

Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

# Лабораторна робота №1

***з дисципліни «Бази даних»***

**«Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL»**

Виконала студентка групи: КВ-33

ПІБ: Щербатюк Є. О.

Перевірив: Павловський В. І.

**Київ 2025**

**Постановка задачі:**

1. Розробити модель «сутність-зв’язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

***Завдання №1:***

**Опис предметної галузі:**

Тема моєї бази даних: Система обліку автомобільного парку компанії. Компанія займається вантажними перевезеннями.

**При створенні даної бази даних я виділила такі сутності:**

1. **Автомобіль (Car)** – представляє транспортні засоби компанії, які використовуються для вантажних перевезень:

* Атрибути: VIN(унікальний номер кузова), реєстраційний номер автомобіля, марка, вантажопідйомність.

1. **Водій (Driver)** – представляє працівників, які мають водійське посвідчення та керують транспортним засобом.

* Атрибути: номер водійського посвідчення, ім’я, прізвище, категорія.

1. **Маршрут (Route) –** ця сутність описує шлях перевезення вантажу:

* Атрибути: пункт відправлення, пункт призначення, відстань.

1. **Клієнт (Customer)** – ця сутність представляє собою компанію, яка замовляє вантажні перевезення:
   * Атрибути: назва компанії, телефон, email.
2. **Рейс (Trip) –** ця сутність відображає факт перевезення вантажу.
   * Атрибути: дата виїзду, дата прибуття, дата повернення, опис вантажу, вага вантажу.

**Зв’язки:**

Зв’язок «Автомобіль» - «Рейс»:

* Тип зв'язку: 1 до N (один автомобіль може здійснювати багато рейсів; рейс виконується тільки одним автомобілем).

Зв’язок «Водій» - «Рейс»:

* Тип зв'язку: 1 до N (один водій може здійснювати багато рейсів; рейс має конкретного відповідального водія).

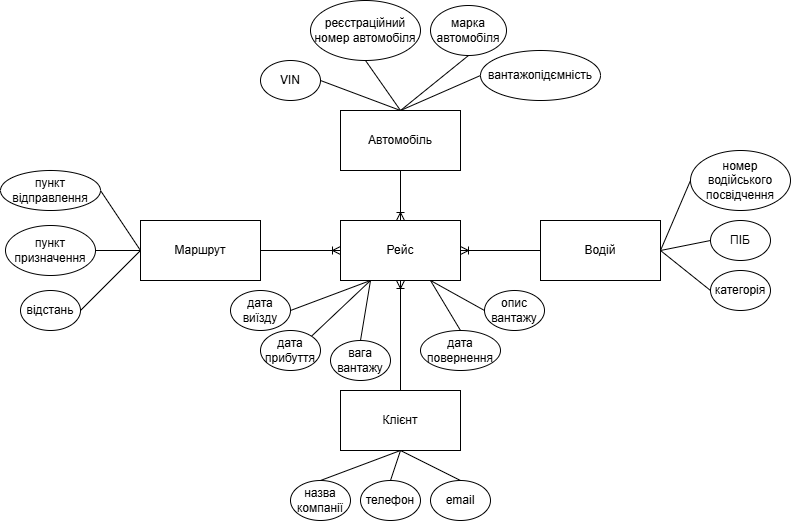
Зв’язок «Маршрут» - «Рейс»:

* Тип зв'язку: 1 до N (один маршрут може використовуватись в багатьох рейсах).

Зв’язок «Клієнт» - «Рейс»:

* Тип зв'язку: 1 до N (один клієнт може замовити багато рейсів).

Графічне подання концептуальної моделі «Сутність-зв’язок» зображено на рисунку 1



*Рисунок 1 – ER-діаграма, побудована за нотацією” Пташина лапка”*

***Перетворення концептуальної моделі у логічну модель та схему бази даних***

Сутність "Автомобіль" перетворено в таблицю Car з первинним ключем car\_id та атрибутами vin, license\_plate, brand, load\_capacity.

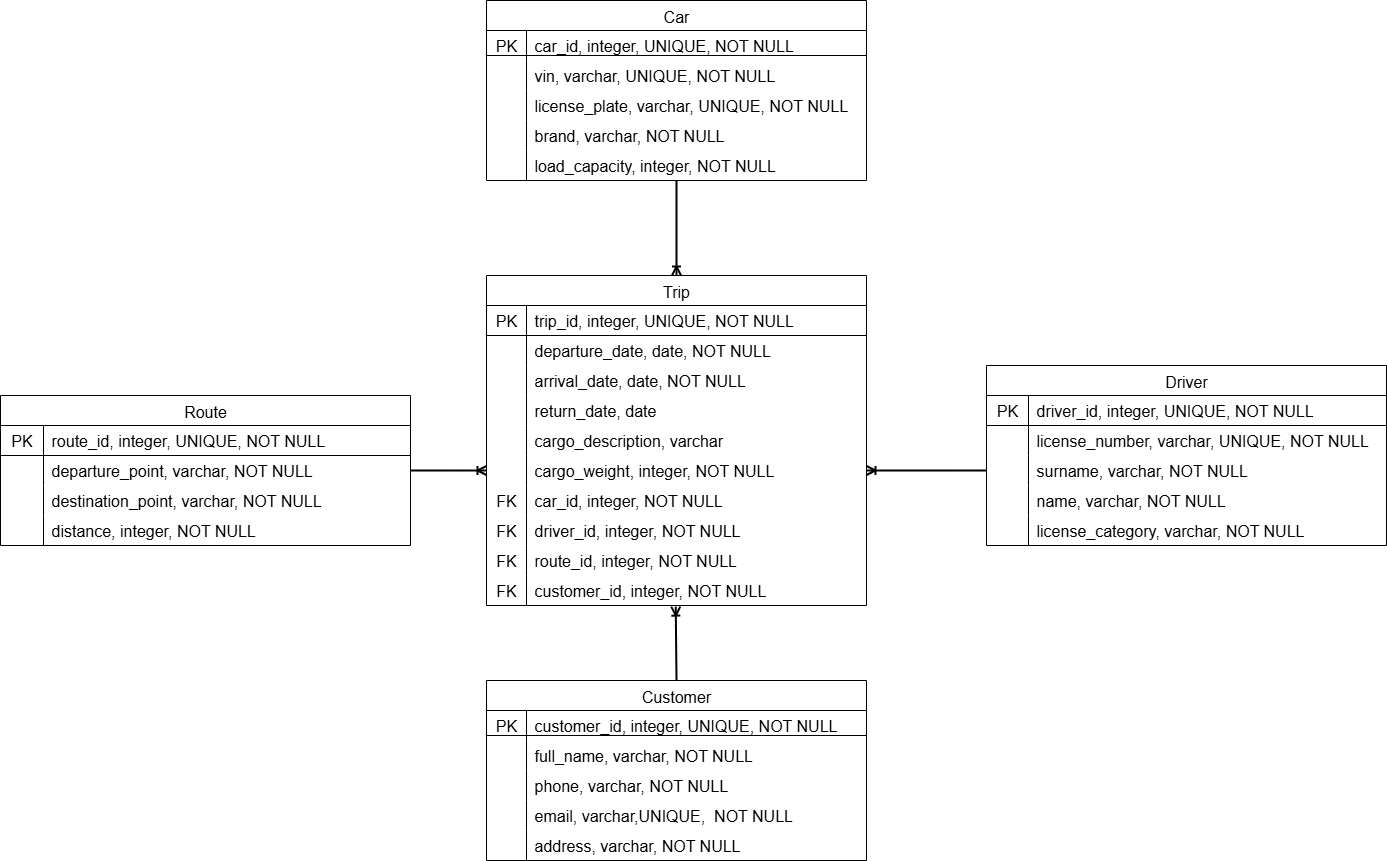
Сутність "Водій" перетворено в таблицю Driver з первинним ключем driver\_ id та атрибутами license\_number, name, surname, license\_category.

Сутність "Маршрут" перетворено в таблицю Route з первинним ключем route\_id та атрибутами departure\_point, destination\_point, distance.

Сутність "Клієнт" перетворено в таблицю Customer з первинним ключем customer\_id та атрибутами full\_name, phone, email.

Сутність “Рейс” перетворено в таблицю Trip з первинним ключем trip\_id та атрибутами departure\_date, arrival\_date, return\_date, cargo\_description, cargo\_weight. У таблиці є зовнішні ключі car\_id, driver\_id, route\_id, customer\_id, які пов’язані з таблицями Car, Driver, Route та Customer відповідно.

Графічне подання логічної моделі «Сутність-зв’язок» зображено на рисунку 2



*Рисунок 2 – Схема бази даних*

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої.

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| *Car* – містить дані про автомобілі компанії | *car\_id* - унікальний ідентифікатор автомобіля | integer (числовий) |
| vin – унікальний VIN автомобіля | character varying (рядок) |
| license\_plate – унікальний номер автомобіля | character varying (рядок) |
| *brand* – марка автомобіля | character varying (рядок) |
| *load\_capacity –* вантажопідйомність автомобіля | integer (числовий) |
| *Driver* - містить дані про водіїв компанії | *driver\_id* - унікальний ідентифікатор водія | integer (числовий) |
| *license\_number* – унікальний номер водійського посвідчення | character varying (рядок) |
| *surname* – прізвище водія | character varying (рядок) |
| *name – ім’я водія* | character varying (рядок) |
| license\_category – категорія водійського посвідчення | character varying (рядок) |
| Route - містить дані про маршрут | *route\_id* – унікальний ідентифікатор маршруту | integer (числовий) |
| departure\_point – пункт відправлення | character varying (рядок) |
| destination\_point – пункт прибуття | character varying (рядок) |
| *distance* – відстань маршруту | integer (числовий) |
| Customer – містить дані про клієнта | *customer\_id* - унікальний ідентифікатор клієнта | integer (числовий) |
| *full\_name* – назва компанії клієнта | character varying (рядок) |
| *phone* – номер телефону компанії клієнта | character varying (рядок) |
| *email* – електронна адреса компанії клієнта | character varying (рядок) |
| *address* – адреса компанії клієнта | character varying (рядок) |
| *Trip -* містить інформацію про рейс | *trip\_id* - унікальний ідентифікатор рейсу | integer (числовий) |
|  | *departure\_date* – дата відправлення | timestamp (дата та час) |
|  | *arrival\_date – дата прибуття* | timestamp (дата та час) |
|  | *return\_date – дата повернення водія* | timestamp (дата та час) |
|  | *cargo\_description – опис вантажу* | character varying (рядок) |
|  | *cargo\_weight – вага вантажу* | integer (числовий) |
|  | *car\_id – ідентефікатор автомобіля* | integer (зовнішній ключ до Автомобіля) |
|  | *driver\_id – ідентифікатор водія* | integer (зовнішній ключ до Водія) |
|  | *route\_id – ідентифікатор маршруту* | integer (зовнішній ключ до Маршруту) |
|  | *customer\_id – ідентифікатор клієнта* | integer (зовнішній ключ до Клієнта) |

*Таблиця 1 – Опис об’єктів бази даних*

***Функціональні залежності для таблиць***

1. *Car:*

*car\_id → vin, license\_plate, brand, load\_capacity*

*vin → id, license\_plate, brand, load\_capacity*

*license\_plate → id, vin, brand, load\_capacity*

1. *Driver*

*driver\_id → license\_number, surname, name, license\_category*

*license\_number → id, surname, name, license\_category*

1. *Route:*

*route\_id → departure\_point, destination\_point, distance\_km*

1. *Customer:*

*id → full\_name, phone, email, address*

*email → id, full\_name, phone, address*

1. *Trip:*

*id → departure\_date, arrival\_date, return\_date, cargo\_description, cargo\_weight, car\_id, driver\_id, route\_id, customer\_id*

Ці функціональні залежності вказують на те, які атрибути в кожній таблиці визначаються від інших атрибутів у моделі бази даних. Це важливо для забезпечення нормалізації та ефективного управління базою даних, оскільки правильне визначення функціональних залежностей дозволяє уникнути надмірності даних і забезпечує їхню цілісність.

Транзитивні функціональні залежності виникають тоді, коли один атрибут визначає інший через третій атрибут. Іншими словами, якщо атрибут A визначає атрибут B, а атрибут B визначає атрибут C, то ми можемо стверджувати, що атрибут A транзитивно визначає атрибут C. Однак у даній моделі транзитивних функціональних залежностей не виявлено, що вказує на відсутність надмірності даних і сприяє ефективному функціонуванню бази даних.

**Завдання №3**

**Перевірка нормальних форм:**

* 1НФ: усі атрибути атомарні, повторюваних груп немає.  
  Приклад: у Car зберігається тільки один VIN, одна марка, один номер.
* 2НФ: усі таблиці мають простий первинний ключ (одне поле id), тому немає часткових залежностей.  
  Приклад: у Trip всі неключові атрибути залежать від id.
* 3НФ: транзитивних залежностей немає.  
  Приклад: телефон клієнта залежить від Customer.id, а не від інших атрибутів; відстань маршруту залежить від Route.id, а не через інші атрибути.

**Висновок: схема відповідає вимогам 3НФ, зміни не потрібні.**

Схема бази даних у pgAdmin 4 зображено на рисунку 3.

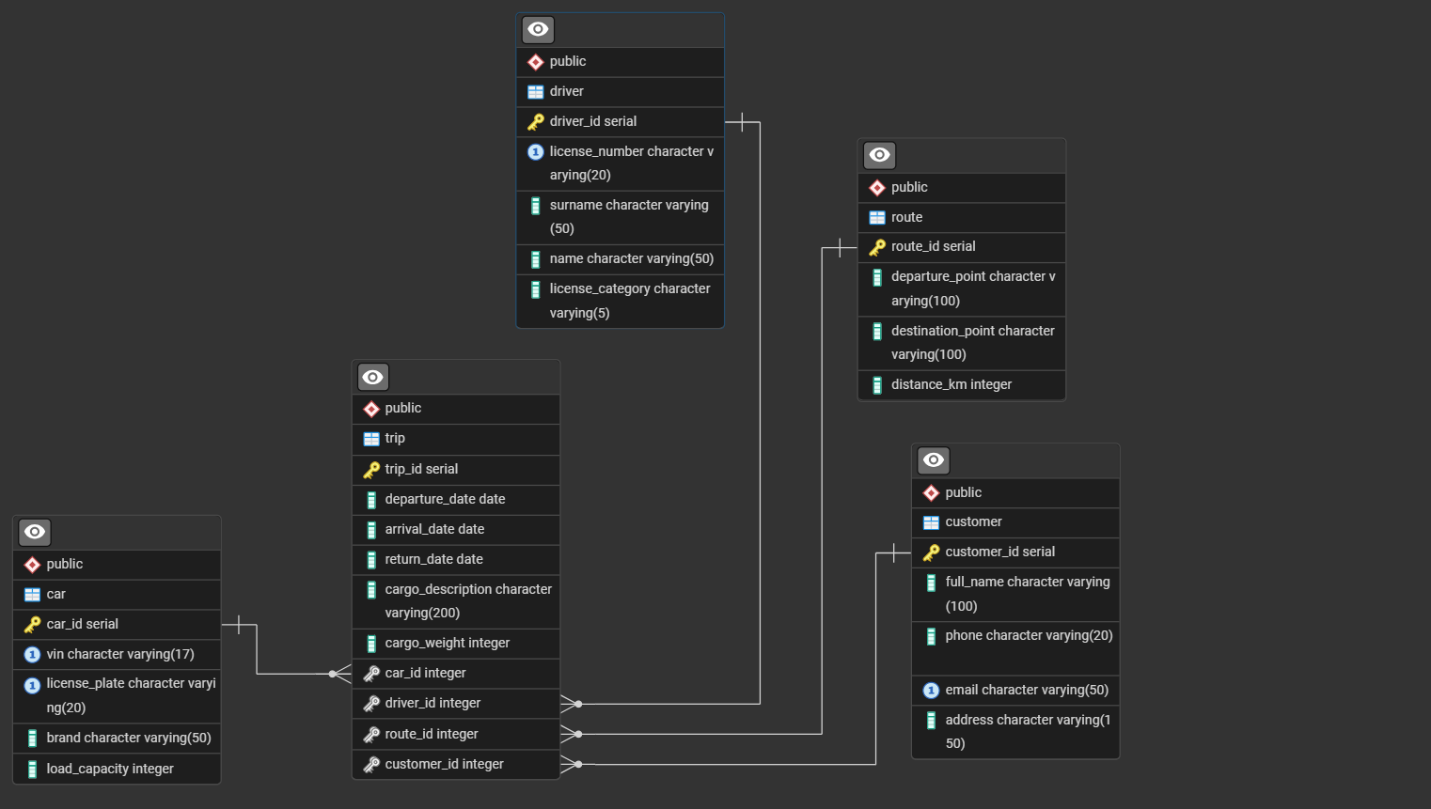
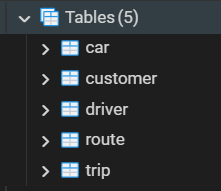
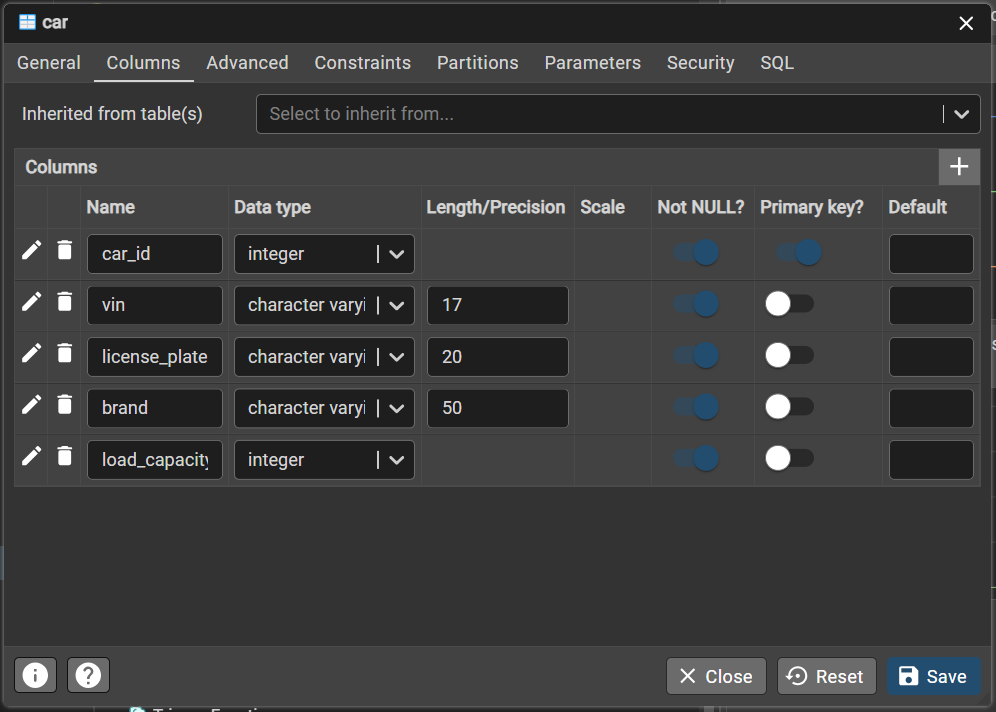
****

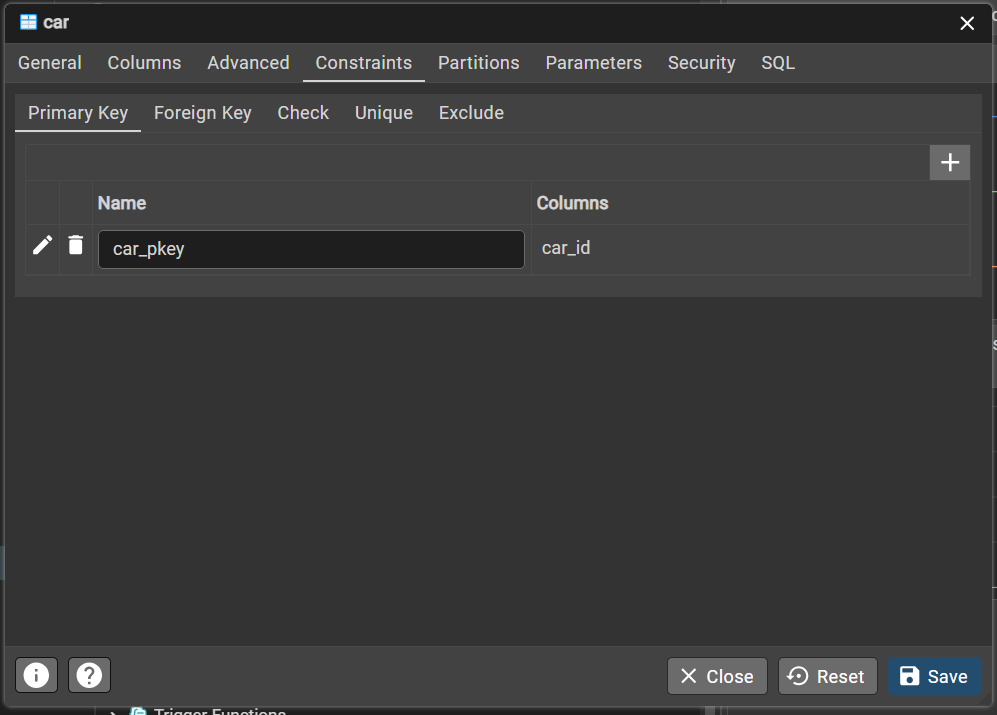
Рисунок 3 - Схема бази даних у pgAdmin 4

**Таблиці бази даних у pgAdmin4**



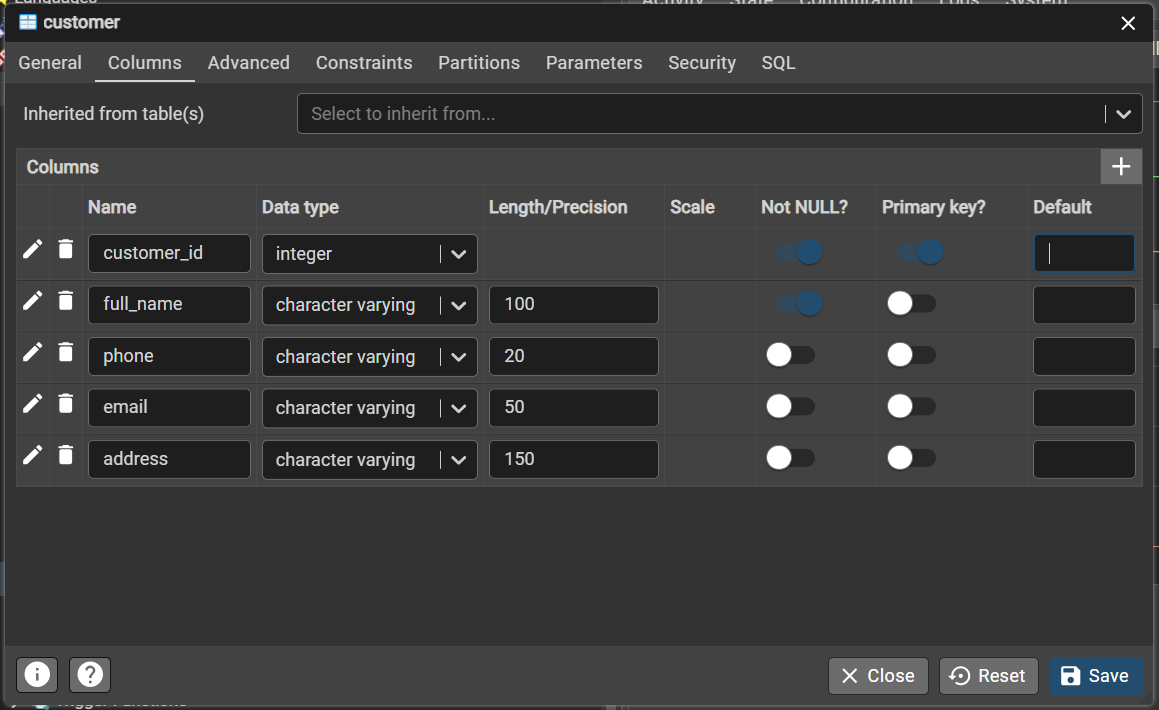
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Car** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

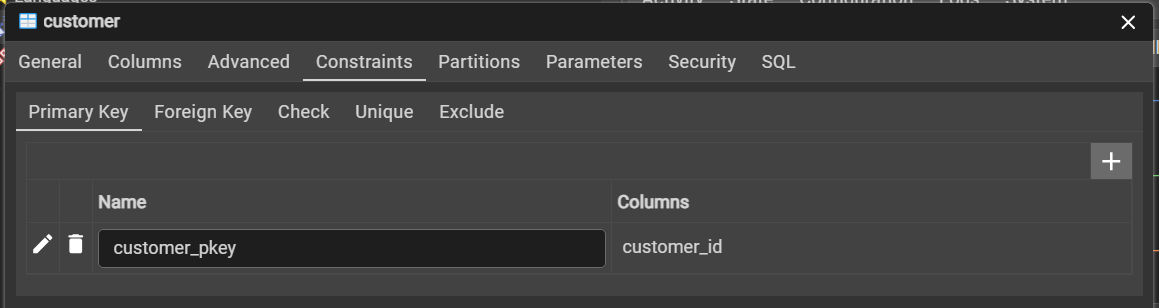




У цієї таблиці немає зовнішніх ключів (FK)

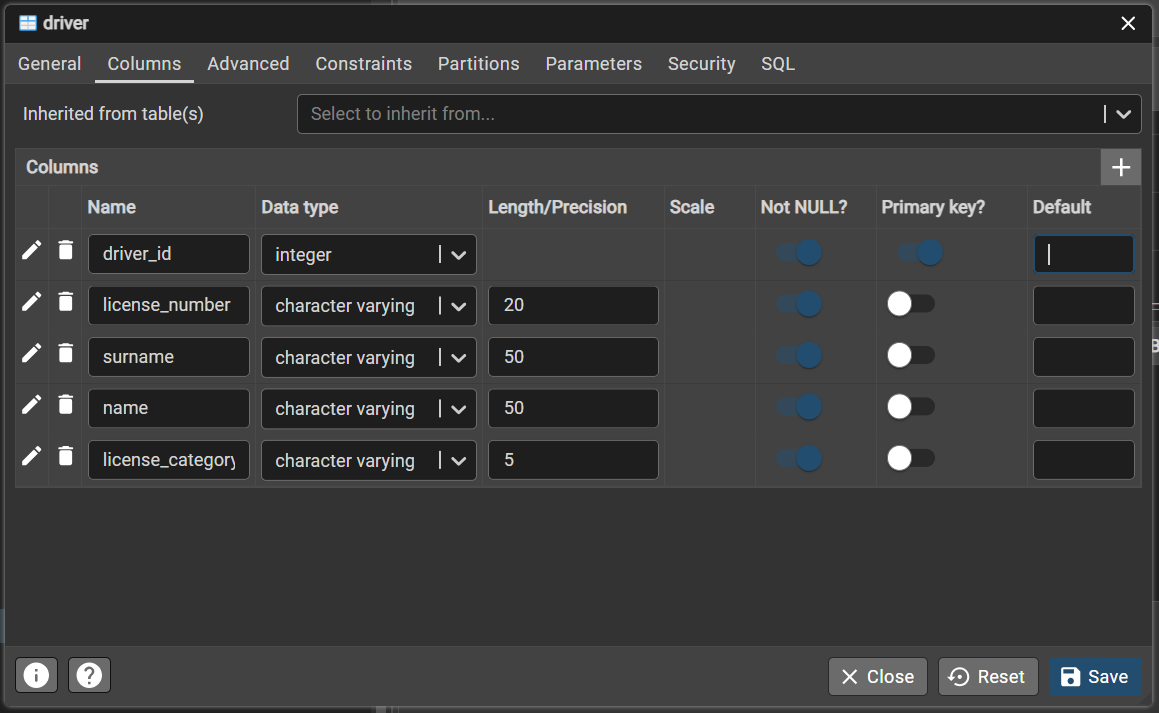
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Customer**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

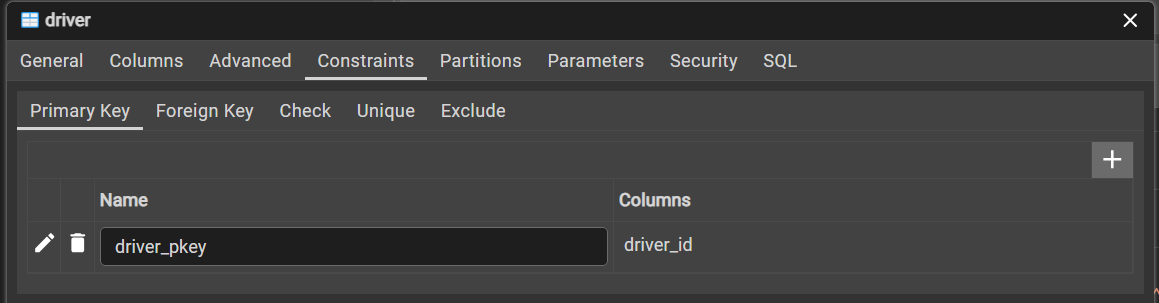




У цієї таблиці немає зовнішніх ключів (FK)

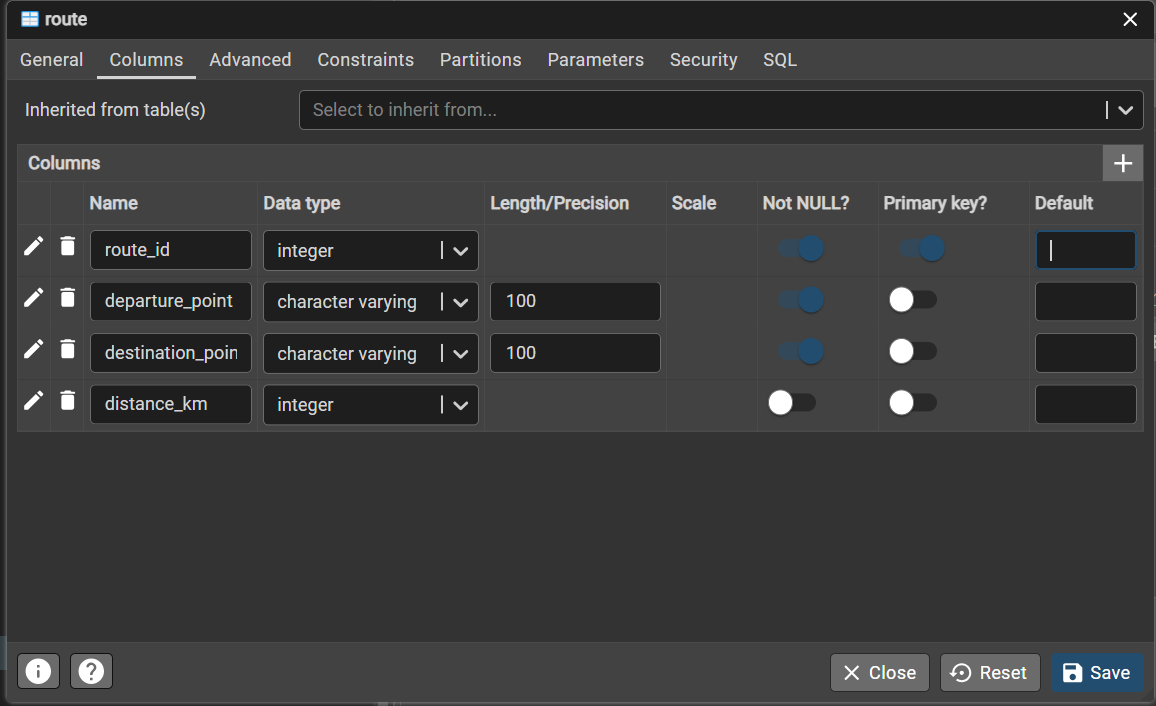
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Driver**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

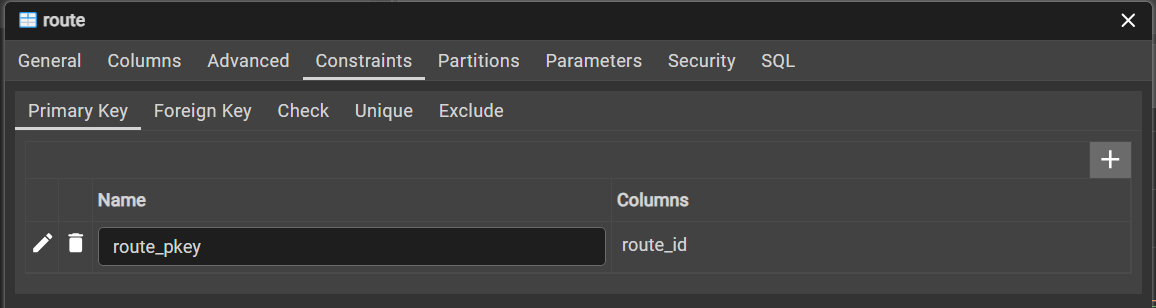




У цієї таблиці немає зовнішніх ключів (FK)

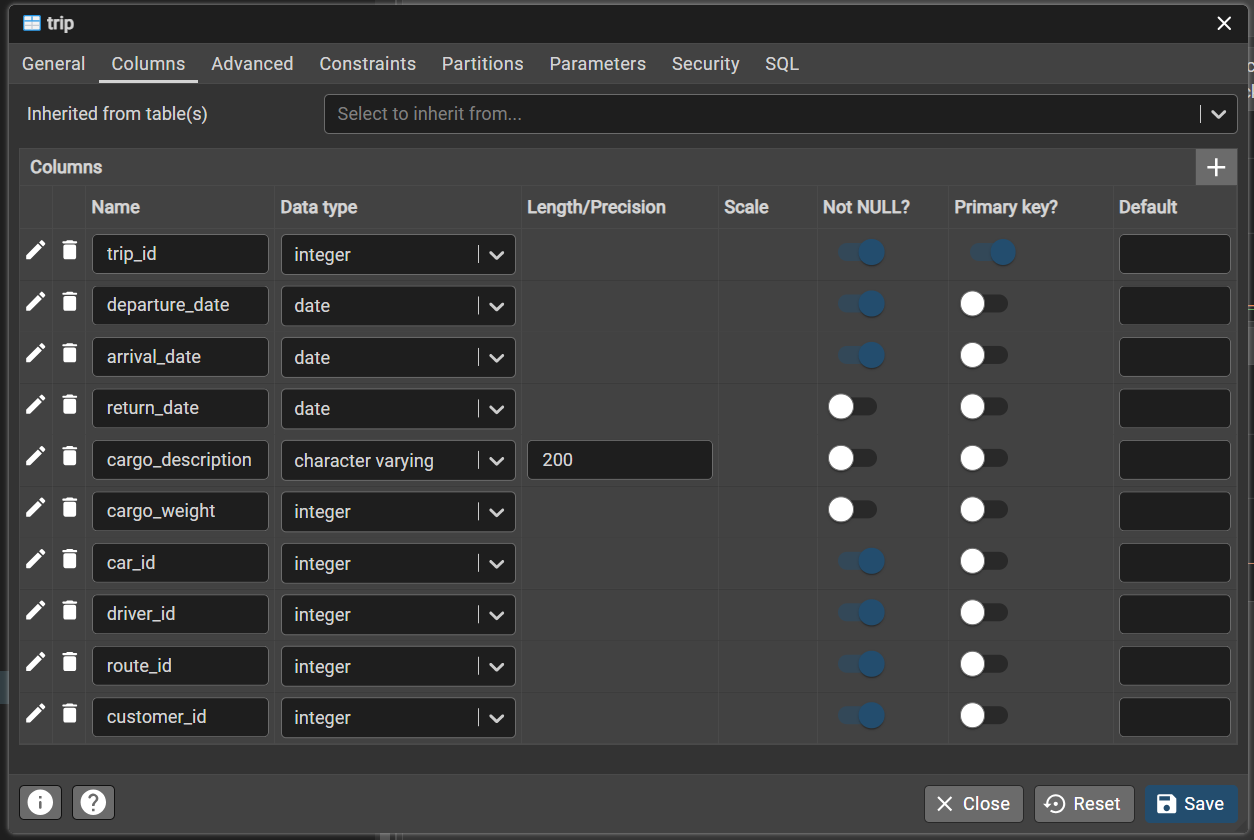
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Route**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

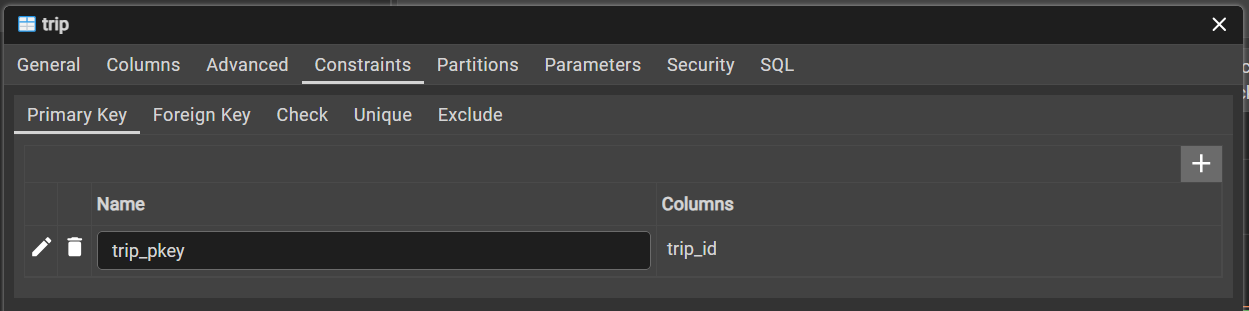


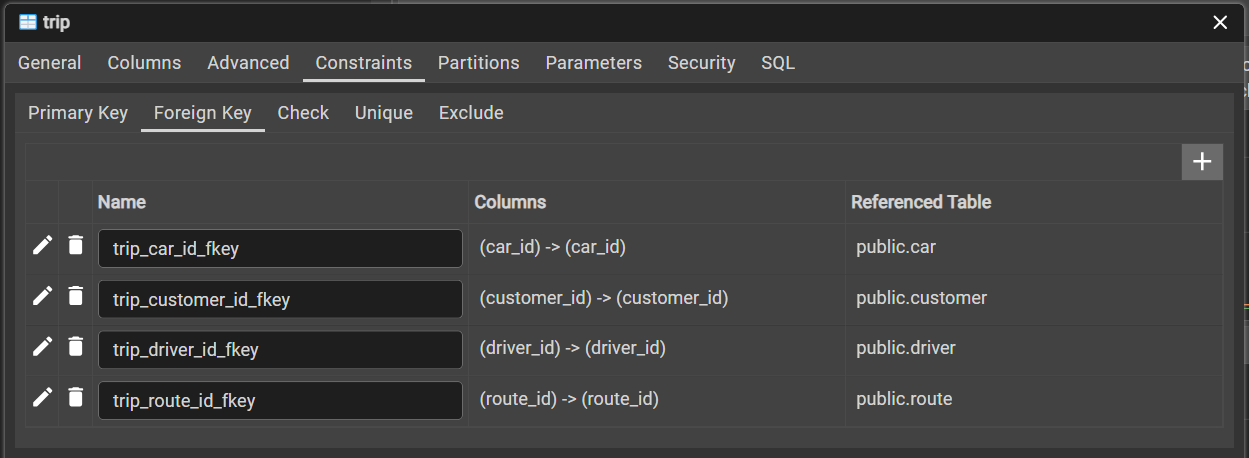


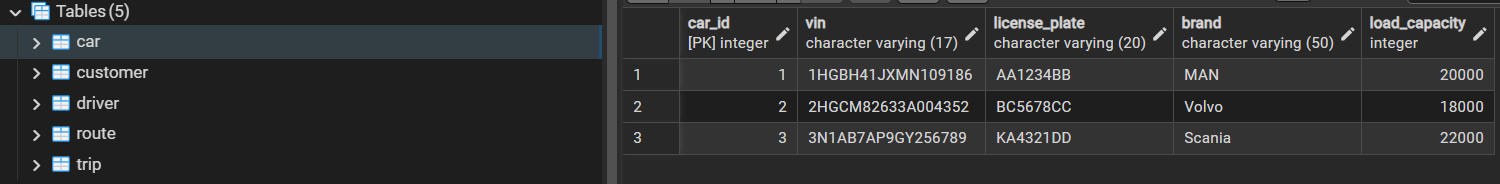
У цієї таблиці немає зовнішніх ключів (FK)

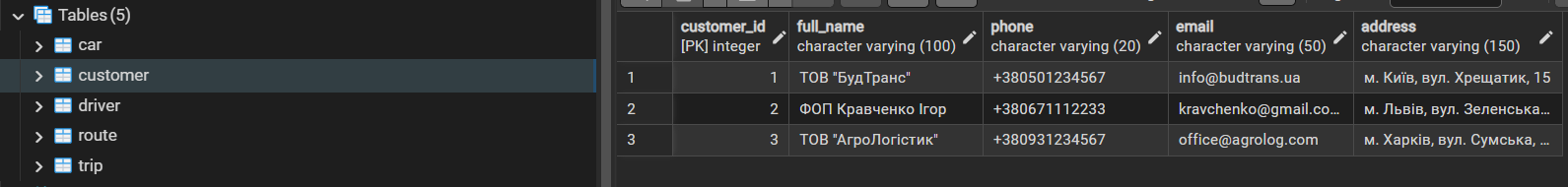
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Trip**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

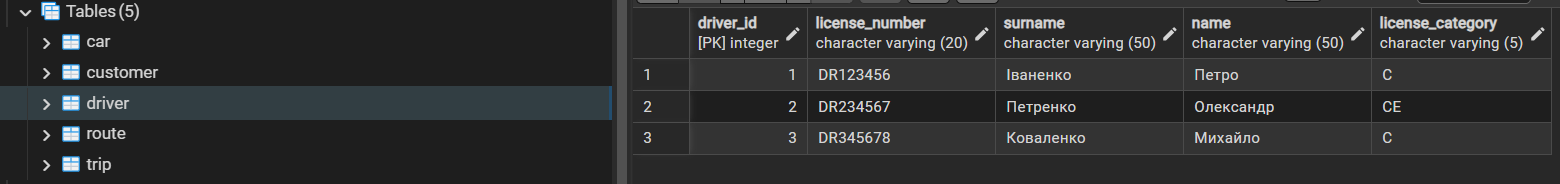


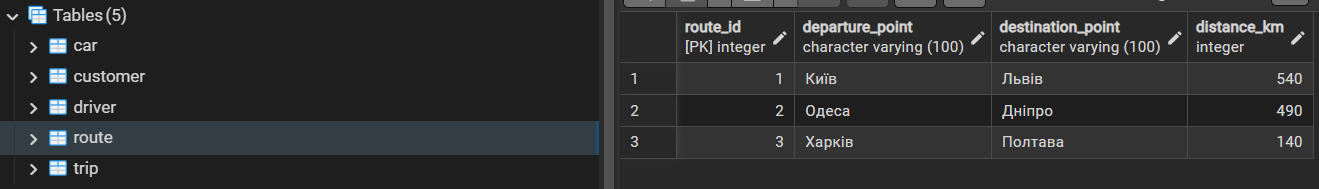


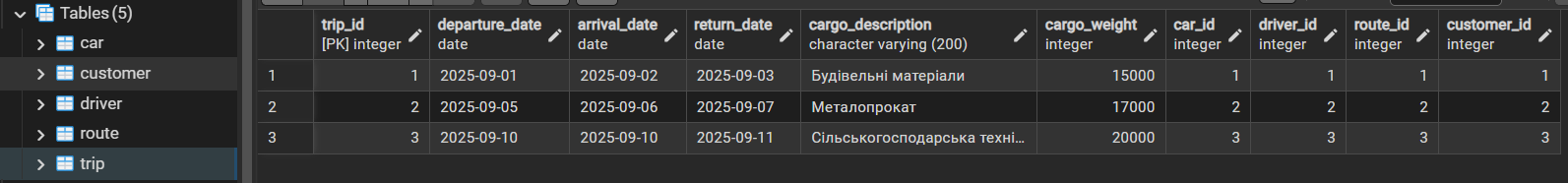












***Контакти:***

*Репозиторій GIT:* [*https://github.com/OverdrafT/kpi\_db\_and\_managment*](https://github.com/OverdrafT/kpi_db_and_managment)

*Telegram:* [*@S\_Eugene\_S*](https://t.me/S_Eugene_S)