

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

# Кафедра системного програмування та спеціалізованих комп’ютерних систем

**Розрахунково-графічна робота**

з дисципліни

**«Бази даних і засоби управління»**

Група: КВ-22

Виконала: Гончар В.

Оцінка:

Київ – 2024

### Створення додатку бази даних, орієнтованого на взаємодію з СУБД PostgreSQL

*Метою роботи* є здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.

*Завдання* роботи полягає у наступному:

1. Реалізувати функції внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі №1, засобами консольного інтерфейсу.

2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.

3. Забезпечити реалізацію пошуку за декількома атрибутами з двох та більше сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як шаблон функції LIKE оператора SELECT SQL, для логічного типу – значення True/False, для дат – у рамках діапазону дат.

4. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

### Модель «сутність-зв’язок» предметної галузі “Система управління клієнтами для компаній”

#### Основні сутності та їх атрибути:

1. **Компанія**:
   * Атрибути: Id, Назва, Власник, Країна.
2. **Клієнт**:
   * Атрибути: Id, Ім’я, Вік, Стать.
3. **Покупка**:
   * Атрибути: Id, Опис, Дата, Сума.
4. **Платіжна система**:
   * Атрибути: Id, Назва, Вебсайт.

#### Зв’язки між сутностями:

* **Компанія ↔ Клієнт** (зв’язок N:M):
  + Одна компанія може обслуговувати багатьох клієнтів.
  + Один клієнт може обслуговуватись у багатьох компаніях.
* **Клієнт ↔ Покупка** (зв’язок 1:N):
  + Один клієнт може здійснити багато покупок.
  + Одна покупка належить лише одному клієнту.
* **Компанія ↔ Покупка** (зв’язок 1:N):
  + Одна компанія може мати багато покупок.
  + Одна покупка завжди належить лише одній компанії.
* **Покупка ↔ Платіжна система** (зв’язок 1:N):
  + Одна покупка оплачується лише однією платіжною системою.
  + Один платіжний сервіс може приймати оплати багатьох покупок.

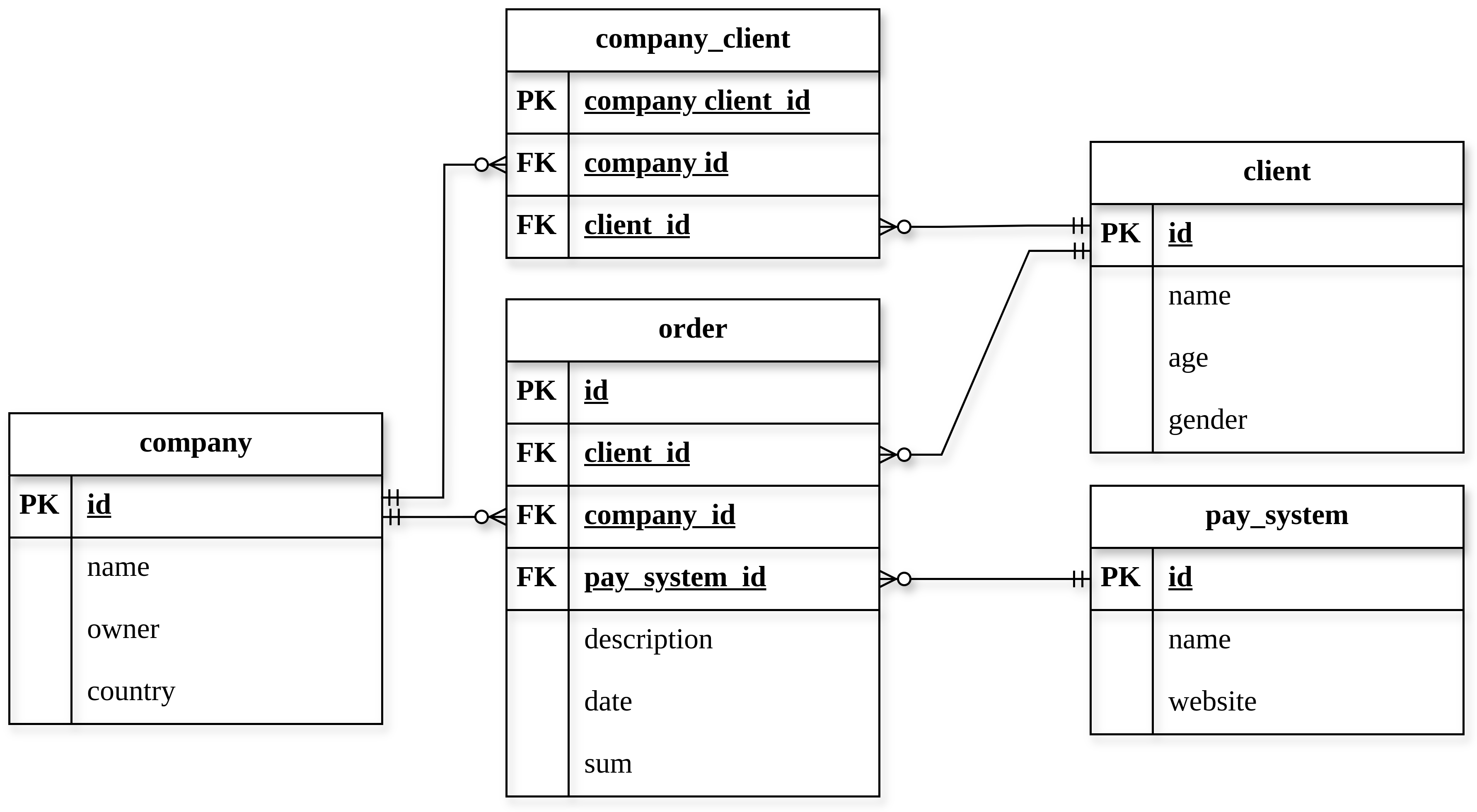


Рис. 1 - ER-діаграма. Нотація «Пташиної лапки» (Crow’s foot)

### Схема меню користувача з описом функціональності кожного пункту

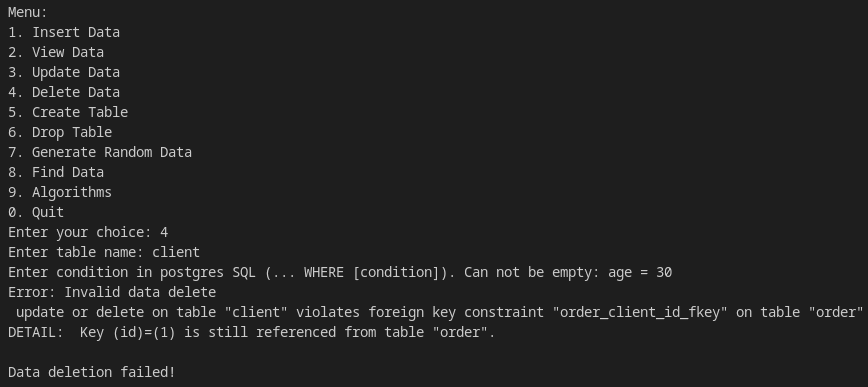
1. **Insert Data (Вставити дані)**  
   Вставка нових записів у вибрану таблицю бази даних. Користувач вводить назву таблиці, стовпці для вставки та значення для цих стовпців.
2. **View Data (Переглянути дані)**  
   Перегляд даних у вибраній таблиці. Користувач може вибрати стовпці для відображення та задати умову для фільтрації даних.
3. **Update Data (Оновити дані)**  
   Оновлення записів у вибраній таблиці. Користувач вводить назву таблиці, дані для оновлення та умову, за якою будуть змінюватись записи.
4. **Delete Data (Видалити дані)**  
   Видалення записів із таблиці. Користувач задає таблицю та умову, за якою буде виконано видалення.
5. **Create Table (Створити таблицю)**  
   Створення нової таблиці. Користувач вводить назву таблиці, список стовпців та відповідні типи даних.
6. **Drop Table (Видалити таблицю)**  
   Видалення існуючої таблиці. Користувач задає назву таблиці для видалення.
7. **Generate Random Data (Згенерувати випадкові дані)**  
   Генерація випадкових даних для вибраної таблиці. Користувач задає стовпці, типи даних, параметри генерації та кількість рядків.
8. **Find Data (Знайти дані)**  
   Пошук даних у таблиці за умовою. Користувач задає таблицю, стовпець та умову пошуку (наприклад, інтервал чисел, частковий текст тощо).
9. **Algorithms (Алгоритми)**  
   Меню алгоритмів із наступними функціями:
   * **1. Pay Systems' Total Income**  
     Відображення загального доходу за платіжними системами для заданого інтервалу сум замовлень.
   * **2. Company's Orders' thru Period**  
     Відображення кількості замовлень за компаніями у вибраному періоді часу.
   * **3. Top 5 Orders' Total Price**  
     Відображення топ-5 замовлень із найбільшою загальною сумою для вибраної компанії.
   * **0. Quit (Повернутися в головне меню)**
10. **Quit (Вихід)**  
    Завершення роботи програми.

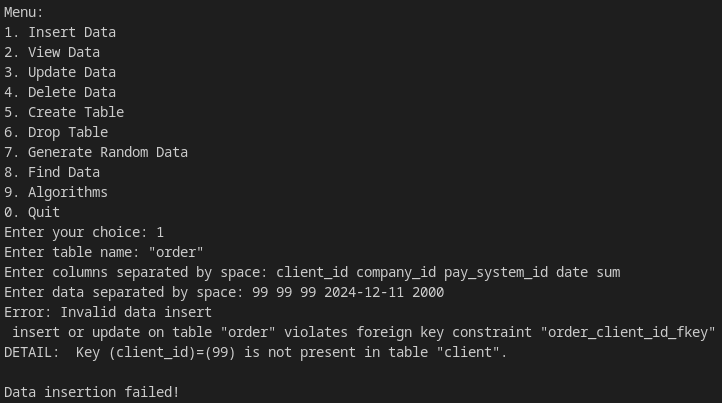
### Мова програмування та бібліотеки, що були використані

python 3.10

psycopg2

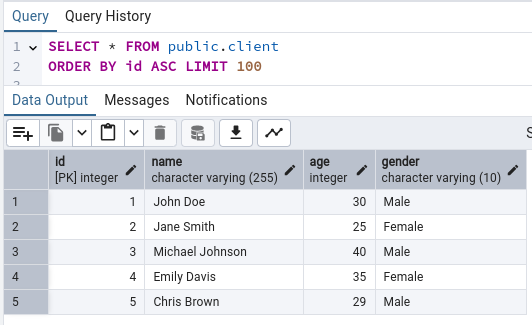
### Функції перегляду, внесення, редагування та вилучення даних



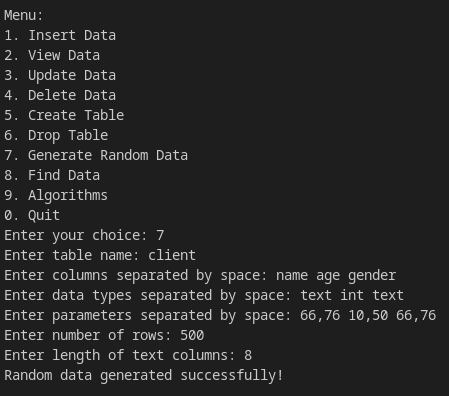


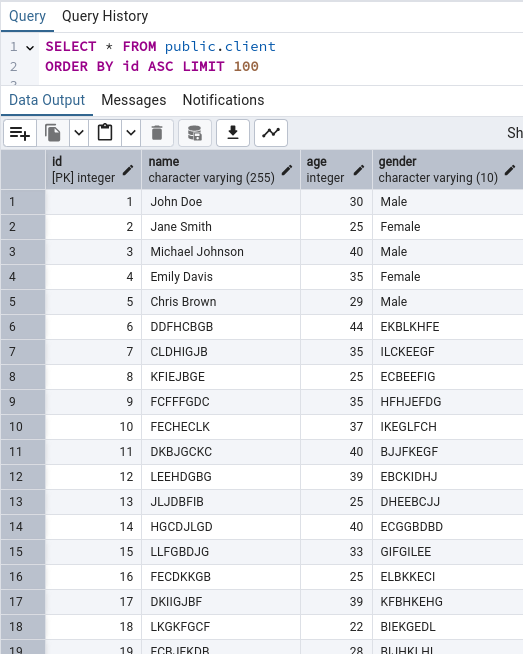
### Автоматичне пакетне генерування «рандомізованих» даних

Таблиця до внесення даних:



Та після:





#### SQL-запит, що ілюструє генерацію при визначених вхідних параметрах

INSERT INTO

   CLIENT (NAME, AGE, GENDER)

SELECT

   CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT),

   TRUNC(RANDOM() \* (50 - 10 + 1) + 10)::INTEGER,

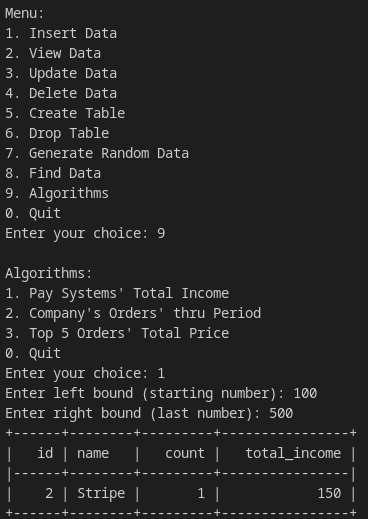
   CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT) || CHR(TRUNC(RANDOM() \* (76 - 66 + 1) + 66)::INT)

FROM

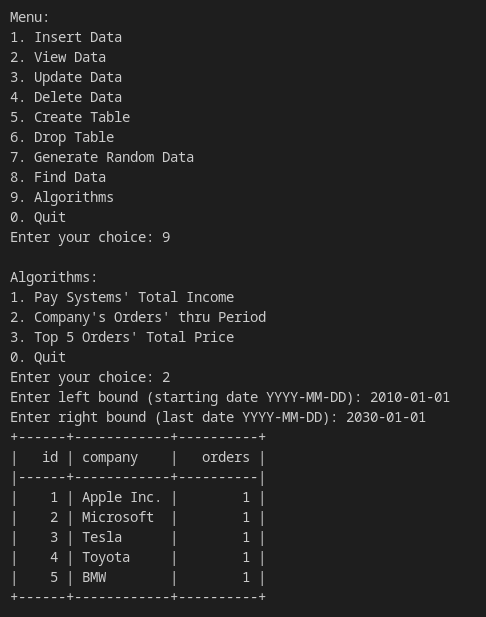
   GENERATE\_SERIES(1, 500)

### Реалізація пошуку за декількома атрибутами з двох та більше сутностей одночасно

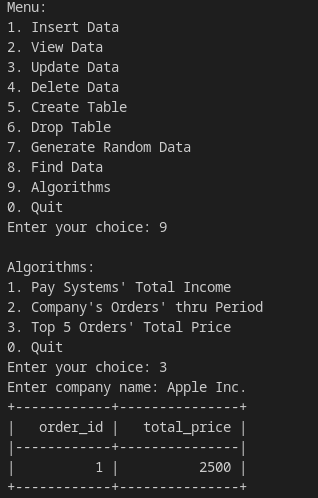
#### 1:



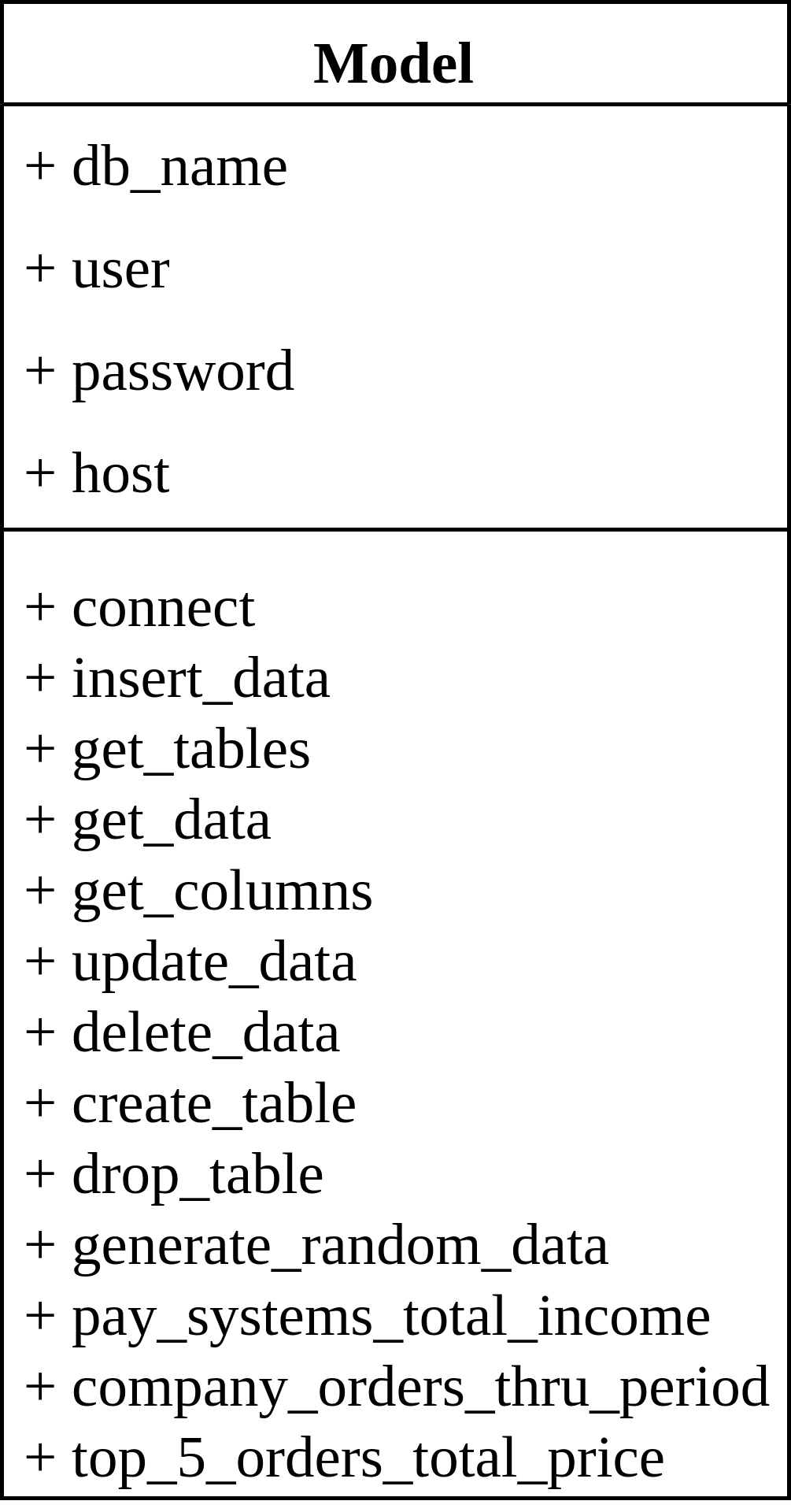
#### 2:



#### 3:



### Програмний код згідно шаблону MVC



Модуль Model є класом для роботи з базами даних PostgreSQL. Він реалізує основні операції для взаємодії з базою даних, включаючи створення таблиць, виконання CRUD-операцій (створення, читання, оновлення, видалення), генерацію випадкових даних та виконання специфічних запитів. Основні функції:

1. **Підключення до бази даних**:
   * connect: встановлює з'єднання з базою даних і створює курсор для виконання SQL-запитів.
2. **Операції з таблицями**:
   * create\_table: створює таблицю з вказаними колонками та типами даних.
   * drop\_table: видаляє таблицю з бази даних.
   * get\_tables: отримує список усіх таблиць у базі даних.
   * get\_columns: отримує список колонок таблиці.
3. **CRUD-операції**:
   * insert\_data: додає записи в таблицю.
   * get\_data: отримує дані з таблиці з можливістю фільтрації за умовою.
   * update\_data: оновлює дані в таблиці на основі заданої умови.
   * delete\_data: видаляє дані з таблиці за умовою.
4. **Генерація даних**:
   * generate\_random\_data: генерує випадкові дані різних типів (числа, текст, дати, boolean тощо) та додає їх у таблицю.
5. **Специфічні запити**:
   * pay\_systems\_total\_income: обчислює загальний дохід для кожної платіжної системи у вказаному діапазоні.
   * company\_orders\_thru\_period: рахує кількість замовлень кожної компанії за заданий період.
   * top\_5\_orders\_total\_price: отримує топ-5 замовлень з найвищою сумою для вказаної компанії.

Модуль призначений для спрощення інтеграції Python із PostgreSQL, автоматизації типових завдань із базами даних та виконання кастомних аналітичних запитів.

github:

telegram: https://t.me/Vika\_Honchar