

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
(повна назва інституту/факультету)  
КАФЕДРА інформатики та програмної інженерії  
(повна назва кафедри)

**КУРСОВА РОБОТА**

з дисципліни «Бази даних»  
(назва дисципліни)

на тему: база даних служби таксі

Студента 2 курсу ІІ-25 групи  
спеціальності 121 «Інженерія програмного  
забезпечення»

Денисенка В'ячеслава Сергійовича  
(прізвище та ініціали)

Керівник Ліщук Катерина Ігорівна  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_

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 – 2023 рік

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки  
(повна назва)

Кафедра Інформатики та програмної інженерії  
(повна назва)

Дисципліна Бази даних

Курс 2 Група ІІІ-25 Семестр 3

**З А В Д А Н Н Я  
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Денисенку В'ячеславу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи База даних служби таксі

керівник роботи Ліщук Катерина Ігорівна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи 29.12.2023

3. Вихідні дані до роботи для вивчення теми "База даних служби таксі"  
включають інформацію про клієнтів та їх замовлення, дані про транспортні  
засоби та їх технічний стан, а також фінансові аспекти та системи аналітики для  
оптимізації роботи служби. Ефективна інтеграція цих елементів дозволить  
підвищити функціональність та безпеку служби таксі.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1)

Аналіз предметного середовища

2) Побудова ER-моделі

3) Побудова реляційної схеми з ER-моделі

4) Створення бази даних, у форматі обраної системи управління базою даних

5) Створення користувачів бази даних

6) Імпорт даних з використанням засобів СУБД в створену базу даних

7) Створення мовою SQL запитів

8) Оптимізація роботи запитів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання 08.11.2023

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметного середовища	12.11.2023	
2	Побудова ER-моделі	13.11.2023	
3	Побудова реляційної схеми з ER-моделі	20.11.2023	
4	Створення бази даних, у форматі обраної системи управління базою даних	25.11.2023	
5	Створення користувачів бази даних	25.11.2023	
6	Імпорт даних з використанням засобів СУБД в створену базу даних	01.12.2023	
7	Створення мовою SQL запитів	10.12.2023	
8	Оптимізація роботи запитів	15.12.2023	
9	Оформлення пояснювальної записки	25.12.2023	
10	Захист курсової роботи	30.12.2023	

Студент

\_\_\_\_\_  
(підпис)

Денисенко В.С.

(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
(підпис)

Лішук К.І.

(прізвище та ініціали)

## Зміст

<b>ВСТУП .....</b>	<b>4</b>
<b>1. АНАЛІЗ ОБ'ЄКТУ ДОСЛІЖЕННЯ.....</b>	<b>7</b>
1.1.Опис предметного середовища .....	7
1.2.Аналіз існуючих програмних продуктів .....	8
<b>2. ПОСТАНОВКА ЗАВДАННЯ .....</b>	<b>12</b>
<b>3. ПОБУДОВА ER-МОДЕЛІ .....</b>	<b>13</b>
3.1.Опис ER-моделі.....	13
3.2.Зв'язки між сутностями .....	15
3.3.ER-модель.....	17
<b>4. РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАННИХ.....</b>	<b>17</b>
<b>5. РЕАЛІЗАЦІЯ БАЗИ ДАННИХ.....</b>	<b>18</b>
5.1.Обґрунтування вибору СУБД .....	18
5.2.Опис таблиць бази даних .....	19
5.3.Програмний код створених таблиць .....	22
5.4.Схема бази даних, згенерована засобами СУБД .....	24
<b>6. СТВОРЕННЯ КОРИСТУВАЧІВ БАЗИ ДАННИХ .....</b>	<b>24</b>
<b>7. РОБОТА З БАЗОЮ ДАННИХ .....</b>	<b>28</b>
7.1.Тексти генераторів.....	28
7.2.Тексти збережених процедур/функцій .....	28
7.3.Тексти тригерів .....	35
7.4.Тексти представлень .....	38
7.5.SQL-запити та результати виконання.....	41
<b>ВИСНОВКИ .....</b>	<b>48</b>
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....</b>	<b>49</b>

## ВСТУП

Таксі – це міський транспорт, зазвичай автомобіль, який клієнти використовують для подорожі в будь-яку точку. Оплата за проїзд вираховується згідно таксометру.

Історія таксі почалася на початку 17 століття в Лондоні. Тоді почали з'являтися карети, які перевозили пасажирів за гроші. Проте назва пішла від французького слова *hacquerée*, що в перекладі означало коня, якого беруть в оренду. На осі карети був закріплений «таксометр». Це був звичайний таз, в який кожні 200 метрів падав камінчик. В кінці поїздки підраховувалась кількість камінців і за їх кількістю розраховувалась вартість поїздки.

На 1988 рік в колишньому СРСР була прийнята система «такса за дистанцію»: 20 копійок за кілометр, при цьому відстань, яку треба було проїхати обговорювалася ще до початку поїздки. Тут з'являлася проблема, що водієві було не вигідно працювати в час пік, проте в ті час пробок не було. Також були і переваги зі сторони пасажирів тому, що вартість поїздки не залежить від дорожніх обставин, а водій може вибрати для себе оптимальний маршрут та рухатись з оптимальною швидкістю.

Існували також інші варіанти тарифікації: такса за хвилину або за кілометр по лічильнику. Проте такі способи розрахунку вартості поїздки були чреваті шахрайством зі сторони водія.

Послуги сервісу з автоперевезення – таксі в сучасному житті відіграє дуже важливу роль, яка виконує найрізноманітніші функції: від традиційної доставки пасажирів та допомоги їм в порятунку життя людини.

Таксі потрібно кожному з нас. Всі служби таксі знаходяться під пильним наглядом конкурентів. Найважливіше для цієї галузі – конкурентоспроможність, яка в свою чергу потребує найкращого обслуговування клієнтів. Тому компанії с автоперевезення, розуміючи це, покращують свій сервіс не тільки за рахунок технічного стану автопарку, а і висококваліфікованих водіїв.

На сьогоднішній день існують компанії, які можуть запропонувати клієнту додаток в мобільний додаток, через який він може здійснити замовлення. В таких компаніях як правило застосовують дорогі іномарки, які не мають вигляд таксі. Вони застосовуються для клієнтів, які б хотіли показати свій високий соціальний статус. Проте в додатку можна обрати і бюджетний варіант, в якому буде запропоновано машину нижче за класом

Актуальність теми.

Сумісні дії з автоматизації процесів сервісу та покращення рівня обслуговування за рахунок контролю рейтингу водіїв стає актуальним питанням з розвитку компаній, які надають послуги автоперевезень Тому тема дипломного проекту є актуальною.

Практичне значення:

Інформаційна система з підтримки служби таксі забезпечує реєстрацію водія адміністратором з характеристиками своєї машини як виконавця замовлення, а також дає можливість клієнту створити замовлення та обрати вільного водія згідно своїх побажань.

Моніторинг процесу обслуговування клієнтів надає додатковий стимул для водія обслуговувати пасажирів на високому рівні тому, що клієнт буде оцінювати услугу, яка буде надаватися компанією.

## 1. АНАЛІЗ ОБ'ЄКТУ ДОСЛІЖЕННЯ

### 1.1.Опис предметного середовища

За оцінками на 2020 рік, обсяг ринку таксі в Україні склав приблизно 40 мільярдів гривень. Водіїв налічується 220 тисяч осіб, з яких 98% працюють у нелегальному статусі. Щоденно таксисти виконують у середньому 1,5 мільйона поїздок. У Києві Uber займає 50% ринкової частки, Uklon - 15%, і Bolt - 10%. Решта ринку розподіляється між численними дрібними перевізниками.

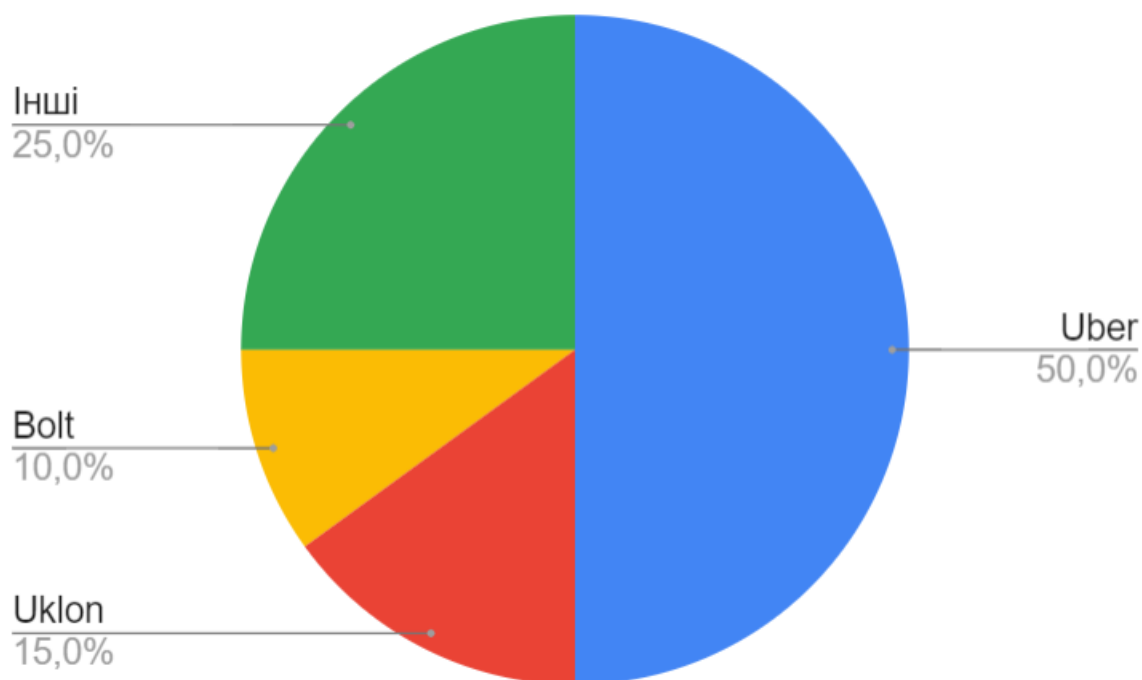


Рисунок 1.1.1 – Кругова діаграма порівняння

Великі служби онлайн-замовлень діють в обмеженій кількості великих міст (наприклад, Uber у семи містах, Uklon - у 13), що призводить до того, що ринок у менших містах переважно контролюється малими операторами. Процес замовлення таксі може відбуватися кількома способами:

- Вуличні таксі: водії рухаються по місту і можуть бути "спіймані" на вулиці або чекати на місцях з інтенсивним потоком людей, таких як аеропорт або вокзал.
- Служби таксі: клієнти можуть замовити таксі по телефону. У таких компаніях є диспетчер, який обробляє замовлення клієнта та розраховує вартість поїздки заздалегідь. Ці компанії, хоча і не мають власних автомобілів, надають інформаційні послуги водіям і отримують за це комісію.

- Інтернет-служби таксі: ці служби дозволяють клієнтам робити замовлення через мобільний додаток або на веб-сайті. Вони використовують інтелектуальні алгоритми машинного навчання для розрахунку ціни поїздки, побудови маршруту на карті для водія та зменшення часу очікування. Ці компанії також не мають власного автопарку.

Попри переваги користування таксі, виникають і певні проблеми.

## 1.2. Аналіз існуючих програмних продуктів

### 1.2.1 Публічна компанія Uber

Uber - американська міжнародна компанія із штаб-квартирою в Сан-Франциско, яка розробила популярний мобільний додаток для замовлення таксі, приватних водіїв та доставки їжі[4]. За допомогою додатка Uber користувач може замовити транспорт і відстежувати його маршрут до зазначеної точки призначення. Більшість водіїв використовують власні автомобілі, а також транспорт таксопарків чи партнерів компанії. У більшості країн 67% вартості поїздки переходить водієві, тоді як 33% залишаються у Uber.

У 2012 році Uber анонсувала плани розширення функцій додатка, що дозволяє водіям шукати попутників, які не надають послуги таксі.

У червні 2015 року Uber запустив новий сервіс у Стамбулі під назвою UberBOAT, який забезпечує транспортування через протоку Босфор. Перевезення здійснюється човнами з вартістю від 17 доларів і вмістом від 7 до 10 осіб.



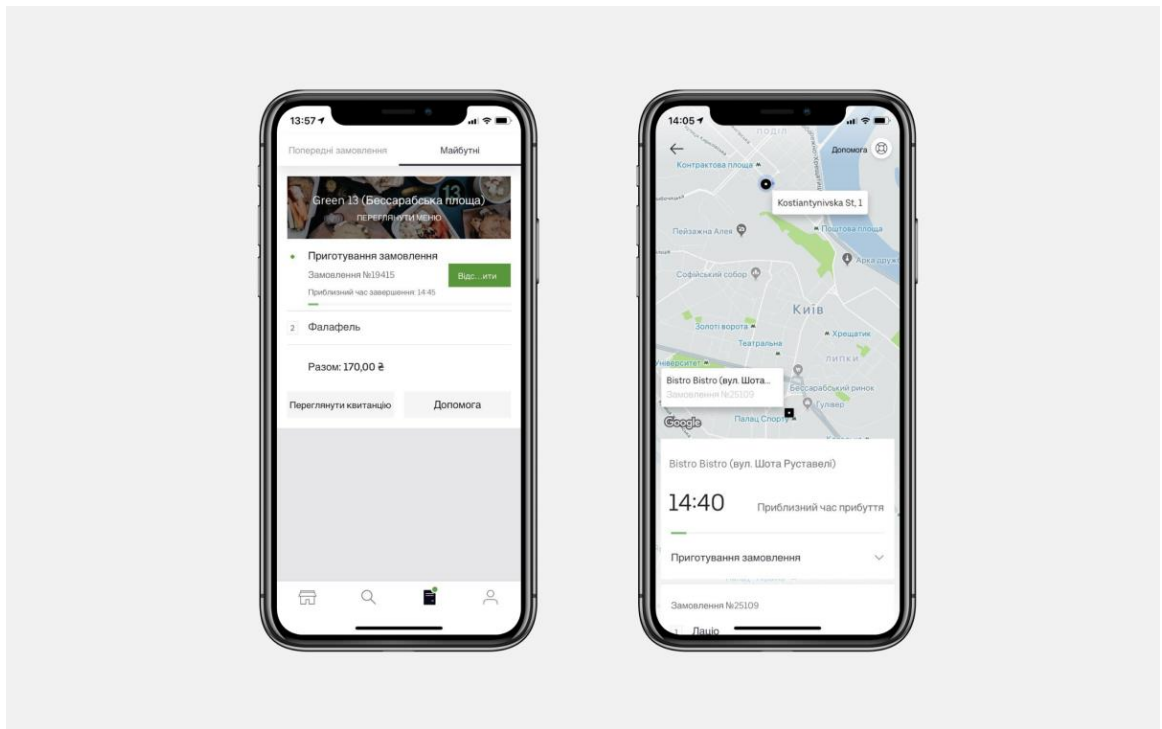


Рисунок 1.2.1.1 – Вигляд інтерфейсу мобільного додатку UBER  
**Переваги:**

- безготівкова оплата через додаток;
- можливість відстежити переміщення автомобіля до вказаної адреси;
- доступність (у багатьох країнах Uber дешевше звичайного таксі).

**Недоліки** (з відгуків клієнтів [5]):

- ненадійні водії, що скасовують замовлення за 1 хв до прибуття;
- служба підтримки відсутній в Україні;
- відсутність належного контролю при підборі співробітників

## 1.2.2 Сервіс Uklon

Uklon - перший сервіс виклику автомобілів через Інтернет на території України. Користувачі можуть замовляти авто за допомогою мобільного додатка на смартфоні або через інтернет-сайт, уникнувши телефонних дзвінків до диспетчерських служб[6]. За деякими оцінками, Uklon вважається однією з провідних платформ в Україні.

Uklon самостійно не є службою таксі; це агрегатор замовлень, який розподіляє виклики між водіями-таксистами. Пасажирські перевезення в основному здійснюються диспетчерськими службами таксі.

На жовтень 2019 року сервіс Uklon активно функціонує в таких містах України: Києві, Одесі, Харкові, Дніпрі, Львові, Запоріжжі, Вінниці, Сумах,

Полтаві, Тернополі, Чернігові, Кривому Розі, Миколаєві, Білій Церкві та Маріуполі.

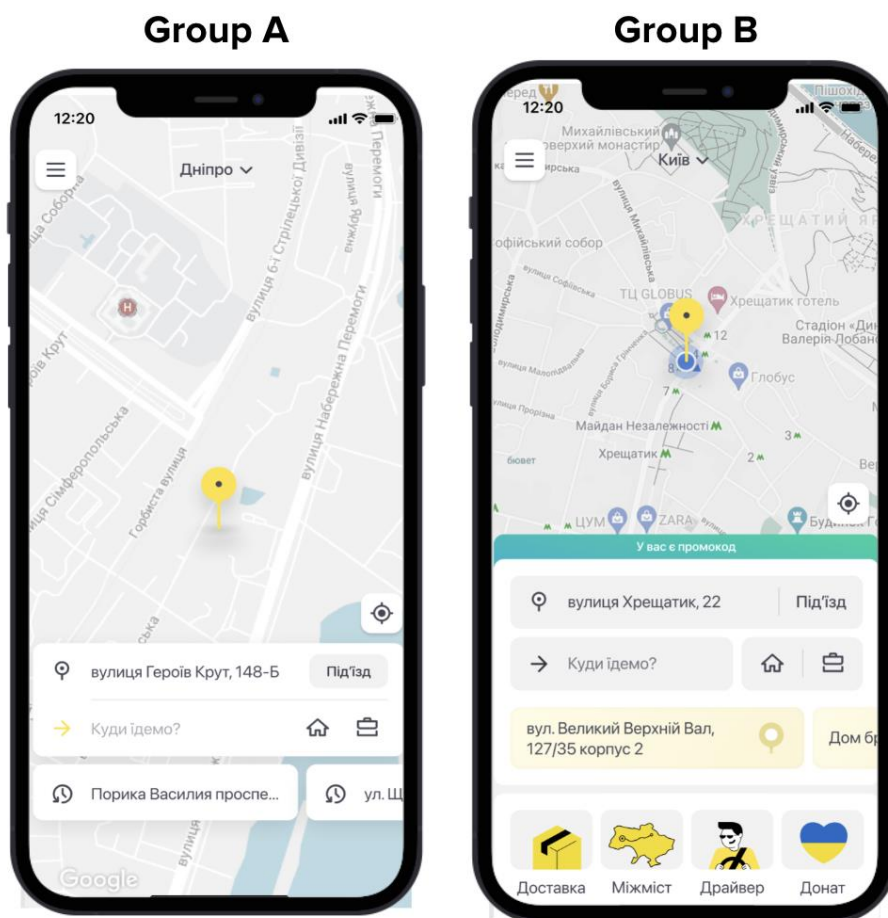


Рисунок 1.2.2.1 – Інтерфейс мобільного додатку Uklon

#### **Переваги:**

- безготівкова оплата через додаток;
- можливість відстежити переміщення автомобіля до вказаної адреси.

#### **Недоліки (з відгуків клієнтів[7]):**

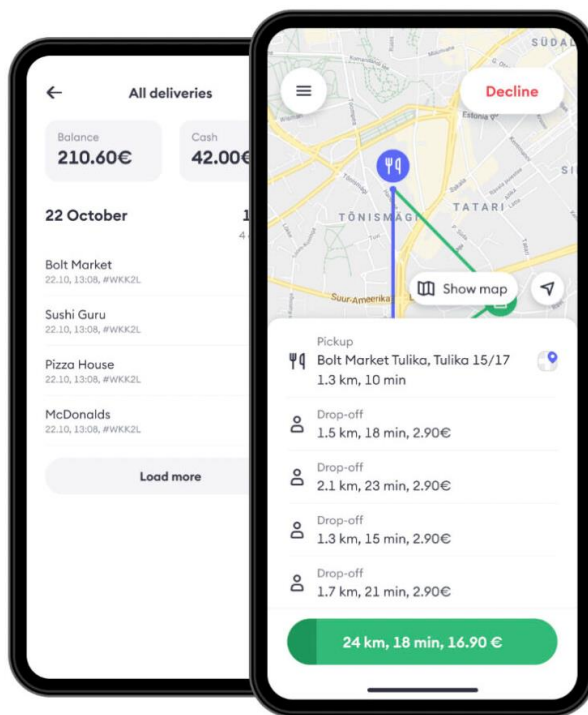
- незручний інтерфейс;
- обман клієнтів;
- авто без маркування таксі.

### **1.2.3 Компанія Bolt**

Bolt - естонська приватна компанія, розробник мобільного додатка для замовлення таксі та надання інших послуг[8]. Головний офіс компанії розташований в Талліні. Bolt пропонує два окремі додатки

- для пасажирів та для водіїв. На березень 2019 року компанія активно функціонує в 30 країнах Європи, Африки, Західної Азії, Центральної Америки та Австралії, обслуговуючи понад 25 мільйонів клієнтів. Українська комісія на платформі становить 25%.

Компанію заснував Маркус Вілліг у 2013 році, коли він був 19-річним студентом, з метою об'єднати всі таксі Талліна та Риги в єдину платформу. Сервіс був запущений в серпні 2013 року, і у 2014 році розпочалося розширення на міжнародні ринки.



Рисунок

#### 1.2.3.1 – Інтерфейс мобільного додатку Bolt

##### **Переваги:**

- можливість відстежити переміщення автомобіля до вказаної адреси;
- доступність;
- ціна фіксована.

##### **Недоліки (з відгуків клієнтів [9]):**

- сервіс не відповідає за водіїв;
- великі тарифи;
- водії, які абсолютно не володіють мовою;
- відсутнє замовлення з сайту.

## 2. ПОСТАНОВКА ЗАВДАННЯ

Дана система забезпечує реєстрацію водія, з характеристиками своєї машини, як виконавця замовлення, а також дає можливість клієнту створити замовлення та обрати вільного водія згідно своїх побажань.

**Метою** інформаційної системи з підтримки служби діяльності таксі є підвищення якості обслуговування клієнтів служби таксі, за рахунок надання вибору послуги за його побажаннями.

Для досягнення поставленої цілі необхідно вирішити наступні задачі:

- Побудова ер-моделі
- Набір сутностей
- Набори атрибутів сутностей
- Зв'язки між сутностями
- Ер-модель
- Модель користувачів
- Реляційна модель бази даних
- Реалізація бази даних
- Обґрунтування вибору субд
- Опис таблиць бази даних
- Програмний код створених таблиць
- Схема бази даних, згенерована засобами субд
- Створення користувачів бази даних
- Робота з базою даних

### 3. ПОБУДОВА ER-МОДЕЛІ

#### 3.1. Опис ER-моделі

##### 1 Order (Замовлення)

- *id (ідентифікатор)*: Унікальний номер замовлення
- *Datetime (Дата і час)*: Час замовлення
- *status (статус)*: Статус замовлення, можливі такі варіанти (Очікування машини, поїздка, замовлення виконано, поїздка скасована)
- *RouteID (ідентифікатор маршруту)*: Посилання на маршрут
- *ClientID (ідентифікатор клієнта)*: Посилання на клієнта
- *PaymentID (ідентифікатор оплати)*: Посилання на оплату
- *DriverID (ідентифікатор водія)*: Посилання на водія

##### 2 Taxi (Таксі)

- *id (ідентифікатор)*: Унікальний номер таксі
- *TaxiNumber (номер таксі)*: Номер автомобіля таксі наприклад (СВ 8136 СН)
- *Model (модель)*: Модель автомобіля, наприклад “ McLaren 720s”
- *DateManufacture (дата виробництва)*: Дата виготовлення автомобіля
- *AvailabilityStatus (статус доступності)*: Статус доступності таксі, наприклад (В замовлені, вільне, не активне)

##### 3 Payment (Оплата)

- *id (ідентифікатор)*: Унікальний номер оплати
- *Amount (сума)*: Сума оплати
- *status (статус)*: Статус оплати
- *Time (час оплати)*: Час оплати

##### 4 Client (Клієнт)

- *id (ідентифікатор)*: Унікальний номер клієнта
- *Name (ім'я)*: Ім'я клієнта
- *Surname (прізвище)*: Прізвище клієнта
- *Address (адреса)*: Адреса клієнта (Який закріплюється за клієнтом як домашній, для зручності, не є обов'язковим для заповнення)
- *PhoneNumber (номер телефону)*: Номер телефону клієнта
- *Email (електронна пошта)*: Електронна пошта клієнта
- *BonusesID (ідентифікатор бонусів)*: Посилання на бонуси

- RideID (ідентифікатор поїздки): Посилання на поїздку
- ClientRatingID (ідентифікатор рейтингу клієнта): Посилання на рейтинг клієнта

## **5 Ride (Поїздка)**

- *id (ідентифікатор)*: Унікальний номер поїздки
- StartTime (час початку): Час початку поїздки
- FinishTime (час завершення): Час завершення поїздки
- Status (статус): Статус поїздки
- RideRatingID (ідентифікатор рейтингу поїздки): Посилання на рейтинг поїздки

## **6 Driver (Водій)**

- *id (ідентифікатор)*: Унікальний номер водія
- Name (ім'я): Ім'я водія
- Surname (прізвище): Прізвище водія
- PhoneNumber (номер телефону): Номер телефону водія
- LicenseNumber (номер ліцензії): Номер ліцензії водія
- DriverRatingID (ідентифікатор рейтингу водія): Посилання на рейтинг водія
- TaxiID (ідентифікатор таксі): Посилання на таксі

## **7 DriverRating (Рейтинг водія)**

- *id (ідентифікатор)*: Унікальний номер рейтингу водія
- Rating (рейтинг): Рейтинг водія
- *Comment (коментар)*: Коментар до рейтингу водія

## **8 Bonuses (Бонуси)**

- *id (ідентифікатор)*: Унікальний номер бонусів
- *Point (бали)*: Кількість балів, за якими складається відсоток знижки
- DiscountPercent (відсоток знижки): Відсоток знижки
- ClientID (ідентифікатор клієнта): Посилання на клієнта

## **9 Route (Маршрут)**

- *id (ідентифікатор)*: Унікальний номер маршруту
- StartAddress (адреса старту): Адреса початку маршруту
- FinishAddress (адреса фінішу): Адреса завершення

- Distance (відстань): Відстань між точками вказаними клієнтом, за наявності точок(**StartAddress, FinishAddress** )
- EstimatedTime (розрахунковий час поїздки): приблизний час поїздки

## **10 ClientRating (Рейтинг клієнта)**

- id (ідентифікатор): Унікальний номер рейтингу клієнта
- Rating (рейтинг): Рейтинг клієнта
- Comment (коментар): Коментар до рейтингу клієнта
- PercentCanceledTrips (відсоток скасованих поїздок): Відсоток скасованих поїздок клієнтом

### **3.2.Зв'язки між сутностями**

#### **1. Order (Замовлення):**

- *Один до одного з Route (Маршрут):* Кожне замовлення пов'язане із одним і тільки одним маршрутом.
- *Один до одного з Client (Клієнт):* Кожне замовлення пов'язане із одним і тільки одним клієнтом.
- *Один до одного з Payment (Платіж):* Кожне замовлення пов'язане із одним і тільки одним платежем.
- *Один до одного з Driver (Водій):* Кожне замовлення пов'язане із одним і тільки одним водієм.

#### **2. Taxi (Таксі):**

- *Один до одного з Driver (Водій):* Кожне таксі пов'язане із тільки одним водієм.

#### **3. Client (Клієнт):**

- *Багато до одного з Bonuses (Бонуси):* Кожен клієнт має одну бонусну програму, але одну і ту бонусну програму можуть мати декілька користувачів.
- *Один до багатьох з Ride (Поїздка):* Кожен клієнт може мати багато поїздок, але кожна поїздка пов'язана тільки з одним клієнтом.
- *Один до одного з ClientRating (Рейтинг клієнта):* Кожен клієнт пов'язаний із своїм рейтингом клієнта.

#### **4. Driver (Водій):**

- *Один до одного з DriverRating (Рейтинг водія):* Кожен водій пов'язаний із своїм рейтингом водія.
- *Один до одного з Taxi (Таксі):* Кожен водій пов'язаний із своїм таксі.

### 5. Bonuses (Бонуси):

- *Багато до одного з Client (Клієнт):* Кожен клієнт має одну бонусну програму, але одну і ту бонусну програму можуть мати декілька користувачів.

### 6. Route (Маршрут):

- *Багато до одного з Order (Замовлення):* Багато замовлень можуть мати один і той самий маршрут, але кожне замовлення пов'язане тільки з одним маршрутом.

### 7. ClientRating (Рейтинг клієнта):

- *Один до одного з Client (Клієнт):* Кожен рейтинг клієнта пов'язаний із конкретним клієнтом.

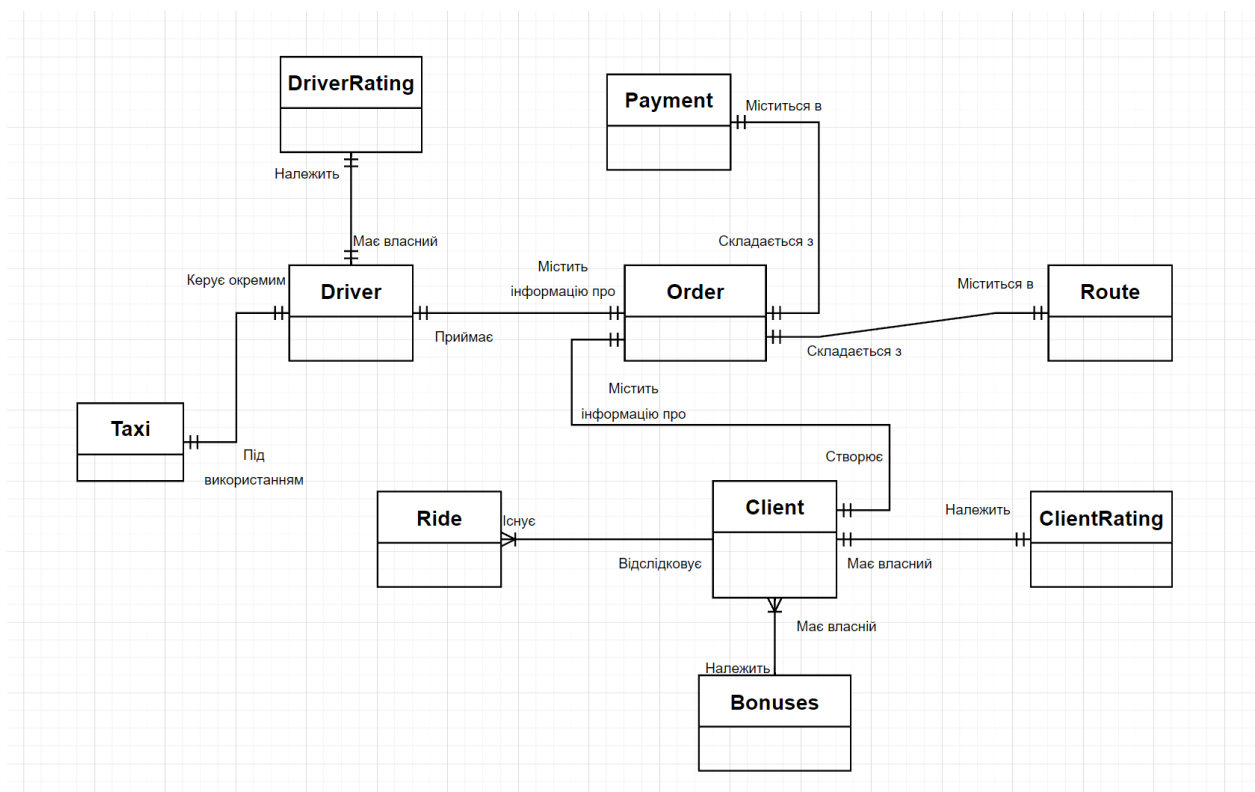


Рисунок 3.2.1 Діаграма "Сутність-зв'язок"

## 3.3.ER-модель



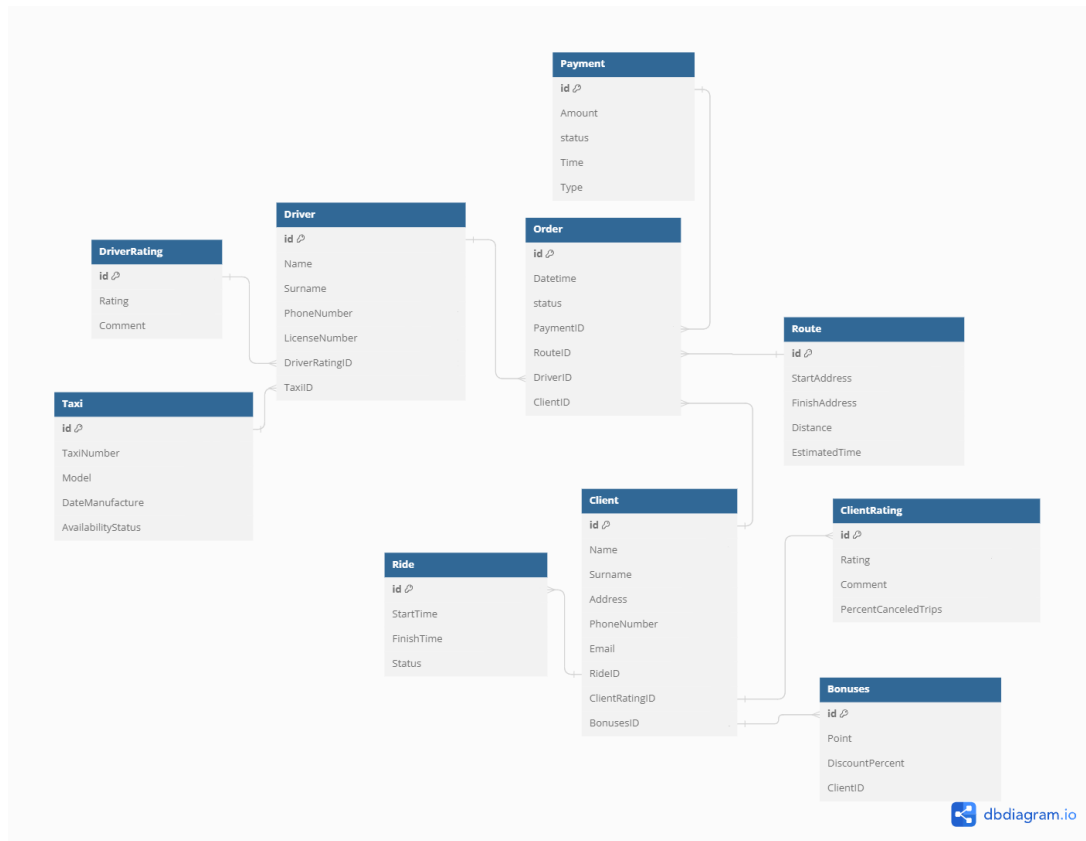


Рисунок 3.3.1 ER-модель

## 4. РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ

OrderTable						
id	DateTime	status	RouteID	ClientID	PaymentID	DriverID
1	2022-03-10	true	1000	200	251	32
2	2020-10-15	true	999	201	252	46
3	2021-03-24	true	998	202	253	2
4	2020-09-23	false	997	203	254	13
5	2020-02-29	false	996	204	255	22

Client									
id	Name	Surname	Address	PhoneNumber	Email	BonusesID	RideID	ClientRatingID	
1	Sunshine	Snalom	413 Garrison Crossing	596-261-1910	ssnalom0@pinterest.com	250	250	1000	
2	Sybil	Gallanders	70559 Northfield Circle	984-492-9645	sgallanders1@networkadvertising.org	251	249	999	
3	Bethanne	Burroughs	1 Paget Crossing	785-997-7750	bburroughs2@theglobeandmail.com	252	248	998	
4	Raymund	Franken	6089 Thierier Road	328-376-2609	rfranken3@amazon.co.uk	253	247	997	
5	Hana	Jent	71 Brown Plaza	479-989-1515	hjent4@xinhuanet.com	254	246	996	

Payment			
id	Amount	status	Time
1	1481.4	true	4:59 AM
2	781.8	true	2:33 PM
3	854.7	false	
4	7687.8	true	9:34 AM
5	2765	true	7:17 PM

Route				
id	StartAddress	FinishAddress	Distance	EstimatedTime
1	1 Corben Crossing	976 Eliot Park	33.7	12:41 PM
2	78188 5th Drive	0002 Maywood Park	55.4	2:13 AM
3	5 Green Ridge Junction	5 Harbort Lane	36.8	11:52 PM
4	59638 Summerview Park	01226 Muir Plaza	31.8	3:17 AM
5	2 Glendale Circle	69957 Cambridge Terrace	11.2	3:00 PM

Bonuses		
id	Points	DiscountPercent
1	4600	23.3
2	1700	7.5
3	1800	4.6
4	1300	98.5
5	3800	73.8

ride			
id	startTime	FinishTime	Status
1	10:02 AM	11:43 PM	false
2	11:45 AM	1:08 PM	true
3	10:31 AM	9:44 PM	false
4	5:51 AM	4:43 PM	true
5	2:41 AM	5:04 PM	true

Driver						
id	Name	Surname	PhoneNumber	LicenseNumber	DriverRatingID	TaxiID
1	Iosep	Sommerling	752-719-1771	9525976610	100	51
2	Edee	Itter	168-419-1842	3971237665	99	52
3	Aggie	Eissold	754-796-0454	2455272982	98	53
4	Thedrick	Dabell	250-679-1601	6772383133	97	54
5	Lock	Took	742-111-8975	2389217907	96	55

DriverRating			
id	Rating	Comment	
1	52	Enterprise-wide global pricing structure	
2	6.7	Diverse scalable info-mediaries	
3	65.2	Reverse-engineered regional open architecture	
4	19.7	Right-sized attitude-oriented access	
5	20.6	Extended needs-based knowledge base	

Taxi				
id	TaxiNumber	Model	DateManufacture	AvailabilityStatus
1	2964944091	BMW	2011	false
2	7849798635	Jaguar	1997	false
3	4719870678	Buick	1996	false
4	4464530974	Mazda	1998	false
5	2212077211	Toyota	2012	true

ClientRating			
id	Rating	Comment	PercentCanceledTrips
1	38.1	Reactive executive initiative	89
2	40.4	Decentralized exploit firmware	90.1
3	63.1	Assimilated full-range service-desk	49.8
4	10	Operative multi-tasking archive	28
5	92.9	Progressive upward-trending productivity	19.5

Рисунок 4.1 Реляційна модель бази даних

## 5. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

### 5.1. Обґрунтування вибору СУБД

#### Обґрунтування вибору MySQL:

- Відкритість і безкоштовність: MySQL є відкритою програмою з ліцензією GPL, що дозволяє використовувати, змінювати і розповсюджувати код безкоштовно. Це робить його доступним для широкого кола користувачів і підприємств, зокрема для студентів та невеликих служб таксі, де обмежений бюджет.
- Висока продуктивність і швидкодія: MySQL відомий своєю високою продуктивністю та швидкістю обробкою запитів. Це особливо важливо для служб таксі, де частість операцій, таких як обробка замовлень і відстеження руху автомобілів, вимагає миттєвої реакції.
- Широкий спектр функцій і можливостей: MySQL має багатий набір функцій і можливостей, таких як транзакції, індексація, забезпечення цілісності даних тощо. Це забезпечує ефективну роботу з базою даних таксі, зокрема для відстеження подій та забезпечення надійності даних.
- Велика спільнота і підтримка: MySQL має велику спільноту користувачів і розробників, що важливо для отримання швидкої підтримки та вирішення проблем. Завдяки активній спільноті можна легко знайти відповіді на питання, а також скористатися різноманітними ресурсами, такими як форуми та документація.
- Інтеграція з іншими технологіями: MySQL легко інтегрується з іншими технологіями та мовами програмування. Це робить його універсальним і гнучким рішенням для будь-якого середовища розробки, яке може використовувати служба таксі.

MySQL є оптимальним рішенням для бази даних служби таксі через свою високу продуктивність, безкоштовність та широкий функціонал, що забезпечує ефективну роботу та надійність системи.

## 5.2.Опис таблиць бази даних

Ім'я сутності	Ім'я поля	Тип даних	Розмір	Ключ	Опис
Payment	Id	Int		<b>PK</b>	Унікальний ідентифікатор оплати
	Amount	Double			Ціна поїздки
	Status	Bool			Статус поїздки(true - оплачено)
	Time	TimeStamp			Час оплати(за наявності)
OrderTable	Id	Int		<b>PK</b>	Унікальний ідентифікатор замовлення
	DateTime	TimeStamp			Час створення замовлення
	Status	Bool			Статус замовлення(true – виконано)
	RouteID	Int		FK	Ідентифікатор маршруту
	ClientID	Int		FK	Ідентифікатор клієнта, який зробив замовлення
	PaymentID	Int		FK	Ідентифікатор оплати
	DriverID	int		FK	Ідентифікатор водія який виконує замовлення
Route	Id	Int		<b>PK</b>	Унікальний ідентифікатор маршруту
	StartAddress	Varchar	128		Початкова точка маршруту
	FinishAddress	Varchar	128		Кінцева точка маршруту
	Distance	Double			Відстань поїздки
	EstimatedTime	Varchar	32		Приблизний час поїздки

Driver	Id	Int		<b>PK</b>	Унікальний ідентифікатор водія
	Name	Varchar	64		Ім'я водія
	Surname	Varchar	64		Прізвище водія
	PhoneNumber	Varchar	16		Номер телефону водія
	LicenceNumber	Int		FK	Номер ліцензії водія
	DriverRatingID	Int		FK	Ідентифікатор рейтингу водія
	TaxiID	Int		FK	Ідентифікатор автомобіля, на якому виконується замовлення
DriverRating	Id	Int		<b>PK</b>	Унікальний ідентифікатор рейтингу водія
	Rating	Double			Рейтинг водія
	Comment	Text	2048	FK	Коментарі водія
Taxi	Id	Int		<b>PK</b>	Унікальний ідентифікатор автомобіля
	TaxiNumber	Int			Значення номерного знаку автомобіля
	Model	Varchar	64		Модель автомобіля
	DateManufacture	Int			Рік випуску автомобіля
	AvalibilityStatus	Bool			Статус авто(true – вільне)
Client	Id	Int		<b>PK</b>	Унікальний ідентифікатор користувача
	Name	Varchar	64		Ім'я клієнта
	Surname	Varchar	64		Прізвище клієнта

	Address	Varchar	128		Домашня адреса клієнта
	PhoneNumber	Varchar	16		Номер телефону клієнта
	Email	Varchar	256		Електронна пошта клієнта
	BonusesID	Int		FK	Ідентифікатор бонусів клієнта
	RideID	Int		FK	Ідентифікатор мар
	ClientRatingID	Int		FK	Ідентифікатор маршруту клієнта
Ride	Id	Int		<b>PK</b>	Унікальний ідентифікатор поїздки
	StartTime	TimeStamp			Час початку поїздки
	FinishTime	TimeStamp			Час кінця поїздки
	Status	Bool			Статус поїздки(true – завершено)
Bonuses	Id	Int		<b>PK</b>	Унікальний ідентифікатор бонусів
	Points	Int			Бонусні бали користувача
	DiscountPercent	Double		FK	Відсоток знижки користувача
ClientRating	Id	Int		<b>PK</b>	Унікальний ідентифікатор рейтингу користувача
	Rating	Double			Рейтинг клієнта
	Comment	Text	2048		Коментарі надані клієнту
	PercentCancelledTrips	Double			Відсоток скасованих поїздок

Таблиця 5.3.1 Опис таблиць бази даних

### 5.3. Програмний код створених таблиць

```
CREATE TABLE OrderTable ( -- 1000
```

```
id int PRIMARY KEY,
```

```
Datetime timestamp,
```

```
status bool,          -- 1 == done, 0 == processing
```

```
RouteID int,
```

```
ClientID int,
```

```
PaymentID int,
```

```
DriverID int
```

```
);
```

```
CREATE TABLE Taxi ( -- 100 шт
```

```
id int PRIMARY KEY,
```

```
TaxiNumber varchar(10),
```

```
Model varchar(20),
```

```
DateManufacture DATE,
```

```
AvailabilityStatus bool      -- 1 == free, 0 == busy
```

```
);
```

```
CREATE TABLE Payment ( -- 1000
```

```
id int PRIMARY KEY,
```

```
Amount float(5,2),
```

```
status bool,          -- 1 == done, 0 == processing
```

```
Time timestamp );
```

```
CREATE TABLE Client ( -- 1000
```

```
id int PRIMARY KEY,
```

```
Name varchar(32),
```

```
Surname varchar(32),
```

```
Address varchar(64),
```

```
PhoneNumber int,
```

```
Email varchar(64),
```

```
BonusesID int,
```

```
RideID int,
```

```
ClientRatingID int
```

```
);
```

```
CREATE TABLE Ride ( -- 1000
  id int PRIMARY KEY,
  StartTime timestamp,
  FinishTime timestamp,
  Status bool          -- 1 == done, 0 == processing
);
```

```
CREATE TABLE Driver (-- 100
  id int PRIMARY KEY,
  Name varchar(32),
  Surname varchar(32),
  PhoneNumber int,
  LicenseNumber varchar(64),
  DriverRatingID int,
  TaxiID int
);
```

```
CREATE TABLE DriverRating ( -- 100
  id int PRIMARY KEY,
  Rating float(2, 1),
  Comment text
);
```

```
CREATE TABLE Bonuses ( -- 1000
  id int PRIMARY KEY,
  Points int,
  DiscountPercent float(2, 1),
);
```

```
CREATE TABLE Route ( -- 1000
  id int PRIMARY KEY,
  StartAddress varchar(64),
  FinishAddress varchar(64),
  Distance float(3, 2),
  EstimatedTime time
);
```

```

CREATE TABLE ClientRating ( -- 1000
    id int PRIMARY KEY,
    Rating float(3, 1),
    Comment text,
    PercentCanceledTrips float(2, 1)
);

```

#### 5.4.Схема бази даних, згенерована засобами СУБД

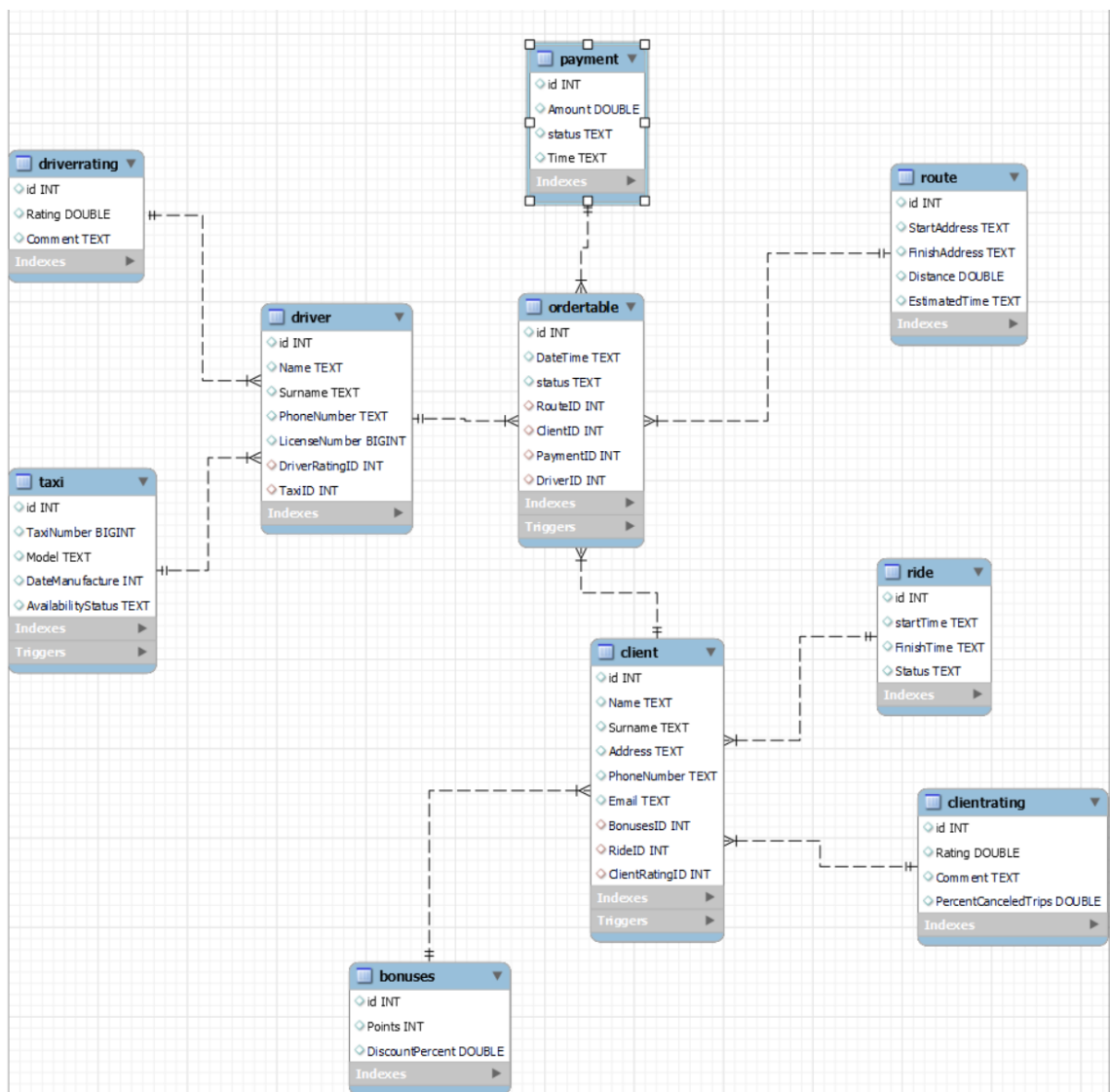


Рисунок 5.4.1 Схема бази даних, згенерована засобами СУБД MYSQL



## 6. СТВОРЕННЯ КОРИСТУВАЧІВ БАЗИ ДАНИХ

### 6.1. Текст створення користувачів

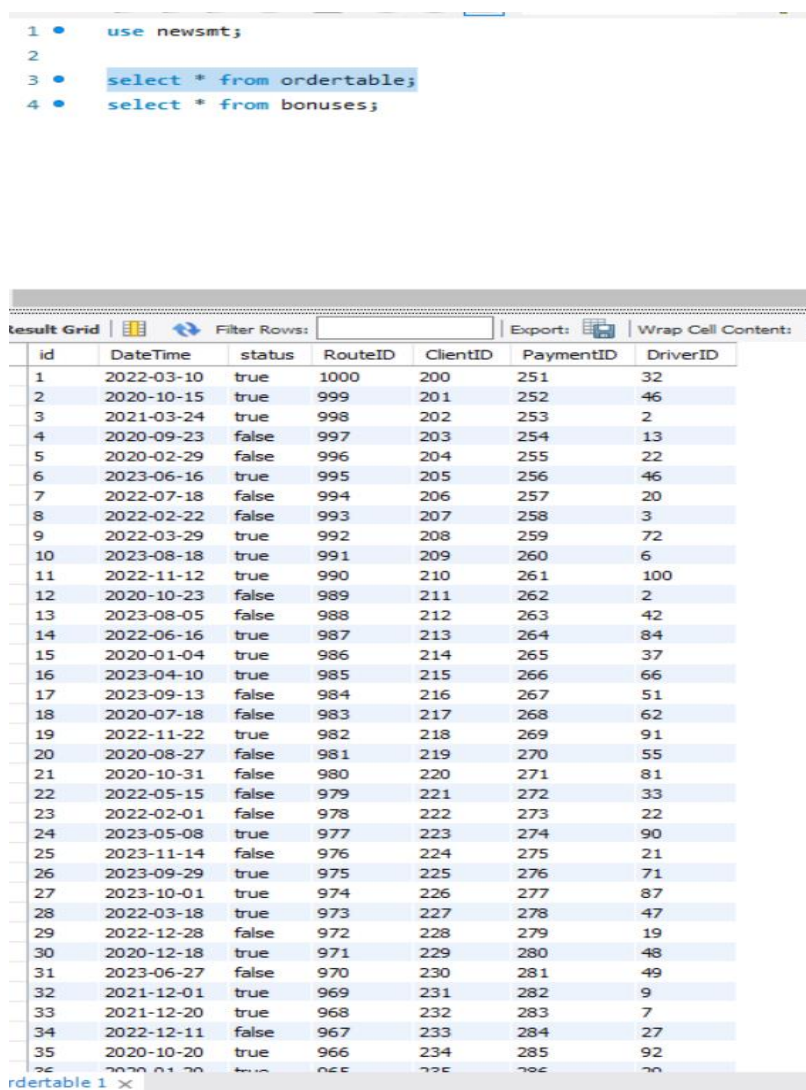
```
create user 'adminAnna'@'localhost' identified By '12345678';
```

```
create user 'DriverMikle'@'localhost' identified By '12345678';
```

```
grant select, insert on newsmt.ordertable to 'adminAnna'@'localhost';
```

```
grant select on newsmt.ordertable to 'DriverMikle'@'localhost';
```

На рисунках 6.1.1 – 6.1.5 показані результати запитів користувачів 'adminAnna' та 'DriverMikle'



```
1 • use newsmt;
2
3 • select * from ordertable;
4 • select * from bonuses;
```

id	DateTime	status	RouteID	ClientID	PaymentID	DriverID
1	2022-03-10	true	1000	200	251	32
2	2020-10-15	true	999	201	252	46
3	2021-03-24	true	998	202	253	2
4	2020-09-23	false	997	203	254	13
5	2020-02-29	false	996	204	255	22
6	2023-06-16	true	995	205	256	46
7	2022-07-18	false	994	206	257	20
8	2022-02-22	false	993	207	258	3
9	2022-03-29	true	992	208	259	72
10	2023-08-18	true	991	209	260	6
11	2022-11-12	true	990	210	261	100
12	2020-10-23	false	989	211	262	2
13	2023-08-05	false	988	212	263	42
14	2022-06-16	true	987	213	264	84
15	2020-01-04	true	986	214	265	37
16	2023-04-10	true	985	215	266	66
17	2023-09-13	false	984	216	267	51
18	2020-07-18	false	983	217	268	62
19	2022-11-22	true	982	218	269	91
20	2020-08-27	false	981	219	270	55
21	2020-10-31	false	980	220	271	81
22	2022-05-15	false	979	221	272	33
23	2022-02-01	false	978	222	273	22
24	2023-05-08	true	977	223	274	90
25	2023-11-14	false	976	224	275	21
26	2023-09-29	true	975	225	276	71
27	2023-10-01	true	974	226	277	87
28	2022-03-18	true	973	227	278	47
29	2022-12-28	false	972	228	279	19
30	2020-12-18	true	971	229	280	48
31	2023-06-27	false	970	230	281	49
32	2021-12-01	true	969	231	282	9
33	2021-12-20	true	968	232	283	7
34	2022-12-11	false	967	233	284	27
35	2020-10-20	true	966	234	285	92

Рисунок 6.1.1

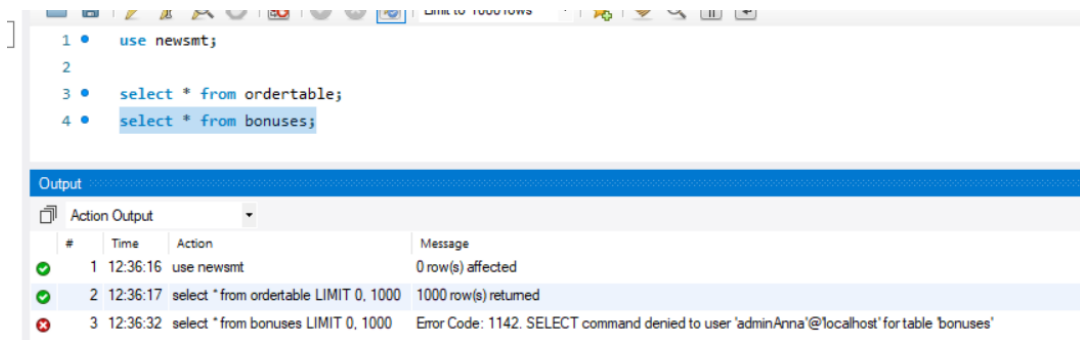


Рисунок 6.1.2

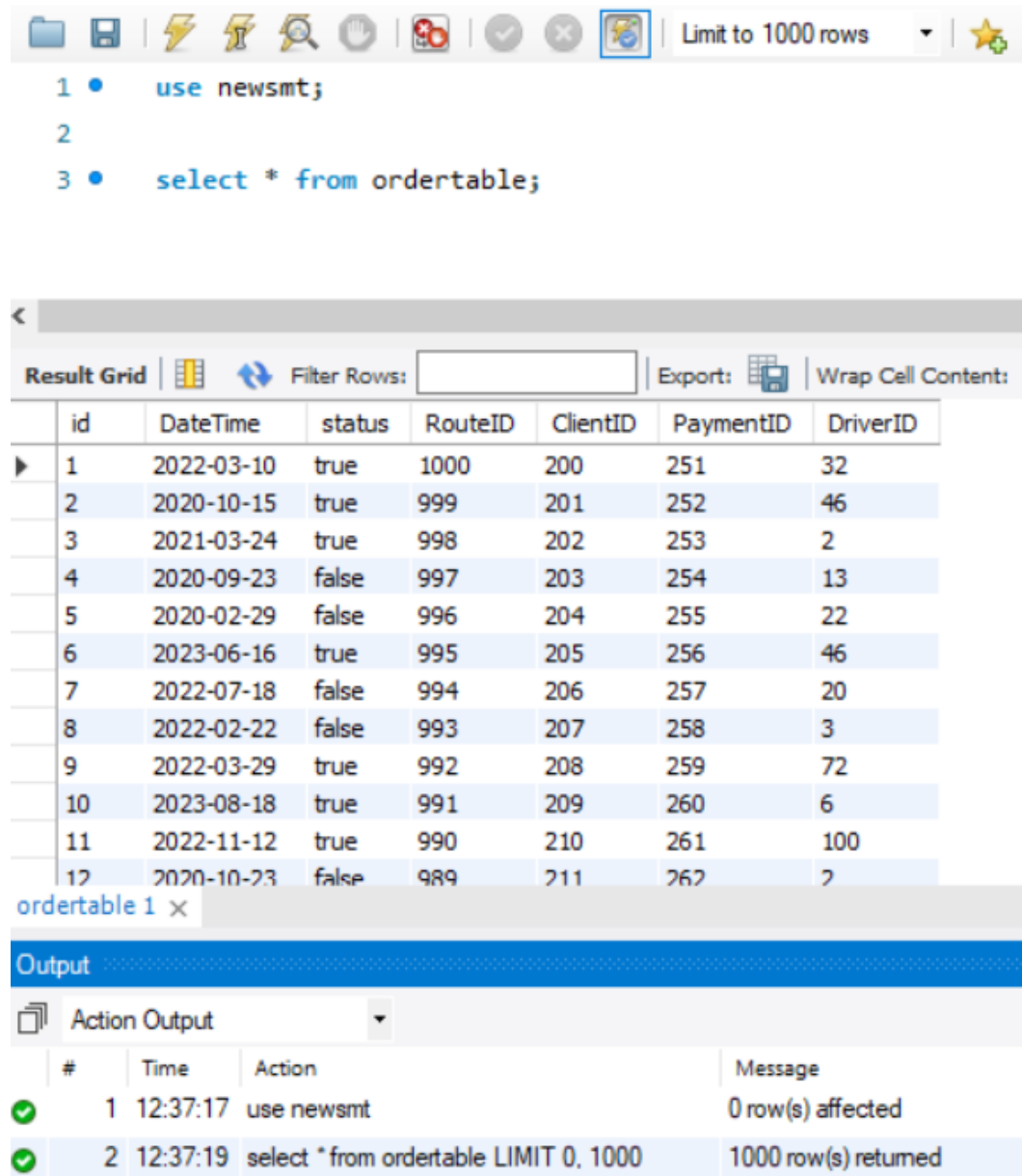


Рисунок 6.1.3

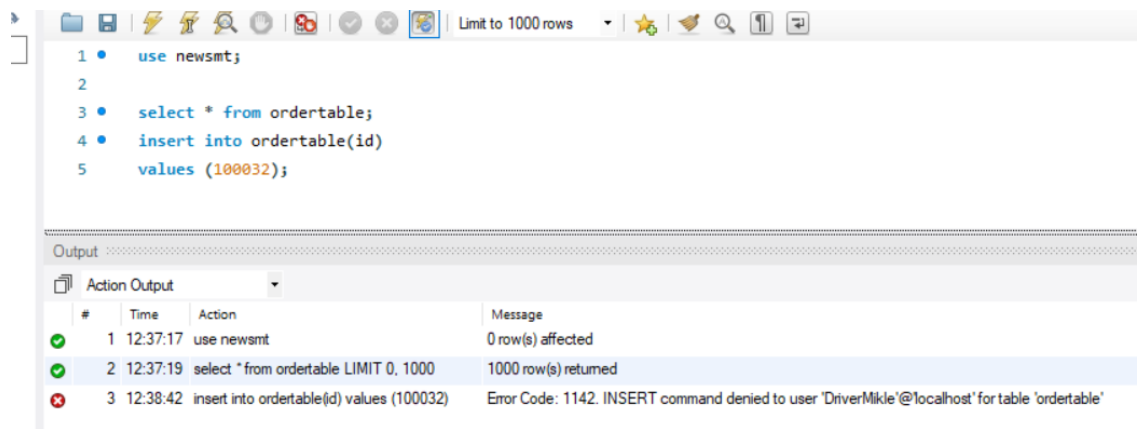


Рисунок 6.1.4

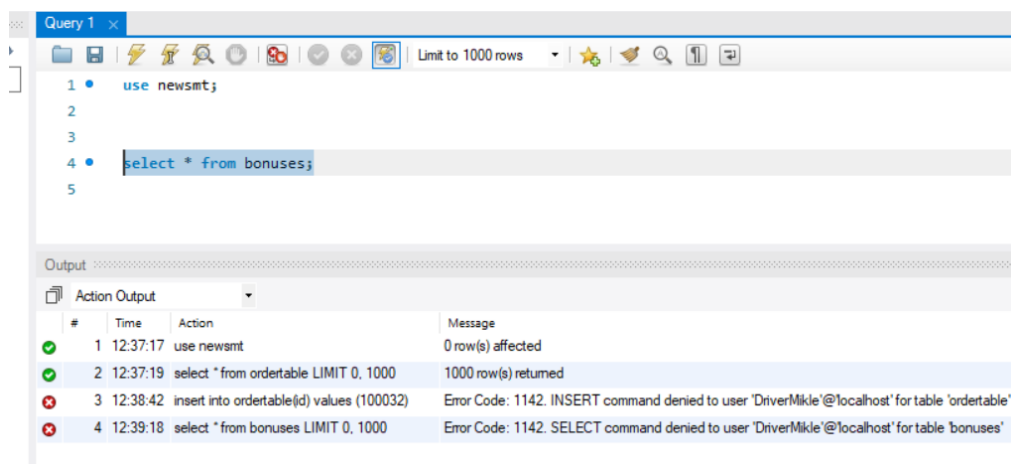


Рисунок 6.1.5

## 7. РОБОТА З БАЗОЮ ДАНИХ

### 7.1.Тексти генераторів

```
create table log(  
id int primary key auto_increment,  
Comment text,  
row_id int,  
Time datetime  
);
```

### 7.2.Тексти збережених процедур/функцій

```
-- procedure  
-- 1)  
DELIMITER //  
  
CREATE PROCEDURE GetAllDriverShortInfo()  
  
BEGIN  
  
    select Name , Surname, PhoneNumber from Driver;  
  
END //  
  
DELIMITER ;  
  
CALL GetAllDriverShortInfo();
```

	Name	Surname	PhoneNumber
►	Iosep	Sommerling	752-719-1771
	Edee	Itter	168-419-1842
	Aggie	Eisold	754-796-0454
	Thedrick	Dabell	250-679-1601
	Lock	Took	742-111-8975
	Rhoda	Wolland	686-570-2166
	Elke	Fitzsimon	811-213-6730
	Marnie	Ruffli	387-417-5398
	Bernhard	Spilsbury	841-672-6080
	Nikolia	Farris	818-691-9869
	Flora	Jefferys	847-929-3698
	Maroareta	Ramsbotham	247-956-2553

Рисунок 7.2.1

```
-- 2) 3 1 вхід парам  
DELIMITER //  
  
CREATE PROCEDURE ChooseOrdersInTimePeriod(in TimeFrom Datetime, in TimeTo Datetime)  
  
BEGIN  
  
    select * from OrderTable where DateTime between TimeFrom and TimeTo;  
  
END //
```

DELIMITER ;

select \* from ordertable;

call ChooseOrdersInTimePeriod('2021-01-01','2021-01-31');



	id	DateTime	status	RouteID	ClientID	PaymentID	DriverID
▶	44	2021-01-05	false	957	243	294	100
	48	2021-01-12	false	953	247	298	26
	105	2021-01-31	false	896	304	355	75
	156	2021-01-11	false	845	355	406	56
	181	2021-01-12	true	820	380	431	93
	328	2021-01-12	true	673	478	78	92
	349	2021-01-12	false	652	499	99	70
	350	2021-01-17	false	651	500	100	15
	368	2021-01-02	true	633	518	118	42
	369	2021-01-12	false	632	519	119	63
	437	2021-01-12	true	564	587	187	70
	457	2021-01-26	true	544	308	207	11
	472	2021-01-26	true	529	323	222	1
	481	2021-01-31	true	520	332	231	21
	581	2021-01-13	false	420	432	831	19
	591	2021-01-13	false	410	442	841	51
	619	2021-01-02	false	382	819	869	38
	669	2021-01-06	true	332	869	919	18
	671	2021-01-24	true	330	871	921	6
	684	2021-01-07	true	317	884	934	23
	706	2021-01-14	true	295	906	956	14
	739	2021-01-12	false	262	939	989	30
	849	2021-01-14	false	152	649	598	71
	885	2021-01-01	true	116	685	634	33
	887	2021-01-01	true	114	687	636	72

Рисунок 7.2.2

-- 3) з 2 парам 1 вхід 1 вихід

DELIMITER //

CREATE PROCEDURE ClientStartSameNameCounter(in RName varchar(20), out ClientCount int)

BEGIN

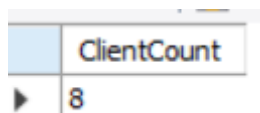
select count(name) into ClientCount from Client where name like concat(RName, '%');

END //

DELIMITER ;

call ClientStartSameNameCounter('Ad', @var);

SELECT @var AS ClientCount;



ClientCount
8

Рисунок 7.2.3 значення змінної var

-- 4)

DELIMITER //

```

CREATE PROCEDURE GetInProgressOrdersByRoute(IN routeId INT)
BEGIN
    SELECT
        ot.id AS OrderID,
        ot.Datetime AS OrderDatetime,
        ot.status AS OrderStatus,
        c.id AS ClientID,
        c.Name AS ClientName,
        c.Surname AS ClientSurname,
        d.id AS DriverID,
        d.Name AS DriverName,
        d.Surname AS DriverSurname
    FROM
        OrderTable ot
    JOIN
        Client c ON ot.ClientID = c.id
    JOIN
        Driver d ON ot.DriverID = d.id
    WHERE
        ot.status = 0 -- Заовлення в обробці
        AND ot.RouteID = routeId;
END //

DELIMITER ;

```

```
call GetInProgressOrdersByRoute(1);
```

	OrderID	OrderDatetime	OrderStatus	ClientID	ClientName	ClientSurname	DriverID	DriverName	DriverSurname
►	1000	2023-07-07	false	800	Fonsie	Boom	13	Peyter	Fawson

Рисунок 7.2.4

```

-- 5) if()

DELIMITER //

CREATE PROCEDURE CheckUser(IN p_username VARCHAR(255))
BEGIN
    DECLARE user_count INT;

```

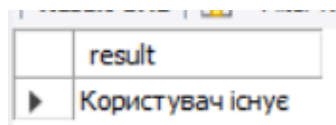
```

SELECT COUNT(*) INTO user_count
FROM client
WHERE Client.Name = p_username;

IF user_count > 0 THEN
    SELECT 'Користувач існує' AS result;
ELSE
    SELECT 'Користувач не знайдений' AS result;
END IF;
END //
DELIMITER ;

```

```
call CheckUser('Sunshine');
```



result	Користувач існує
--------	------------------

Рисунок 7.2.5

```

-- 6)
DELIMITER //

create procedure IncreaseClientPoints(in Percent float, in Amount int)
begin
    if Percent <> 0 then
        update bonuses
        set Points = Points + Points * Percent;
    end if;

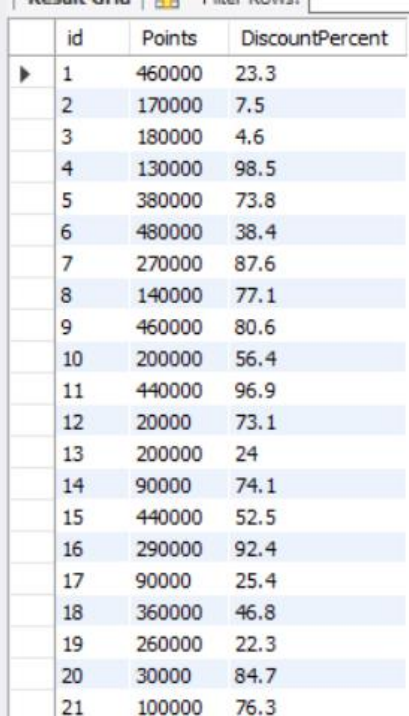
    if Amount <> 0 then
        update bonuses
        set Points = Points + Amount;
    end if;

    select * from bonuses;
end //
DELIMITER //;

```

```
select * from bonuses;
```

```
call IncreaseClientPoints(100,0 );
```



	id	Points	DiscountPercent
▶	1	460000	23.3
	2	170000	7.5
	3	180000	4.6
	4	130000	98.5
	5	380000	73.8
	6	480000	38.4
	7	270000	87.6
	8	140000	77.1
	9	460000	80.6
	10	200000	56.4
	11	440000	96.9
	12	20000	73.1
	13	200000	24
	14	90000	74.1
	15	440000	52.5
	16	290000	92.4
	17	90000	25.4
	18	360000	46.8
	19	260000	22.3
	20	30000	84.7
	21	100000	76.3

Рисунок 7.2.6

```
-- function (4)
```

```
-- 7)
```

```
select * from bonuses;
```

```
Delimiter //
```

```
create function BonusLevel( points int )
```

```
returns varchar (32)
```

```
begin
```

```
    declare BonusLevel varchar(32);
```

```
    if points < 2000 then
```

```
        set BonusLevel = "Normal";
```

```
    elseif (points >= 2000 and points < 3000) then
```

```
        set BonusLevel = "Gold";
```

```
    elseif points >=3000 then
```

```
        set BonusLevel = "Elit";
```

```
    end if;
```

```
    return BonusLevel;
```



```
end //
```

```
delimiter ;
```

```
select c.name, c.surname, BonusLevel(b.Points)
```

```
from Client c
```

```
left join bonuses b on c.bonusesID = b.id;
```

	name	surname	BonusLevel(b.Points)
►	Zorine	Klimas	Elit
	Zeke	Rolstone	Elit
	Yvor	Symcock	Elit
	Yvor	Denacamp	Elit
	Yoshiko	Gon	Elit
	Yorker	Rivalland	Elit
	Yasmeen	Bengall	Elit
	Yard	Ratazzi	Elit
	Yard	Gianolini	Elit
	Yancy	Husbands	Elit
	Yancey	Pashba	Elit
	Xaviera	Merrigans	Normal
	Xavier	Cameli	Elit
	Winnie	Tice	Elit
	Win	Dicker	Elit
	Willyt	Bickerstasse	Elit
	Willie	Christofe	Elit
	Willie	Jeans	Elit
	Willie	Edess	Elit
	Wilhelmine	Flag	Elit
	Weylin	Chrisp	Elit
	Weston	Drewell	Elit
	West	Zmitruk	Elit

Рисунок 7.2.7 результат запиту з використанням збереженої функції

```
-- 8)
```

```
DELIMITER //
```

```
CREATE FUNCTION no_of_years(date1 date)
```

```
RETURNS int
```

```
BEGIN
```

```
DECLARE date2 DATE;
```

```
SELECT current_date()into date2;
```

```
RETURN year(date2)-year(date1);
```

```
END
```

```
//
```

```
DELIMITER ;
```

```
select o.id AS OrderID, o.status as IsDone, no_of_years(o.datetime) as NumYearsAgo
```

```
from ordertable o;
```

	OrderID	IsDone	NumYearsAgo
▶	1	true	1
	2	true	3
	3	true	2
	4	false	3
	5	false	3
	6	true	0
	7	false	1
	8	false	1
	9	true	1
	10	true	0
	11	true	1
	12	false	3
	13	false	0
	14	true	1
	15	true	3
	16	true	0
	17	false	0
	18	false	3
	19	true	1
	20	false	3

Рисунок 7.2.8 результат запиту з використанням збереженої функції

-- 9)

```
select * from route;
```

```
DELIMITER //
```

```
CREATE FUNCTION calcProfit(Distance float)
```

```
RETURNS float
```

```
BEGIN
```

```
    DECLARE profit float;
```

```
    SET profit = Distance * 0.5;
```

```
    RETURN profit;
```

```
END //
```

```
DELIMITER ;
```

```
SELECT *, calcProfit(distance) AS KoefOfProfit FROM route;
```

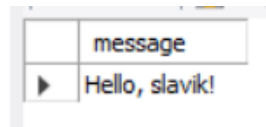
	id	StartAddress	FinishAddress	Distance	EstimatedTime	KoefOfProfit
▶	1	1 Corben Crossing	976 Eliot Park	33.7	12:41 PM	16.85
	2	78188 5th Drive	0002 Maywood Park	55.4	2:13 AM	27.7
	3	5 Green Ridge Junction	5 Harbort Lane	36.8	11:52 PM	18.4
	4	59638 Summerview Park	01226 Muir Plaza	31.8	3:17 AM	15.9
	5	2 Glendale Circle	69957 Cambridge Terrace	11.2	3:00 PM	5.6
	6	08 Lighthouse Bay Trail	238 Mosinee Terrace	26.8	7:52 AM	13.4
	7	092 Prentice Lane	51 Glendale Lane	40.7	12:34 AM	20.35
	8	9 Debs Trail	4 Maple Wood Alley	62.5	12:22 PM	31.25
	9	976 Melby Drive	130 Claremont Drive	21.4	3:51 PM	10.7
	10	37364 Hoffman Trail	32 Hansons Way	39.7	3:58 PM	19.85
	11	3855 High Crossing Par...	4 Di Loreto Terrace	69.7	6:24 PM	34.85
	12	291 Crescent Oaks Court	7 Ilene Point	29.3	10:13 PM	14.65
	13	208 Doe Crossing Point	54211 Debra Junction	7.3	7:58 PM	3.65
	14	9319 Gateway Road	630 Sunfield Junction	44.9	9:06 PM	22.45
	15	778 Calypso Street	853 Spohn Plaza	64.9	2:31 AM	32.45
	16	46 Corry Terrace	2902 Buena Vista Junction	11.8	5:06 PM	5.9
	17	2241 Shasta Pass	939 South Terrace	45.7	12:00 PM	22.85
	18	2815 Melrose Alley	9615 Rowman Park	79.7	1:43 PM	39.85

Рисунок 7.2.9 результат запиту з використанням збереженої функції

```
-- 10)
DELIMITER //
CREATE FUNCTION hello (s CHAR(20))
RETURNS CHAR(50)
RETURN CONCAT('Hello, ',s,'!');

// DELIMITER ;

select hello('slavik') as message;
```



message
Hello, slavik!

Рисунок 7.2.10 результат запиту з використанням збереженої функції

### 7.3. Тексти тригерів

```
use newsmt;

create table log(
id int primary key auto_increment,
Comment text,
row_id int,
Time datetime
);

-- 1)
DELIMITER //
create trigger insertInt after insert on Client
for each row
begin
insert into log(Comment, row_id, Time)
values ('insert', New.id, NOW());
end //
DELIMITER ;
```

```
insert into Client values(10023, 'TestName', 'TestSurname', '78 GoodS treet', 343-234-1233, 'testemail@gmail.com', null, null, null);
```

```
select * from log;
```

```
select* from client;
```

```
DELIMITER //
```

```
CREATE TRIGGER update_taxi
```

```
BEFORE UPDATE ON Taxi
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO log(Comment, row_id, Time)
```

```
VALUES (CONCAT('Update Taxi: OldValue - ', OLD.TaxiNumber, ', ', OLD.Model, ', ', OLD.DateManufacture, ', ', OLD.AvailabilityStatus,
```

```
'; NewValue - ', NEW.TaxiNumber, ', ', NEW.Model, ', ', NEW.DateManufacture, ', ', NEW.AvailabilityStatus), NEW.id, NOW());
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
update taxi
```

```
set Model = 'BMW'
```

```
where id = 1;
```

```
-- 3)
```

```
DELIMITER //
```

```
CREATE TRIGGER delete_client
```

```
BEFORE delete ON client
```

```
FOR EACH ROW
```

```
BEGIN
```

```
INSERT INTO log(Comment, row_id, Time)
```

```
VALUES (CONCAT('Deleted value from Client- ', OLD.Name, ', ', OLD.Surname, ', ', OLD.Address, ', ', OLD.PhoneNumber, ', ', OLD.Email), OLD.id, NOW());
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
-- 4)
```

```
DELIMITER //
```

```

CREATE TRIGGER AfterUpdateOrderStatus
AFTER UPDATE ON OrderTable
FOR EACH ROW
BEGIN
    IF NEW.status = 'true' THEN
        UPDATE Taxi
        SET AvailabilityStatus = 'true'
        WHERE id = (SELECT TaxiID FROM Driver WHERE id = NEW.DriverID);
    END IF;
END;
//
DELIMITER ;

```

```

update orderTable
set status = 'true'
where id = 2;

```

```
-- 5)
```

```
DELIMITER //
```

```

CREATE TRIGGER AfterInsertOrder
AFTER INSERT ON OrderTable
FOR EACH ROW
BEGIN

```

```
    DECLARE bonusPoints INT;
```

```
    SET bonusPoints = (select ROUND(payment.Amount * 0.1) from payment where
payment.id = (select paymentID from orderTable where id = new.id ));
```

```
    UPDATE Bonuses SET Points = Points + bonusPoints WHERE id = (select BonusesID
from client where id = (select clientid from ordertable where id = new.id));
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
INSERT INTO orderTable values(1001, '2023-26-12', 'true', null, 1, 1, null);
```

```
-- client 1 = 250 bonusid = 803 point
```

```
-- payment 1 = '1481.4' amount
```

```
-- 803 + cast(1481.4 * 0.1 to int) = 803 + 148 = 951 points in client 1 with bonus id = 250
```

```
select * from bonuses;
```

```
select * from payment;
```

```
select * from client;
```

```
show triggers;
```

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client
insertInt	INSERT	client	begin insert into log(Comment, row_id, Time) ...	AFTER	2023-12-26 13:08:48.97	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4
delete_client	DELETE	client	BEGIN INSERT INTO log(Comment, row_id, Ti...	BEFORE	2023-12-26 13:25:09.85	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4
AfterInsertOrder	INSERT	ordertable	BEGIN DECLARE bonusPoints INT; SET b...	AFTER	2023-12-26 11:57:44.74	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4
AfterUpdateOrderStatus	UPDATE	ordertable	BEGIN IF NEW.status = 'true' THEN UPD...	AFTER	2023-12-26 11:37:13.29	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4
update_taxi	UPDATE	taxi	BEGIN INSERT INTO log(Comment, row_id, Ti...	BEFORE	2023-12-26 13:14:31.91	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	root@localhost	utf8mb4

Рисунок 7.3.1 Всі створені тригери

## Результати роботи

	id	Comment	row_id	Time
▶	1	insert	10023	2023-12-25 21:51:36
	2	Update Taxi: OldValue - 2964944091, Ford, 2011, false; NewValue - 2964944091, BMW, 2011, false	1	2023-12-26 13:15:42
	5	insert	1001	2023-12-26 13:26:46

Рисунок 7.3.2 Результуюча таблиця log з інформацією про спрацювання тригерів

## 7.4. Тексти представлень

```
use newsmt;
```

```
create view DriverInfo as
```

```
select ROW_NUMBER() OVER (ORDER BY dr.rating desc) AS №,
```

```
d.name DriverName, d.surname DriverSurname, dr.rating DriverRating, t.model TaxiModel
```




```
from driver d
```

```
left join DriverRating dr on d.DriverRatingID = dr.id
```

```
left join Taxi t on d.TaxiID = t.id;
```

3 • `select * from DriverInfo;`

<

Result Grid   Filter Rows:  Export:  Wrap

	№	DriverName	DriverSurname	DriverRating	TaxiModel
▶	1	Cathleen	Curthoys	99.7	Mitsubishi
	2	Napoleon	Andreuzzi	99.3	Subaru
	3	Baudoin	Simyson	98.5	Pontiac
	4	Efrem	Southwood	98.4	Jaguar
	5	Boycey	Glyde	97.8	Pontiac
	6	Leupold	Guiness	96.5	Merkur
	7	Nikolos	Exter	95.9	Lexus
	8	Malvin	Mockford	95.6	Mitsubishi
	9	Knox	Trevithick	92.3	Volkswagen
	10	Orlan	Mattson	91.5	Subaru
	11	Elke	Fitzsimon	89.8	Volkswagen
	12	Sarena	Mandrake	89.4	Dodge
	13	Aubree	Egerton	88.2	Toyota
	14	Stephannie	Peeke	86.9	Lexus
	15	Gay	Coppock.	86.1	Chevrolet
	16	Ingaberg	Lipyeat	84.9	Chevrolet
	17	Maynord	Pridmore	84.8	BMW
	18	Derby	Breed	84.6	Ford
	19	Ashlee	Broscombe	84	Chevrolet

Рисунок 7.4.1 результат представления DriverInfo

```
CREATE VIEW RidesAboveAveragePrice AS
```

```
SELECT ROW_NUMBER() OVER (ORDER BY Amount desc) AS №, p.Amount, p.Time,  
c.id ClientID
```

```
FROM payment p
```

```
left join ordertable on p.id = ordertable.PaymentID
```

```
left join Client c on ordertable.ClientID = c.id
```

```
WHERE Amount > (SELECT (AVG(Amount) * 1.5) FROM payment);
```

3 • `select * from RidesAboveAveragePrice;`

Result Grid     Filter Rows: <input type="text"/>   Export:				
	Nº	Amount	Time	ClientID
▶	1	9976.9	11:31 AM	383
	2	9963.2	12:45 AM	606
	3	9957.6	12:35 AM	678
	4	9949		308
	5	9946.8	4:23 PM	805
	6	9945.9		420
	7	9931.8	12:55 AM	808
	8	9927.9	3:33 PM	683
	9	9921.8	11:34 AM	881
	10	9912.4		898
	11	9906.1		520
	12	9905.3		720
	13	9897.9		359
	14	9886.3		513
	15	9885		800
	16	9884.6	6:39 PM	700
	17	9879.3		545
	18	9875.3		776
	19	9871.1	8:06 AM	230

RidesAboveAveragePrice 19 ×

Рисунок 7.4.2 результат представления RidesAboveAveragePrice

```
create view MostSimilarByRating as
select concat(d.Name, ' ', d.surname) Driver, concat(c.Name, ' ', c.surname) Client,
ROUND(abs(cr.rating - dr.rating),2) as DifferencesInRating
from client c
left join ordertable o on c.id = o.ClientID
left join clientrating cr on c.clientratingID = cr.id
right join driver d on o.DriverID = d.id
left join driverrating dr on d.DriverRatingID = dr.id
order by ROUND(abs(cr.rating - dr.rating),2);
```



2

3 • `select * from MostSimilarByRting;`

Driver	Client	DifferencesInRating
Vidovik Gregh	Corny Wiffill	0
Cassie Langcaster	Dian Kleinfeld	0.1
Keriann Ruppeli	Darnell Sigsworth	0.2
Gordie Fairall	Una Cameli	0.2
Lynne Couchman	Mordecai D'eath	0.3
Lynne Couchman	Artur Pendleton	0.3
Collen Bestall	Lira Cornelle	0.5
Timmy Peet	Camile Grossman	0.6
Ingaberg Lipyeat	Elvera Eaklee	0.7
Justina Crumpton	Sumner Bridewell	0.7
Lia Barabisch	Brendin Goulder	0.7
Stephannie Peeke	Jerrylee McLachlan	0.7
Bondie Attenburrow	Rodney Wycliffe	0.8
Justina Crumpton	Jason Malek	0.9
Vannie Curteis	Mercie Rove	1
Vidovik Gregh	Kenyon Shinefield	1
Aubree Egerton	Cammy Treven	1
Paulo Boog	Catina Durrand	1.1
Chrysa Kohter	Paige Ilbert	1.1

Рисунок 7.4.3 результат представлення MostSimilarByRting

## 7.5.SQL-запити та результати виконання

На рисунках 5.1.1 – 5.1.20 зображені результати до відповідних запитів  
`use newsmt;`

-- 1)

`SELECT * FROM Ride WHERE Status = 'true';`

id	startTime	FinishTime	Status
2	11:45 AM	1:08 PM	true
4	5:51 AM	4:43 PM	true
5	2:41 AM	5:04 PM	true
7	7:59 AM	12:33 PM	true
9	5:53 AM	7:33 PM	true
10	11:36 AM	3:35 PM	true
11	1:25 AM	4:43 PM	true
12	4:12 AM	6:34 PM	true
17	4:43 AM	1:02 PM	true
20	5:39 AM	8:30 PM	true
22	12:52 AM	1:16 PM	true
24	1:13 AM	7:59 PM	true

Рисунок 5.1.1

-- 2)

`select count(id) from ride where status = 'true';`

	count(id)
►	507

Рисунок 5.1.2

-- 3)

```
SELECT c.Name, c.PhoneNumber FROM Client c WHERE id NOT IN (SELECT id FROM ClientRating
WHERE Rating < 80);
```

	Name	PhoneNumber
►	Hana	479-989-1515
	Mohammed	550-106-0031
	Aarika	848-957-7137
	Cynthia	230-939-7701
	Genevieve	917-876-7334
	Georgette	344-973-5363
	Raffarty	444-865-4599
	Felita	310-323-4947
	Naila	060-057-6301

Рисунок 5.1.3

-- 4)

```
SELECT SUM(Amount) AS DailyEarning FROM Payment WHERE Status = 'true';
```

	DailyEarning
►	2623034.9999999999

Рисунок 5.1.4

-- 5)

```
SELECT d.Name, d.Surname, dr.Rating FROM Driver d LEFT JOIN DriverRating dr ON d.DriverRatingID
= dr.id WHERE dr.Rating between 70 and 90;
```

	Name	Surname	Rating
►	Brendan	Lashmar	70.2
	Ashlee	Broscombe	84
	Chrisse	Cote	77.1
	Cirillo	Fenkel	77.6
	Joela	Garmons	75.8
	Stephannie	Peeke	86.9
	Chase	Coogan	73.3
	Aubree	Egerton	88.2
	Hanni	Thurner	77.1
	Vidovik	Gregg	78.9
	Scottie	Wincom	74.7
	Sarena	Mandrake	89.4
	Gay	Coppock	86.1
	Jasper	Guisler	72.4
	Maria	Kurdani	83.3
	Maynard	Pridmore	84.8
	Engelbert	Litchmore	79.1
	Vannie	Curteis	76.4
	Gerrard	Paoletto	73.7
	Derby	Breed	84.6
	Ingaberg	Lipyeat	84.9
	Peyter	Fawson	76.9
	Nikolia	Farris	73.3
	Elke	Fitzsimon	89.8
	Aggie	Eisold	76.2

Рисунок 5.1.5

-- 6)

```
SELECT AVG(Rating) AS AverageDriverRating FROM DriverRating;
```

AverageDriverRating
53.33000000000001

Рисунок 5.1.6

-- 7)

```
SELECT StartAddress, FinishAddress, Distance FROM Route ORDER BY Distance DESC LIMIT 1;
```

StartAddress	FinishAddress	Distance
9958 Darwin Terrace	6 Heffernan Parkway	84.9

Рисунок 5.1.7

-- 8)

```
SELECT COUNT(*) AS BusyTaxis, (select count(*) from taxi WHERE AvailabilityStatus ='true') as FreeTaxis FROM Taxi WHERE AvailabilityStatus ='false';
```

BusyTaxis	FreeTaxis
52	48

Рисунок 5.1.8

-- 9)

```
SELECT c.Name , c.Surname, c.Address, c.PhoneNumber, cr.PercentCanceledTrips FROM Client c left JOIN ClientRating cr ON c.ClientRatingID = cr.id WHERE cr.PercentCanceledTrips > 60;
```

Name	Surname	Address	PhoneNumber	PercentCanceledTrips
Kalle	Perche	33 Bunting Drive	303-227-0825	89
Sabrina	Pickering	173 Continental Drive	520-828-8287	90.1
Helga	Delgardillo	1691 Ridgeway Pass	904-533-9678	88.8
Daryl	Snasdell	0 Bultman Avenue	678-173-6538	98.5
Arney	Skoggings	60 Oak Valley Trail	560-265-9383	81.7
Almeta	Watkin	94709 Continental Parkway	976-114-0464	94.7
Lane	Corradino	67 Mandrake Lane	314-284-3228	82.1
Nevsa	McKitterick	85198 Manley Terrace	215-697-3580	86.4
Vito	Wilce	95 Sherman Lane	638-232-2265	61.4
Andie	Whimp	6750 Kipling Plaza	682-843-1490	63.7
Alabanza	Wolkovich	75 Gelfing Drive	828-308-0034	83.7

Рисунок 5.1.9

-- 10)

```
SELECT d.* FROM Driver d LEFT JOIN OrderTable o ON d.id = o.DriverID WHERE o.Datetime BETWEEN '2023-12-01' and '2023-12-31';
```

	id	Name	Surname	PhoneNumber	LicenseNumber	DriverRatingID	TaxiID
▶	35	Dona	Bloor	211-860-8619	7985086935	66	85
	45	Gay	Coppock	426-944-5475	3026115436	56	95
	34	Engelbert	Litchmore	355-343-5300	7002322656	67	84
	64	Rhianon	Randle	758-439-2632	9078216875	37	36
	92	Ashlee	Broscombe	898-586-4579	9814085316	9	8
	43	Shane	Janus	517-159-8718	6292190722	58	93
	43	Shane	Janus	517-159-8718	6292190722	58	93
	57	Jasen	Whittenbury	189-920-2987	7535035299	44	43
	41	Maria	Kurdani	772-555-3914	6227461075	60	91
	79	Roselia	Taree	812-297-3794	9255782983	22	21
	37	Efrem	Southwood	489-124-8139	5740821061	64	87
	29	Gerrard	Paoletto	801-345-3242	7225361813	72	79
	78	Nikolos	Exter	107-498-9143	3338228503	23	22
	47	Chrysa	Kohter	658-520-7390	6147367889	54	97
	20	Orlan	Mattson	614-812-6769	913216933	81	70
	38	Kerian	Ruppell	742-790-0051	7471610301	63	88
	47	Chrysa	Kohter	658-520-7390	6147367889	54	97
	30	Rheba	Wloch	193-201-4132	3139939418	71	80
	94	Broddy	Kitchingham	288-470-6971	4402151252	7	6
	63	Vidovik	Gregh	728-193-7280	9496272924	38	37
	3	Aggie	Eisold	754-796-0454	2455272982	98	53
	65	Cassandry	Brandsen	912-608-3297	5256090232	36	35

Рисунок 5.1.10

-- 11)

```
SELECT b.* FROM Bonuses b JOIN Client c ON b.ClientID = c.id WHERE c.id = 1;
```

	id	Points	DiscountPercent	id	Name	Surname	Address	PhoneNumber	Email	BonusesID	RideID	Client
▶	250	900	30.6	1	Sunshine	Snalom	413 Garrison Crossing	596-261-1910	ssnalom0@pinterest.com	250	250	1000

Рисунок 5.1.11

-- 12)

```
select count(*) CountNonPaidTrip from payment where status = 'false';
```

	CountNonPaidTrip
▶	501

Рисунок 5.1.12

-- 13)

```
select count(o.id) OrderCount, d.name DriverName, d.surname DriverSurname, d.id DriverId
from ordertable o
right join driver d on o.driverID = d.id
group by d.name, d.surname, d.id
having OrderCount > 15;
```

	OrderCount	DriverName	DriverSurname	DriverId
▶	20	Boycey	Glyde	15
	16	Moll	Soffe	51
	16	Bonnie	Redford	55
	16	Lynne	Couchman	72

Рисунок 5.1.13

-- 14)

```
select c.Name, c.Surname, b.DiscountPercent, cr.Rating ClientRating from client c
left join bonuses b on c.BonusesID = b.id
left join clientRating cr on c.ClientRatingID = cr.id
```

order by b.DiscountPercent desc, ClientRating

limit 3;

	Name	Surname	DiscountPercent	ClientRating
►	Sibyl	Denman	98.7	33.9
	Arden	Ibell	98.7	96.6
	Kissee	Camoletto	98.5	15.4

Рисунок 5.1.14

-- 15)

select c.name, c.surname, p.amount, p.time

from client c

left join ordertable o on c.id = o.ClientID

left join payment p on o.PaymentID = p.id

where p.amount > 5000 or p.amount = 781.8;

	name	surname	amount	time
►	Romona	Bruggen	781.8	2:33 PM
	Dare	Ottery	7687.8	9:34 AM
	Johnna	Plank	6576.5	2:54 PM
	Neille	Rentell	7594.7	10:30 AM
	Felita	Collabine	8175	
	Bethina	Dericut	7106.3	
	Elsy	Milland	8611.3	
	Kelsey	Kelsey	6366.9	10:13 PM
	Marina	Vickerstaff	5370.9	3:11 AM
	Anica	Hurle	6370.5	
	Timothea	Szymanski	7325.5	
	Ulberto	Rhyme	5855.2	2:30 AM
	Alick	MacGrath	7282.8	7:46 AM
	Jennifer	Verlander	9696.8	
	Revkah	Mylchreest	6589.5	1:59 AM
	Craggie	Summersett	8954.6	
	Alameda	Apdell	5815.7	11:52 AM
	Sindare	Scully	6644	
	Marybeth	Summerson	5805.8	
	Greg	Ventom	8299.8	
	Mattie	Stowell	9149.6	1:15 AM
	Honorio	Elsbury	7620.2	12:46 AM
	Walter	Thompson	6477.6	

Рисунок 5.1.15

-- 16)

select count(o.id) OrderCount, d.name DriverName, d.surname DriverSurname, d.id DriverId, row\_number()  
over(ORDER BY count(o.id) desc) as BestDriver

from ordertable o

right join driver d on o.driverID = d.id

group by d.name, d.surname, d.id

having OrderCount > 15;

	OrderCount	DriverName	DriverSurname	DriverId	BestDriver
►	20	Boycey	Glyde	15	1
	16	Bonnie	Redford	55	2
	16	Moll	Soffe	51	3
	16	Lynne	Couchman	72	4

Рисунок 5.1.16

-- 17)

select id, taxinumber

from taxi

where model like 'C%' or AvailabilityStatus = true;

	id	taxinumber
▶	8	1557404755
	14	2475564741
	17	1092504885
	49	4751502301
	71	6739859000
	74	2581661623
	76	398874352
	80	9858155964
	89	2393637249
	93	6496296359
	94	7208986231
	95	8451244394

Рисунок 5.1.17

-- 18)

select id, taxinumber

from taxi

where model not like 'C%' and AvailabilityStatus = false ;

	id	taxinumber
▶	1	2964944091
	2	7849798635
	3	4719870678
	4	4464530974
	5	2212077211
	6	1294611348
	7	5872541910
	9	2066460702
	10	8130986434
	11	1122316933
	12	6781775367
	13	5636696116
	15	7382716422
	16	7982436854

Рисунок 5.1.18

-- 19)

select \* from client

where name not in('Sybil', 'Sunshine', 'Bethanne');

	id	Name	Surname	Address	PhoneNumber	Email	BonusesID	RideID	ClientRatingID
▶	4	Raymund	Franken	6089 Thierer Road	328-376-2609	rfranken3@amazon.co.uk	253	247	997
	5	Hana	Jent	71 Brown Plaza	479-989-1515	hjent4@xinhuanet.com	254	246	996
	6	Skyler	Neads	1867 Buhler Lane	929-321-3628	sneads5@theatlantic.com	255	245	995
	7	Gisella	Vesty	6990 Holy Cross Drive	945-193-4074	gvesty6@webnode.com	256	244	994
	8	Lissy	McBrady	9266 Brickson Park Place	721-924-4268	lmcbrady7@wix.com	257	243	993
	9	Adolphe	Ondrak	0541 Gina Street	621-625-6004	aondrak8@earthlink.net	258	242	992
	10	Dael	Paterno	38 Bartillon Court	151-586-2476	dpaterno9@nih.gov	259	241	991
	11	Alice	Daly	2930 Heffernan Park	288-373-6156	adalya@bing.com	260	240	990
	12	Conan	Fallis	886 Petterle Lane	535-928-2031	cfallisb@sakura.ne.jp	261	239	989

Рисунок 5.1.19

-- 20)

select \* from client where

PhoneNumber like '\_2%9' or address like '%Gina%';

	id	Name	Surname	Address	PhoneNumber	Email	BonusesID	RideID	ClientRatingID
►	4	Raymund	Franken	6089 Thierer Road	328-376-2609	rfranken3@amazon.co.uk	253	247	997
	9	Adolphe	Ondrak	0541 Gina Street	621-625-6004	aondrak8@earthlink.net	258	242	992
	116	Alleyn	Backshaw	666 Melvin Circle	520-326-2029	abackshaw37@bbb.org	365	135	885
	225	Mollie	Burgett	6 Quincy Drive	128-656-7589	mburgett68@goo.gl	474	26	776
	232	Isabel	Harman	40135 Grayhawk Terrace	724-646-8789	iharman6f@soundcloud.com	481	19	769
	291	Zorine	Klimas	7 Kensington Road	621-342-1319	zklimas82@nytimes.com	41	460	710
	309	Correy	Lewsey	077 Farmco Plaza	823-151-8639	clewsey8k@trellian.com	59	442	692
	549	Carlye	Ion	2 Dawn Crossing	627-262-8949	clonf8@sakura.ne.jp	49	952	452
	703	Mill	Woodbridge	01 Carpenter Pass	523-565-6079	mwoodbridgej@accuweather.com	203	798	298
	711	Felic	McKelvey	1253 Corben Center	224-671-9109	fmckelveyjq@linkedin.com	211	790	290
	734	Homere	Lambert-Ci...	03082 Summerview Center	226-870-5989	hlambertciornykd@tmall.com	234	767	267
	738	Bobine	Lachaize	2 Superior Pass	425-571-7689	blachaizekh@purevolume.com	238	763	263
	895	Florina	Cleyburn	157 Lyons Street	822-342-2739	fcleyburnou@unblog.fr	395	606	106

Рисунок 5.1.20



## ВИСНОВКИ

У даній курсовій роботі досліджено історію та еволюцію таксі як виду громадського транспорту, а також визначено актуальні проблеми та перспективи розвитку сучасних служб таксі. В розділі "Вступ" розглянуті основні етапи становлення таксі, включаючи початки в Лондоні та подальший розвиток, зокрема в колишньому СРСР. Зазначено, що таксі в сучасному житті виконує різноманітні функції та має велике значення для громадськості.

У розділі "Постановка завдання" визначено мету та завдання інформаційної системи, спрямованої на підвищення якості обслуговування клієнтів служби таксі. Зазначено, що для досягнення цієї мети необхідно вирішити ряд завдань, таких як побудова ег-моделі, набір сутностей та атрибутів, визначення зв'язків між сутностями, розробка реляційної моделі бази даних, реалізація бази даних та інші.

Важливим етапом в розвитку сучасних служб таксі є впровадження інформаційних технологій, зокрема мобільних додатків для замовлення послуг. Такі технології сприяють автоматизації процесів, полегшують вибір оптимального водія для клієнта та підвищують рівень обслуговування.

У висновку варто відзначити, що подальший розвиток і вдосконалення баз даних та інформаційних систем таксі може сприяти покращенню конкурентоспроможності компаній, їхньому пристосуванню до сучасних вимог ринку та забезпечити більш ефективне задоволення потреб клієнтів.



## Використані джерела

1.

<https://uklon.com.ua/>

2.

<https://www.uber.com/ua/uk/>

3.

<https://uklon.com.ua/>

4.

<https://uk.wikipedia.org/wiki/%D0%A2%D0%B0%D0%BA%D1%81%D1%96>

5.

<https://chat.openai.com/>

Посилання на репозиторій GitHub де всі файли з даними які були завантажені в базу даних.

<https://github.com/slavikDe/cursova-bd>