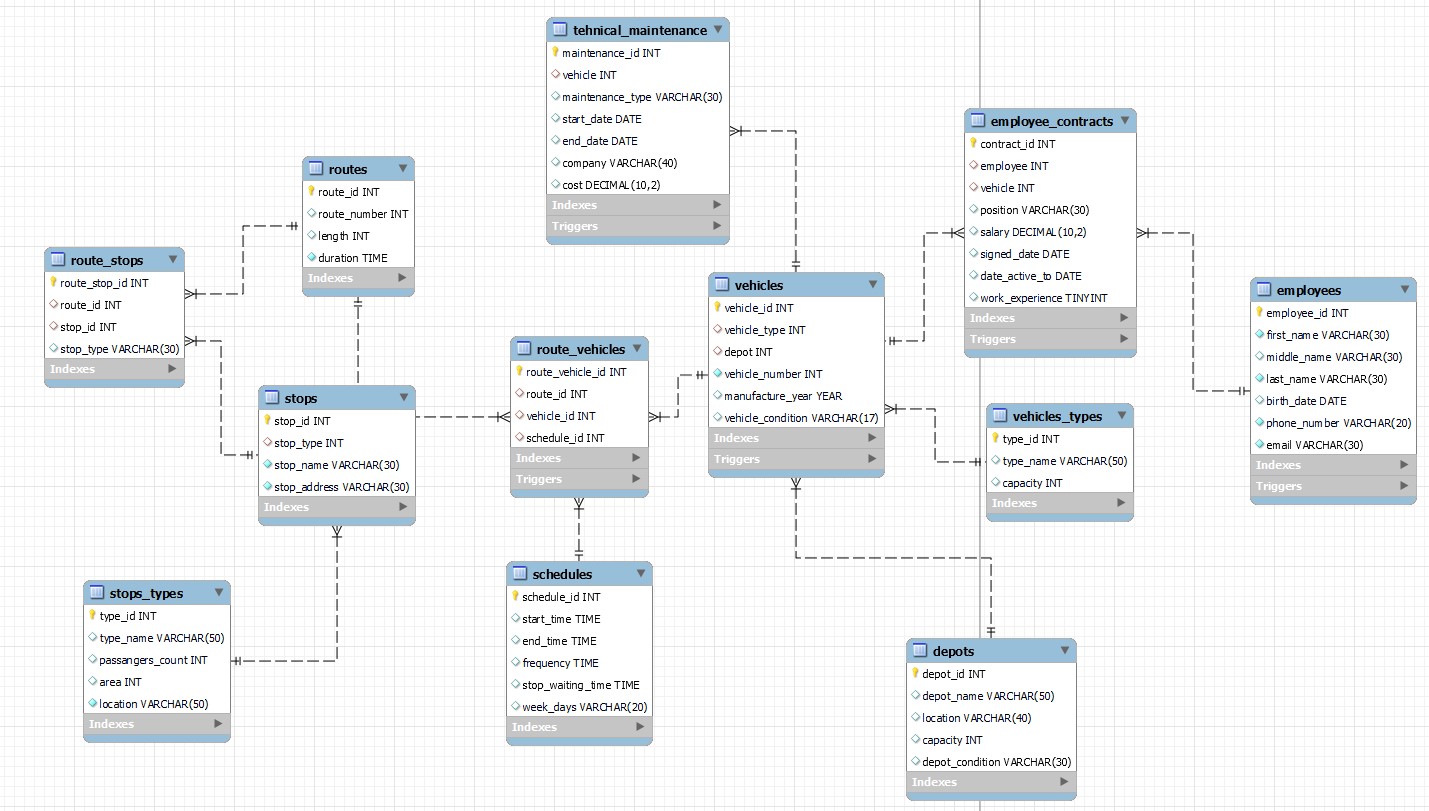




## Схема бази даних

На основі попередньо створеної ER моделі, створимо логічну модель

бази даних, вона ж реляційна база даних:

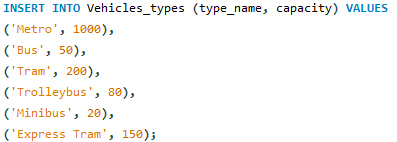


# РОБОТА З БАЗОЮ ДАНИХ

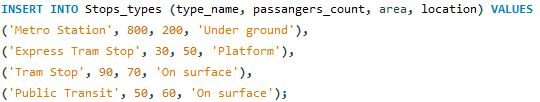
## Тексти вставки даних

Приклади вставки даних у таблиці засобами СУБД:

Таблция Vehicles\_types:



Таблиця Stops\_types:



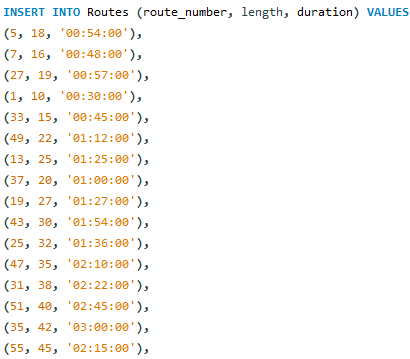
Таблиця Employees:



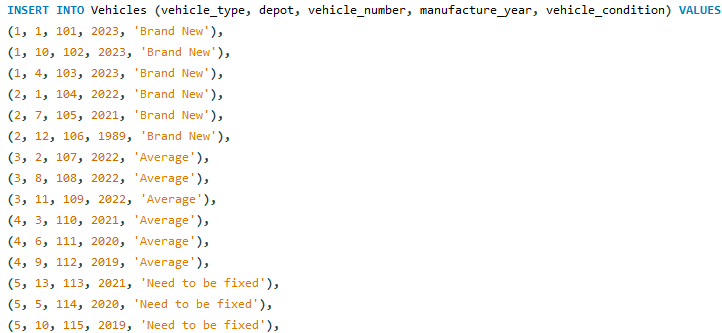
Таблиця Depots:



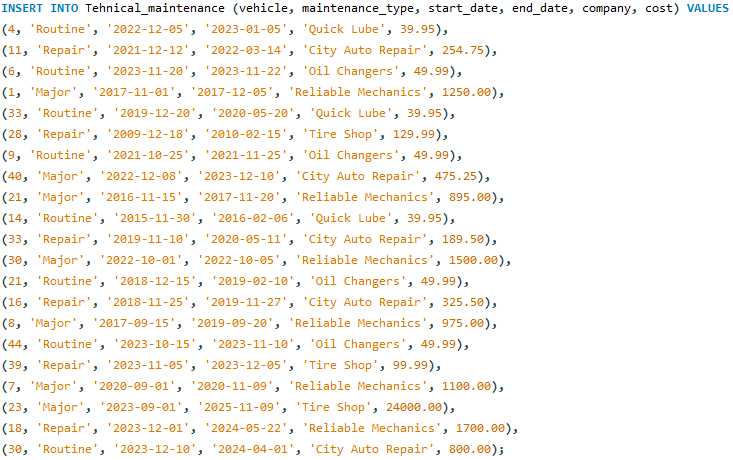
Таблиця Routes:



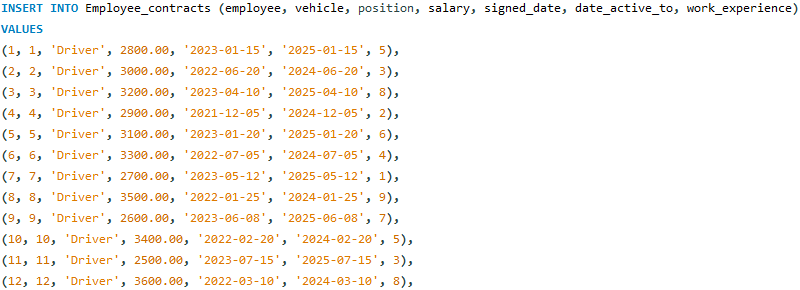
Таблиця Vehicles:



Таблиця Tehnical\_maintenance:



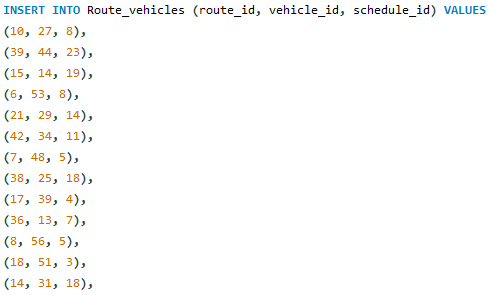
Таблиця Employee\_contracts:



Таблиця Schedules:



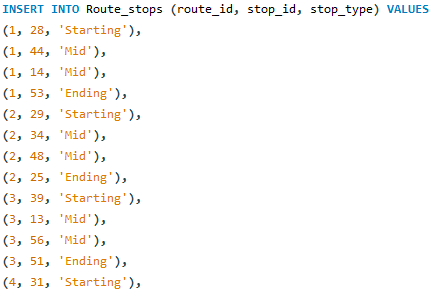
Таблиця Route\_vehicles:



Таблиця Stops:



Таблиця Route\_stops:



## Тексти збережених процедур/функцій

**Процедура GetEmployeeContractsByPosition:**

Для ефективної та зручної роботи менеджера була розроблена процедура, яка дозволяє отримати повну інформацію про контракт та власну інформацію працівника за посадою.

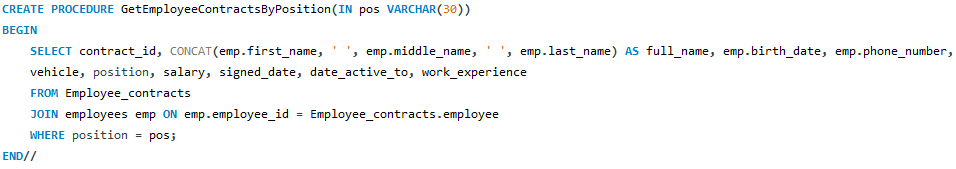
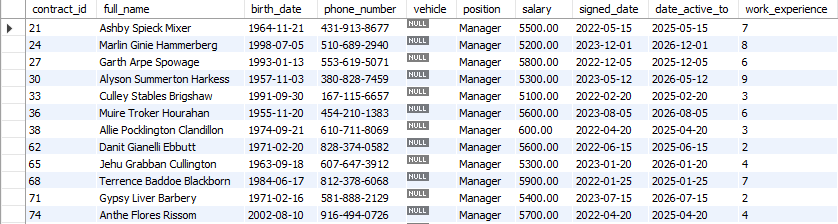


Рисунок 6.1 – Створення процедури **GetEmployeeContractsByPosition**





Рисунки 6.2 та 6.3 – Приклади виконання процедури

**Процедура GetVehiclesInMaintenance**

Інспектору необхідно слідкувати за станами транспортних засобів та відправляти конкретні транспортні засоби на технічні обслуговування та слідкувати за ними, саме тому була розроблена процедура, яка дозволяє переглянути інформацію про транспортні засоби, що зараз перебувають на технічному обслуговуванні.

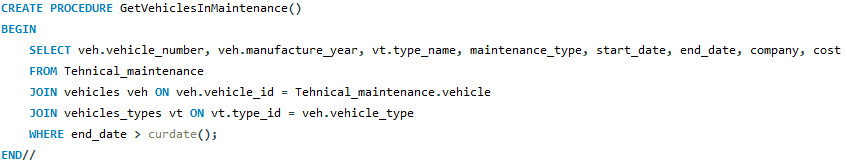
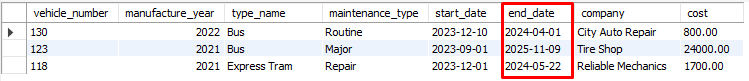


Рисунок 6.4 – Створення процедури **GetVehiclesInMaintenance**





Рисунки 6.5 та 6.6 – Приклади виконання процедури

**Функція CalculateRouteMaxPassangerAmount:**

Водію важливо знати максимальну кількість пасажирів, які можуть його чекати на конкретному маршруті, також це важливо знати й адміністратору при плануванні, створенні маршрутів і розподілені транспортних засобів на них, саме тому була розроблена функція, що дозволяє переглянути максимальну кількість пасажирів для заданого маршруту.

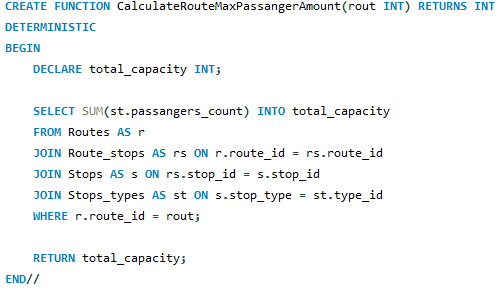


Рисунок 6.7 – Створення функції **CalculateRouteMaxPassangerAmount**



Рисунки 6.8 та 6.9 – Приклад виконання функції

**Процедура GetRoutesByStop:**

Водію та Адміністратору важливо знати, які маршрути проходять через задану зупинки, для планування свого маршруту та завантаженості зупинки. Саме тому була розроблена процедура, що дозволяє переглядати маршрути, що проходять через задану зупинку.

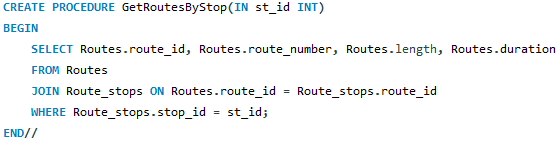
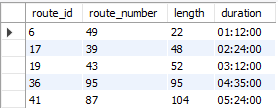


Рисунок 6.10 – Створення процедури **GetRoutesByStop**





Рисунки 6.11 та 6.12 – Приклади виконання процедури

**Процедура GetVehicleMaintenanceHistory:**

Інспектору важливо мати можливість переглядати історію технічних обслуговувань конкретного транспортного засобу, для знаходження проблемних транспортних засобів та загалом для перевірки проходжень. Саме тому була розроблена процедура, що дозволяє переглядати історію технічних обслуговувань для заданого засобу.

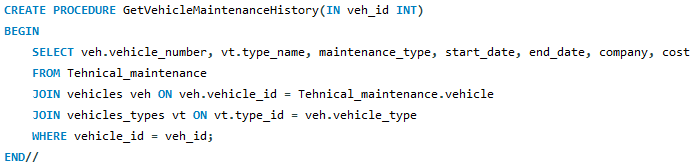


Рисунок 6.13 – Створення процедури **GetVehicleMaintenanceHistory**



Рисунки 6.14 та 6.15 – Приклади виконання процедури

**Процедура GenerateMonthlyMaintenanceReport:**

Інспектор повинен мати можливість генерувати звіт технічних обслуговувань конкретного місяця, для перегляду та вивчення його та передачі його керівництву. Саме тому була розроблена процедура, що дає інформацію про усі технічні обслуговування в конкретному році та місяці

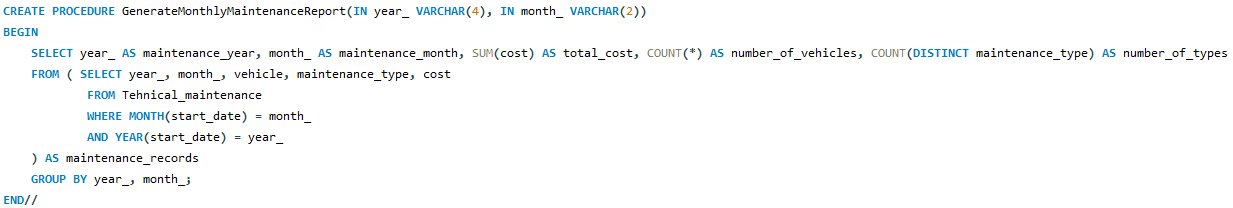


Рисунок 6.16 – Створення процедури **Generate MonthlyMaintenanceReport**





Рисунки 6.17 та 6.18 – Приклади виконання процедури

**Процедура GetEmployeeContracts:**

Менеджер повинен мати можливість отримувати інформацію про усі контракти конкретних працівників, саме тому була розроблена процедура, що дає повну інформацію про заданого робітника та його контрактів.

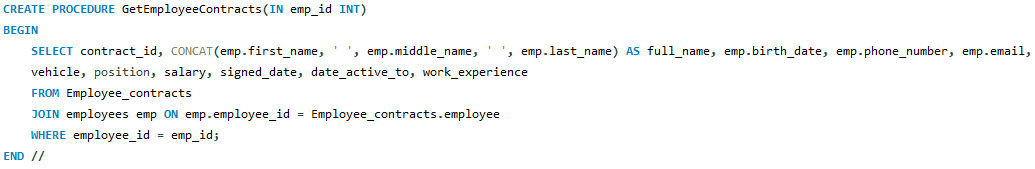


Рисунок 6.19 – Створення процедури **GetEmployeeContracts**





Рисунки 6.20 та 6.21 – Приклади виконання процедури

**Функція GetDepotCapacity:**

Інспектор, Водій та Адміністратор повинні мати інформацію про кількість вільного місця в депо. Саме тому була розроблена функція, яка повертає кількість вільних місць.

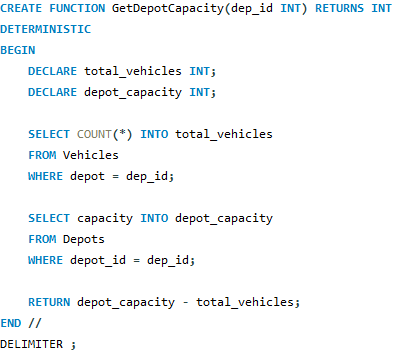


Рисунок 6.22 – Створення функції **GetDepotCapacity**





Рисунки 6.23 та 6.24 – Приклади виконання функції

**Процедура UpdateEmployeeContractInformation:**

Менеджер повинен мати можливість змінювати інформацію про контракт працівника, саме тому була розроблена процедура, що приймає на вхід нові дані та змінює їх.

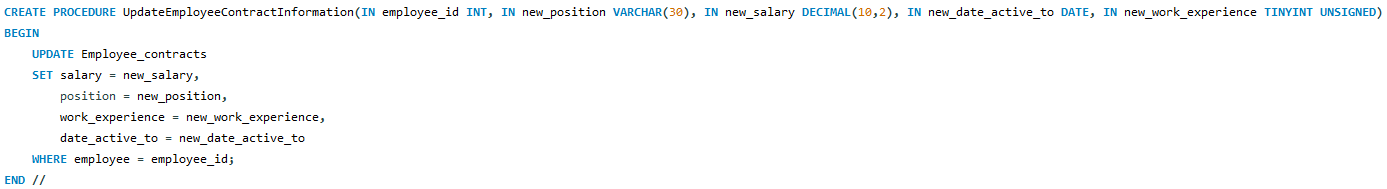


Рисунок 6.25 – Створення процедури **UpdateEmployeeContractInformation**



Рисунок 6.26 – Значення контракту до зміни





Рисунки 6.27 та 6.28 – Приклади виконання процедури

**Процедура UpdateEmployeeContactInformation:**

Менеджер повинен мати можливість змінювати контактну інформацію про працівника, що змінив номер телефону або пошту, саме тому була розроблена процедура, що приймає на вхід нові дані та змінює їх.

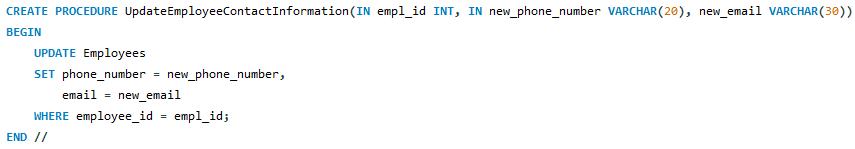


Рисунок 6.29– Створення процедури **UpdateEmployeeContactInformation**



Рисунок 6.30 – Значення персональних даних до зміни





Рисунки 6.31 та 6.32 – Приклади виконання процедури

**Процедура AdjustScheduleTimeForward:**

Ніхто не виключає появу непередбачуваних ситуацій, тому Адміністратор повинен мати можливість переносити графік на час вперед, саме тому була розроблена процедура, що приймає кількість годин та хвилин, і змінює вказаний графік, додаючи вказаний час

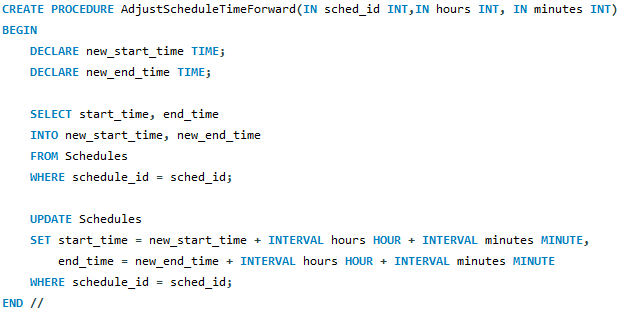


Рисунок 6.33 – Створення процедури **AdjustScheduleTimeForward**



Рисунок 6.34 – До зміни часу графіку





Рисунки 6.35 та 6.36 – Приклади виконання процедури

**Процедура AdjustScheduleTimeBackward:**

Ніхто не виключає появу непередбачуваних ситуацій, тому Адміністратор повинен мати можливість переносити графік на час назад, саме тому була розроблена процедура, що приймає кількість годин та хвилин, і змінює вказаний графік, віднімаючи вказаний час

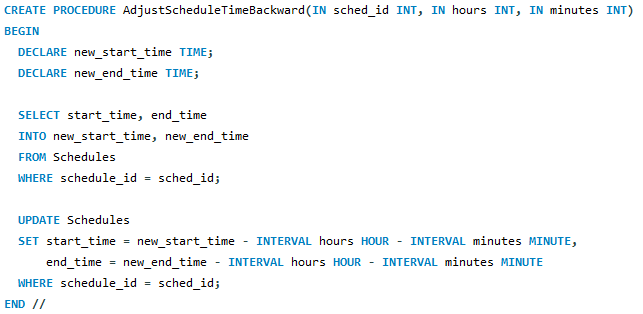


Рисунок 6.37 – Створення процедури **AdjustScheduleTimeBackward**



Рисунок 6.38 – До зміни часу графіку





Рисунки 6.39 та 6.40 – Приклади виконання процедури

## Тексти тригерів

**Тригер birth\_date\_restriction:**

При створенні контракту та додані його, Менеджеру слід перевіряти чи дійсно працівнику більше 18 років, саме тому був розроблений тригер, котрий унеможливлює додання робітника, вік якої менше 18 років

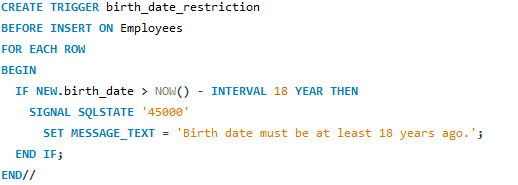


Рисунок 6.41 – Створення тригеру **birth\_date\_restriction**





Рисунки 6.42 та 6.43 – Приклади роботи з не валдіними даними





Рисунки 6.44 та 6.45 – Приклади роботи з валдіними даними

**Тригер maintenance\_date\_restriction:**

При додані технічного обслуговування Інспектором, слід перевіряти валідність дати початку та кінця обслуговування, саме тому був розроблений тригер, який унеможливлює додання технічного обслуговування з датою початку більшою за поточну та кінцеву дату раніше початкової.

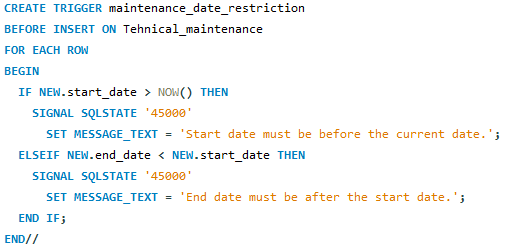


Рисунок 6.46 – Створення тригеру **maintenance\_date\_restriction**





Рисунки 6.47 та 6.48 – Приклади роботи з не валдіною початковою датою





Рисунки 6.49 та 6.50 – Приклади роботи з не валдіною кінцевю датою

**Тригер manufacture\_year\_restriction:**

При доданні транспортного засобу Адміністратором, слід перевіряти чи правильно вказана дата виробу транспорту, саме тому був розроблений тригер, який унеможливлює додання транспортного засобу з роком виробу більше за поточний.

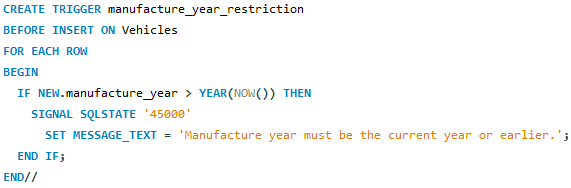


Рисунок 6.51 – Створення тригеру **manufacture\_year\_restriction**





Рисунки 6.52 та 6.53 – Приклади роботи з не валдіною датою створення

**Тригер contract\_dates\_restriction:**

При заповненні контракту Менеджером, слід перевіряти коректність дати підписання та кінцевої дати, тому був розроблений тригер, котрий унеможливлює додання контракту, з датою підписання більшою за поточну та кінцевою датою меншою за дату підписання.

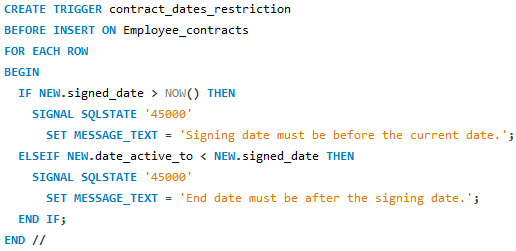


Рисунок 6.54 – Створення тригеру **contract\_dates\_restriction**





Рисунки 6.55 та 6.56 – Приклади роботи з не валдіною датою підписання





Рисунки 6.57 та 6.58 – Приклади роботи з не валдіним кінцевим терміном

**Тригер ensure\_vehicle\_availability:**

При додані Адміністратором конкретного транспортного засобу на маршрут, слід перевіряти чи транспортний засіб в справному стані, саме тому був розроблений тригер, котрий унеможливлює додання транспортного засобу до маршруту, якщо він не в справному стані.

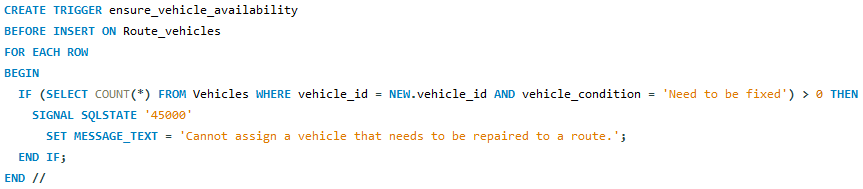


Рисунок 6.59 – Створення тригеру **ensure\_vehicle\_availability**







Рисунки 6.60, 6.61 та 6.62 – Приклади роботи з не несправним станом транспортного засобу

**Тригер update\_vehicle\_condition:**

Після проходження транспортного засобу технічного обслуговування, слід змінити його поточний стан, саме тому був розроблений тригер, який змінює поточний стан транспортного засобу, в залежності від типу обслуговування.

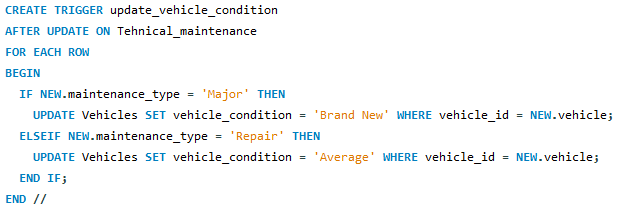
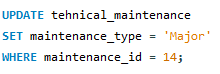


Рисунок 6.63 – Створення тригеру **update\_vehicle\_condition**



Рисунок 6.64 – До зміни





Рисунки 6.65 та 6.66 – Приклади роботи після зміни типу обслуговування

**Тригер salary\_increase\_for\_experience:**

Слід підвищувати заробітну плату працівникам, в яких великий стаж роботи, саме тому був розроблений тригер, який підвищує заробітну плату працівника на 15%, якщо його стаж роботи 5 або більше років.

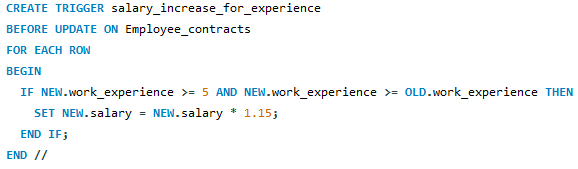


Рисунок 6.67 – Створення тригеру **salary\_increase\_for\_experience**



Рисунок 6.68 – До зміни





Рисунки 6.69 та 6.70 – Приклади роботи після зміни стажу роботи

## Тексти представлень

**Представлення detailed\_route\_info:**

Водію та Адміністратору важливо знати детальну інформацію по маршрутам та зупинкам, саме тому було розроблено представлення для отримання детальної інформації про маршрути.

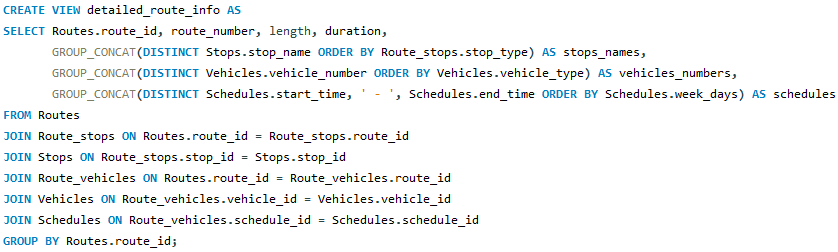
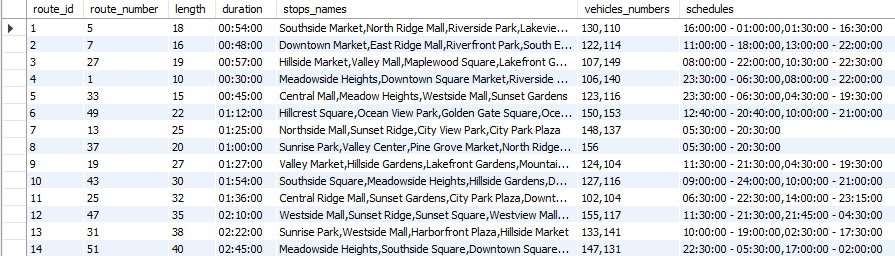


Рисунок 6.71 – Створення представлення **detailed\_route\_info**





Рисунки 6.72 та 6.73 – Приклад роботи представлення

**Представлення contracts\_for\_renewal:**

Менеджеру важливо своєчасно перепідписувати контракти з працівниками, саме тому було розроблено представлення для полегшення цього процесу, за допомогою представлення можна дізнатися інформацію про працівників та їх контракти, що скоро завершуються, а саме в проміжку 30 днів.

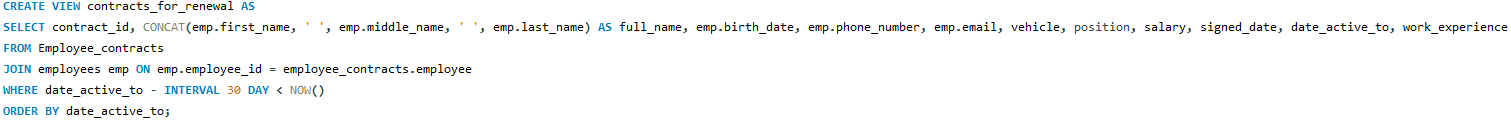
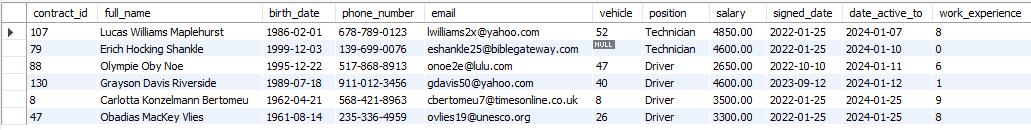


Рисунок 6.74 – Створення представлення **contracts\_for\_renewal**





Рисунки 6.75 та 6.76 – Приклад роботи представлення

**Представлення top\_10\_busiest\_routes:**

Адміністратору важливо покращувати маршрути та перерозподіляти транспортні засоби на найбільш завантажені, саме тому було розроблено представлення для отримання інформації про 10 найбільш завантажених маршрутів.

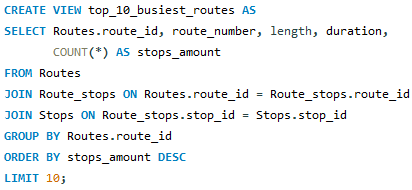
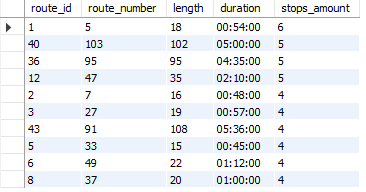


Рисунок 6.77 – Створення представлення **top\_10\_busiest\_routes**





Рисунки 6.78 та 6.79 – Приклад роботи представлення

**Представлення depot\_vehicles:**

Інспектору, Адміністратору важливо знати інформацію про депо та транспортні засоби в них, саме тому було розроблено представлення для отримання детальної інформації про депо та транспортні засоби в них.

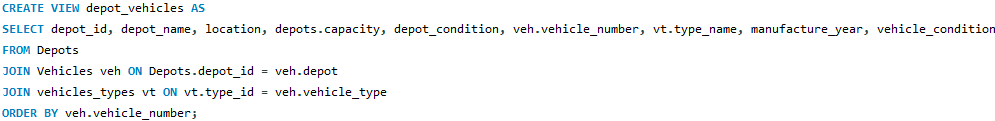
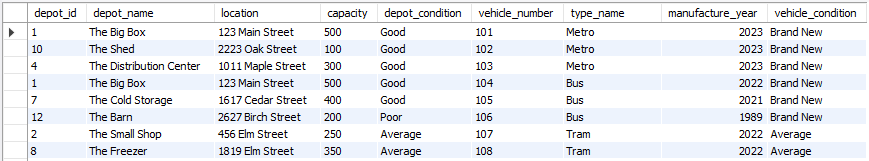


Рисунок 6.80 – Створення представлення **depot\_vehicles**

Рисунки 6.81 та 6.82 – Приклад роботи представлення

**Представлення vehicles\_with\_employees:**

Адміністратору та Менеджеру важливо знати детальну інформацію про транспортні засоби та людей що їх обслуговують, саме тому було розроблене представлення для відображення детальної інформації про транспортні засоби та працівників, що відповідають за них.

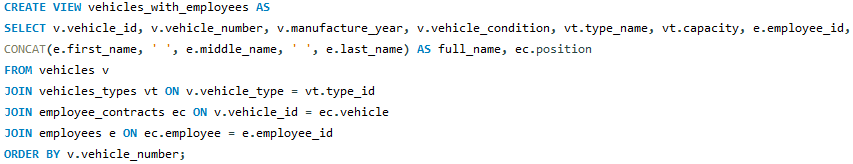
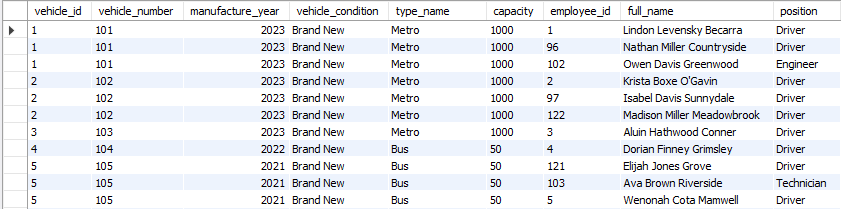


Рисунок 6.83– Створення представлення **vehicles\_with\_employees**





Рисунки 6.84 та 6.85 – Приклад роботи представлення

## SQL-запити

Отримання інформації про працівника та його контракт за шаблоном електронної пошти:

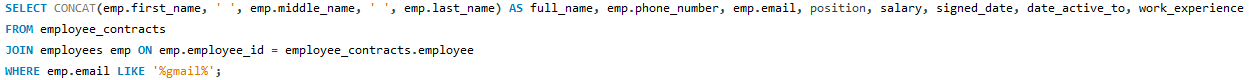


Рисунок 6.86 – Створення запиту

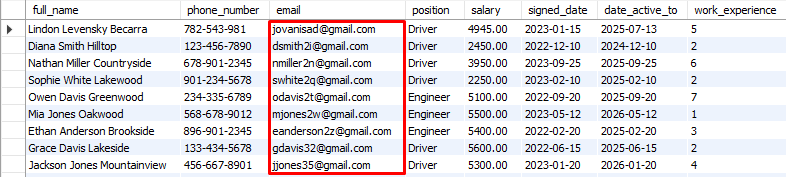


Рисунок 6.87 – Приклад роботи запиту

Отримання інформації про працівника та його транспортний засіб:

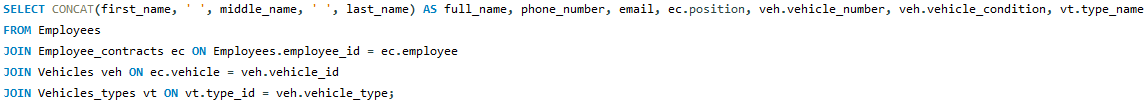


Рисунок 6.88 – Створення запиту

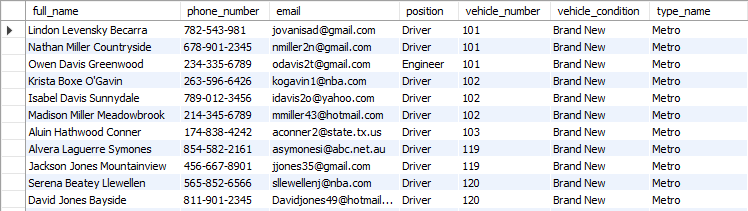
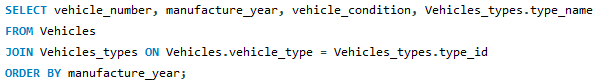


Рисунок 6.89 – Приклад роботи запиту

Отримання інформації про транспортний засіб та його тип, в порядку зростання року випуску:

Рисунок 6.90 – Створення запиту

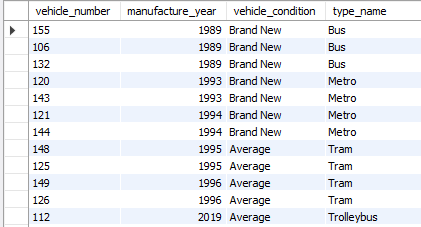


Рисунок 6.91 – Приклад роботи запиту

Отримання інформації транспорт та історію його обслуговувань, де поточний стан транспорту – середній:

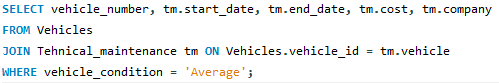


Рисунок 6.92 – Створення запиту

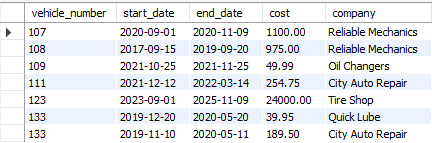


Рисунок 6.93 – Приклад роботи запиту

Отримання кількості грошей витрачених на транспортні засоби що були на технічних обслуговуваннях, впорядковано за кількістю грошей за спаданням:

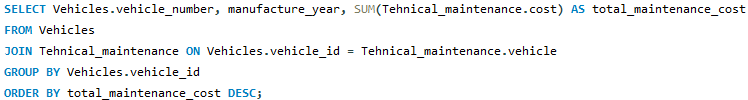


Рисунок 6.94 – Створення запиту

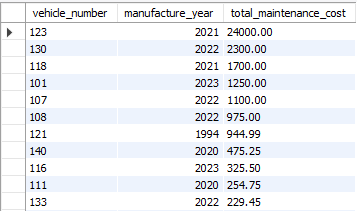


Рисунок 6.95 – Приклад роботи запиту

Отримання інформації про депо та транспортні засоби в ньому, де стан депо – хороший:

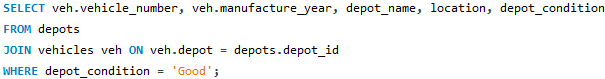


Рисунок 6.96 – Створення запиту

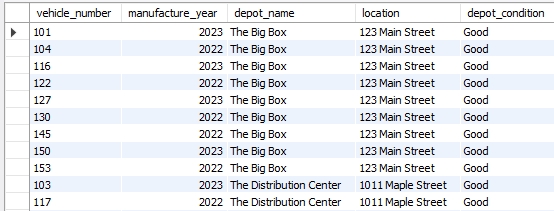


Рисунок 6.97 – Приклад роботи запиту

Отримання кількості транспортних засобів для кожного депо:

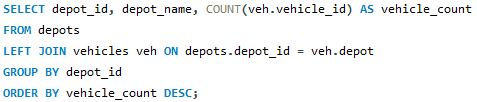


Рисунок 6.98 – Створення запиту

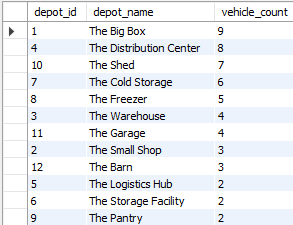


Рисунок 6.99 – Приклад роботи запиту

Отримання повної інформації по депо та транспортним засобам у ньому:

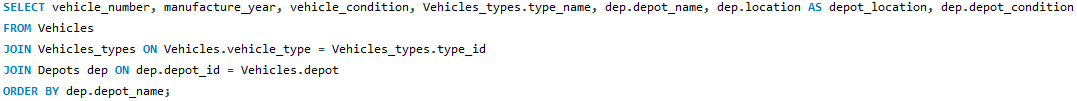


Рисунок 6.100 – Створення запиту

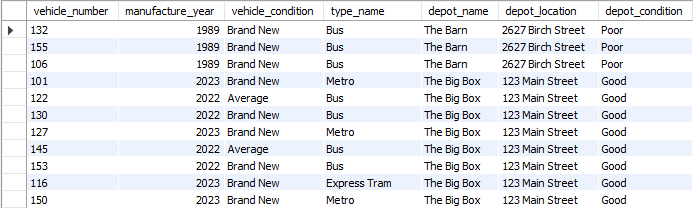


Рисунок 6.101 – Приклад роботи запиту

Отримання інформації про маршрути, котрі обслуговують транспортні засоби вироблені в 2020 році:

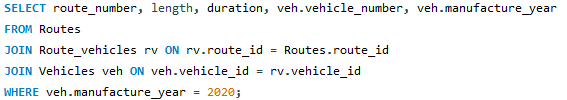


Рисунок 6.102 – Створення запиту

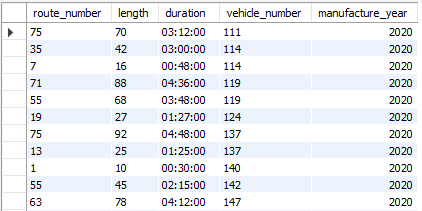


Рисунок 6.102 – Приклад роботи запиту

Отримання повної інформації по маршруту та транспортному засобу, де графік роботи – вихідні:

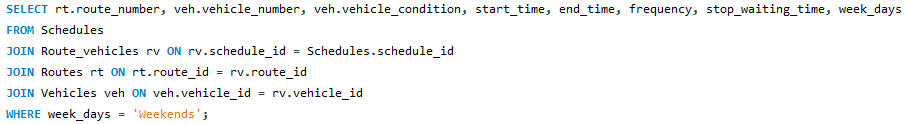


Рисунок 6.103 – Створення запиту

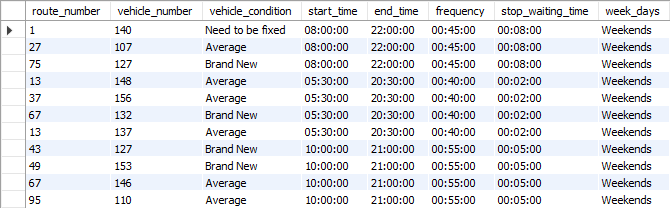


Рисунок 6.104 – Приклад роботи запиту

Отримання інформації про працівників та їх контракти, де зарплата в діапазоні між 2100 та 3000:

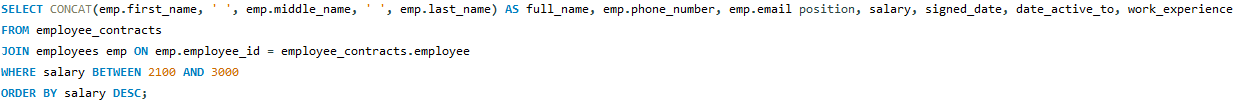


Рисунок 6.105 – Створення запиту

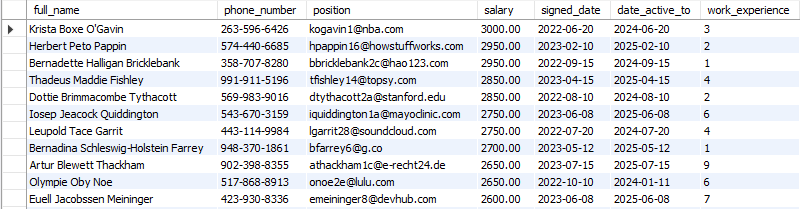


Рисунок 6.106 – Приклад роботи запиту

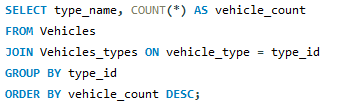
Отримання кількості транспортних засобів кожного типу:  


Рисунок 6.107 – Створення запиту

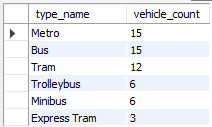


Рисунок 6.108 – Приклад роботи запиту

Отримання інформації про транспортні засоби, котрі не проходили технічні обслуговування:

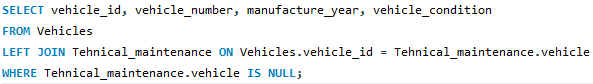


Рисунок 6.109 – Створення запиту

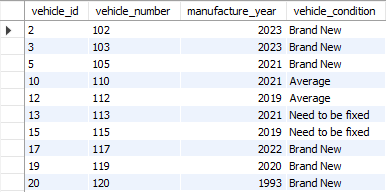


Рисунок 6.110 – Приклад роботи запиту

Отримання початкової та кінцевої зупинки кожного маршруту:

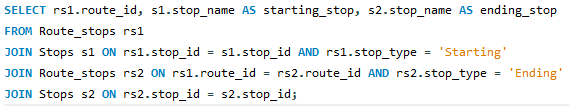


Рисунок 6.111 – Створення запиту

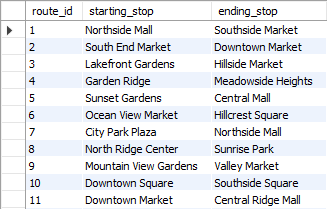


Рисунок 6.112 – Приклад роботи запиту

Отримання повної інформації про робітника та його контракти, в яких заробітна плата більша за середню:

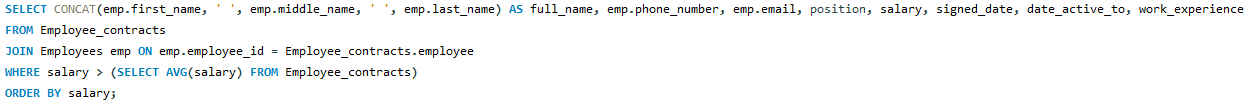


Рисунок 6.113 – Створення запиту

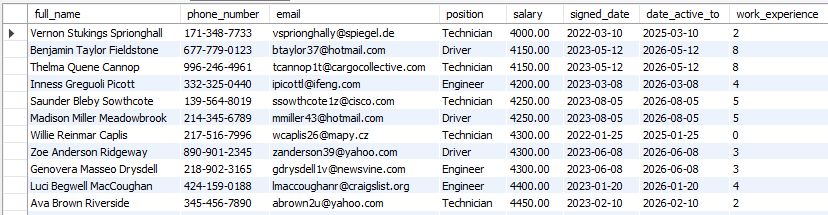


Рисунок 6.114 – Приклад роботи запиту

Отримання номеру транспортного засобу та року випуску, де сума технічного обслуговування більша за середню:

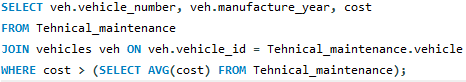


Рисунок 6.115 – Створення запиту



Рисунок 6.116 – Приклад роботи запиту

Отримання кількості маршрутів котрі проходять через кожну зупинку:

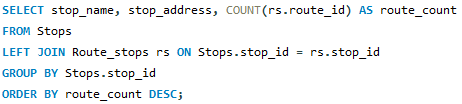


Рисунок 6.117 – Створення запиту

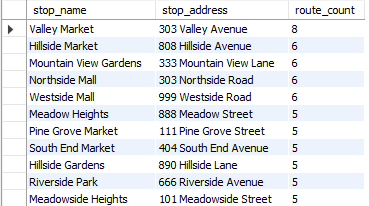


Рисунок 6.118 – Приклад роботи запиту

Отримання маршрутів, час проходу яких більший за середнє значення:

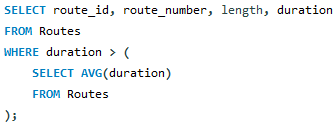


Рисунок 6.119 – Створення запиту

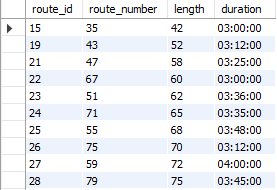


Рисунок 6.120 – Приклад роботи запиту

Отримання маршрутів, котрі проходять через визначену зупинку – 3:

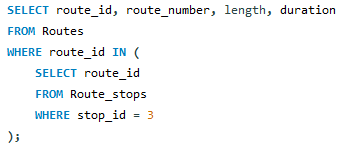


Рисунок 6.121 – Створення запиту



Рисунок 6.122 – Приклад роботи запиту

Отримання найдовшого маршруту за часом проїзду:

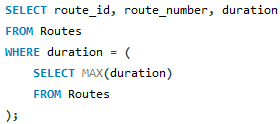


Рисунок 6.123 – Створення запиту



Рисунок 6.124 – Приклад роботи запиту

## Результати оптимізації

Створення індексу для пришвидшення роботи пошуку працівника за прізвищем, прізвищем + іменем та прізвищем + іменем + по батькові:



Рисунок 6.125 – Створення індексу





Рисунки 6.126 та 6.127 – Приклад пошуку за прізвищем до використання індексу



Рисунок 6.128 – Приклад пошуку за прізвищем після додання індексу

Як ми можемо побачити до використання індексу, було переглянуто 131 рядок та відфільтровано 10 відсотків, а після додання індексу переглянуто 1 рядок та відфільтровано 100 відсотків.

Створення індексу для пришвидшення роботи пошуку за позицією та позицією + заробітною платою:



Рисунок 6.129 – Створення індексу





Рисунки 6.130 та 6.131 – Приклад пошуку за заробітною платою та посадою до використання індексу



Рисунок 6.132 – Приклад пошуку за заробітною платою та посадою після додання індексу

Як ми можемо побачити до використання індексу, було переглянуто 131 рядок та відфільтровано 1.11 відсотків, а після додання індексу переглянуто 12 рядків та відфільтровано 100 відсотків.

Створення індексу для пришвидшення роботи пошуку за номером транспортного засобу:



Рисунок 6.133 – Створення індексу





Рисунки 6.134 та 6.135 – Приклад пошуку за номером транспортного засобу до використання індексу

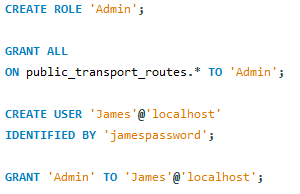


Рисунок 6.136 – Приклад пошуку за номером транспортного засобу після додання індексу

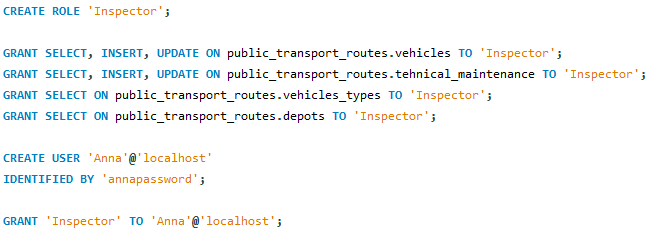
Як ми можемо побачити до використання індексу, було переглянуто 57 рядок та відфільтровано 10 відсотків, а після додання індексу переглянуто 1 рядок та відфільтровано 100 відсотків.

## Створення користувачів

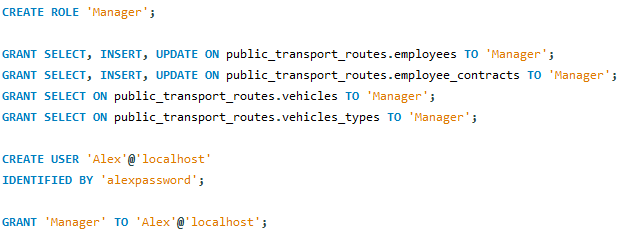
Створення ролі Admin та надання користувачу цієї ролі:



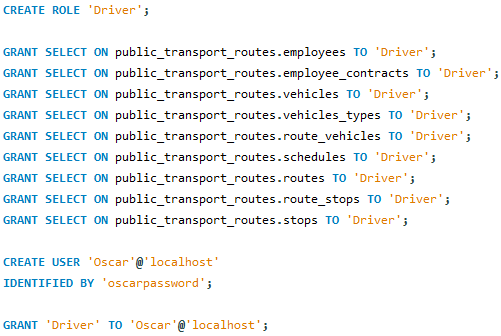
Створення ролі Inspector та надання користувачу цієї ролі:



Створення ролі Manager та надання користувачу цієї ролі:



Створення ролі Driver та надання користувачу цієї ролі:



# ВИСНОВКИ

У рамках даної курсової роботи було проведено комплексний аналіз та розробка бази даних, спрямована на ефективне управління маршрутами громадського транспорту міста. Розглянуті етапи включають в себе не лише опис об'єкту дослідження, а й здійснення аналізу існуючих ринкових продуктів для виявлення найкращих практик та можливостей оптимізації.

Визначені основні бізнес-процеси використання даних стали основою для структурування бази даних, виокремлення інформаційних об'єктів та визначення зв'язків між ними. Детально розглянута структура бази даних, її логічне проектування та реалізація за допомогою SQL-скриптів, а також оптимізація для підвищення продуктивності та швидкості обробки даних.

Основною метою роботи було створення зручного інструменту управління, який відповідає потребам системи громадського транспорту. Аналіз бізнес-процесів використання даних дозволив ідентифікувати ключові етапи оптимізації роботи транспортної системи та підвищення якості обслуговування пасажирів.

Окрім цього, важливою частиною роботи стала обґрунтована вибірка Системи Управління Базами Даних (СУБД) MySQL для реалізації бази даних. Обрана СУБД відповідає високим вимогам до продуктивності, стабільності та гнучкості, що є критичними для успішного функціонування системи управління маршрутами.

У результаті виконаної роботи створено концептуальну основу для ефективного управління громадським транспортом, що сприятиме покращенню якості послуг та задоволенню потреб пасажирів. Дана база даних може слугувати важливим інструментом для прийняття управлінських рішень та оптимізації роботи системи громадського транспорту міста.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

EasyWay. (без дати). *EasyWay Kyiv Routes*. Отримано з EasyWay: https://www.eway.in.ua/ua/cities/kyiv/routes

Mockaroo. (без дати). *Mockaroo data set generator*. Отримано з Mockaroo App: https://www.mockaroo.com/

Moovit. (без дати). *Маршрутне таксі - Графіки руху, маршрути та зупинки*. Отримано з Moovit App: https://moovitapp.com/index/uk/%D0%93%D1%80%D0%BE%D0%BC%D0%B0%D0%B4%D1%81%D1%8C%D0%BA%D0%B8%D0%B9\_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82-Kyiv-3468

MySQL. (без дати). *MySQL Documentation*. Отримано з MySQL dev: https://dev.mysql.com/doc/