**Національний технічний університет України**

**“Київський політехнічний інститут ім. Ігоря Сікорського”**

**Факультет прикладної математики**

**Кафедра системного програмування і спеціалізованих комп’ютерних систем**

**ЛАБОРАТОРНА РОБОТА №1**

**з дисципліни**

**“Бази даних та засоби управління”**

**Груцпа: КВ-33**

**Виконав: Козлов С. О.**

**Оцінка:**

**Тема роботи:** Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL.

**Мета роботи:** здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

**Виконав:** Козлов С.О.

**Telegram:** [@Teollan](https://t.me/Teollan)

**Github:** <https://github.com/Teollan/2025-bdzu-lab1>

**Предметна галузь:** Система управління клієнтами для компаній.

**Перелік сутностей та їх призначення**

*Company* (компанія) – сутність, що узагальнено описує компанію. В якості ключа використовує унікальний ідентифікатор. Також містить назву компанії у зрозумілому для людини форматі.

*Customer* (клієнт) – сутність що узагальнено описує людину як потенційного або поточного клієнта компанії. В якості ключа використовує унікальний ідентифікатор. Містить ім’я та прізвище людини, а також її номер телефону та адресу електронної пошти для маркетингових задач компанію. Email адреса є унікальною для кожного клієнта але не є стабільною (напр. клієнт може подати запит на видалення персональних даних), що створить труднощі в підтримці БД.

*Lead* (лід, зачіпка) – слабка сутність, що створюється компанією і пов’язує компанію з клієнтом.

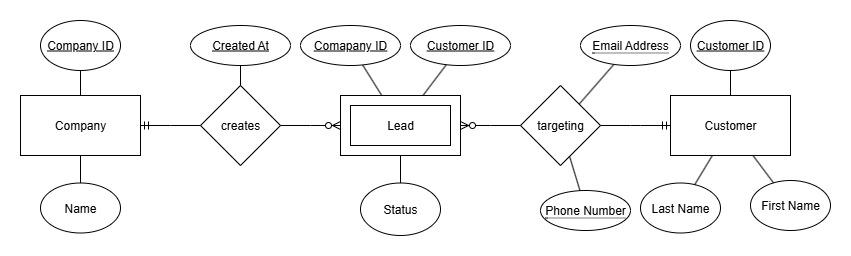
Унікальним ключем є комбінація ідентифікаторів компанії і клієнта з часовою міткою (timestamp) створення ліда. Це зумовлено тим, що клієнт може мати кілька точок контакту з компанією, розподілених у часі.

Сутність також зберігає поточний статус обробки ліда:

* Pending - очікує обробки
* In progress - в процесі взаємодії з клієнтом
* Won - клієнт купив товар/послугу
* Lost - клієнт відмовився від співпраці

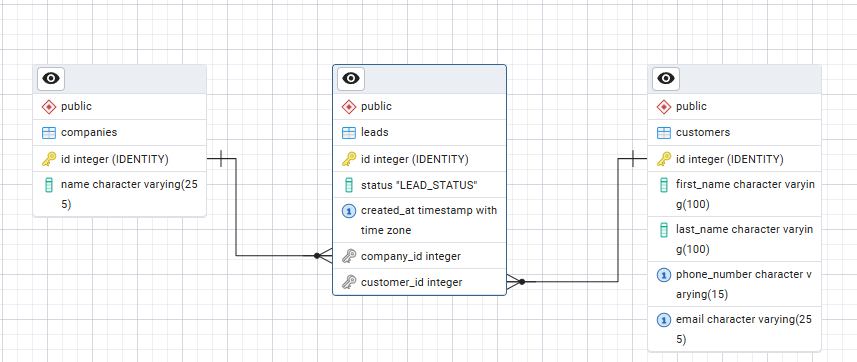
**ER модель**

Тут і далі використовується нотація “Пташина лапка (Crow’s foot)”



**Перетворення моделі на таблиці БД**

Оскільки в моделі відсутні зв’язки M:N, то ER модель можна представити за допомогою 3-х таблиць: companies, leads та customers.



**Нормалізація БД**

1НФ

Умови:

* Кожен запис унікальний.
* Кожна колонка є атомарною.

Аналіз:

Для кожної з таблиць визначений унікальний PK, а для ключових атрибутів leads та companies визначений UNIQUE CONSTRAINT. Отже кожен запис є унікальним.

Кожна представляє тільки одне значення, спискові типи не використовуються. Отже кожна колонка є атомарною.

Висновок: Таблиці нормалізовані до 1НФ.

2НФ

Умови:

* Таблиці нормалізовані до 1НФ.
* Кожен не-ключовий атрибут функціонально залежить від ключа.

Аналіз:

2НФ актуальна тільки для композитних ключів, тому потрібно перевірити тільки таблицю leads.

Випишемо її функціональні залежності:

id → company\_id, customer\_id, created\_at, status

company\_id, customer\_id, created\_at → status, id

Отже, статус ліда функціонально залежить від кожного з компонентів композитного ключа.

Висновок: Таблиці нормалізовані до 2НФ.

3НФ

Умови:

* Таблиці нормалізовані до 3НФ.
* Немає транзитивних залежностей неключових атрибутів у межах однієї таблиці.

Аналіз:

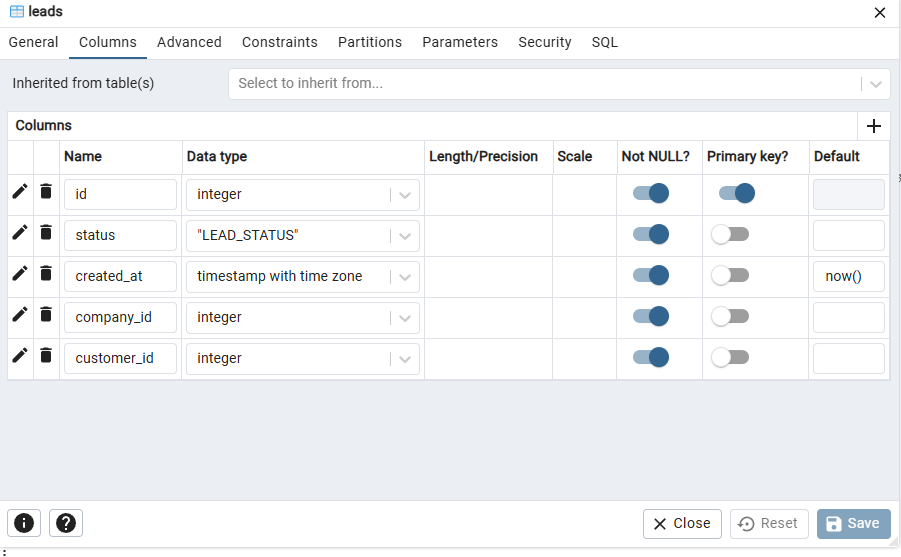
Випишемо функціональні залежності:

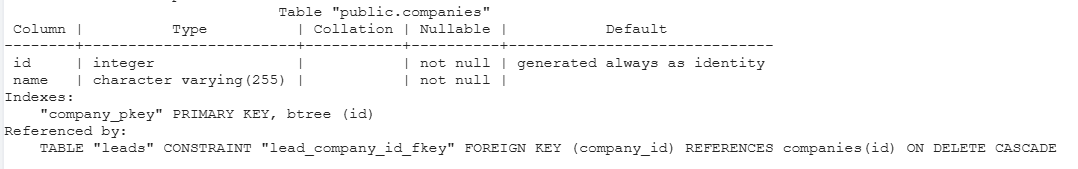
* companies:
  + id → name
* customers:
  + id → first\_name, last\_name, phone\_number, email
  + email → first\_name, last\_name, phone\_number, id
  + phone\_number → first\_name, last\_name, email, id
* leads:
  + id → company\_id, customer\_id, created\_at, status
  + company\_id, customer\_id, created\_at → status, id

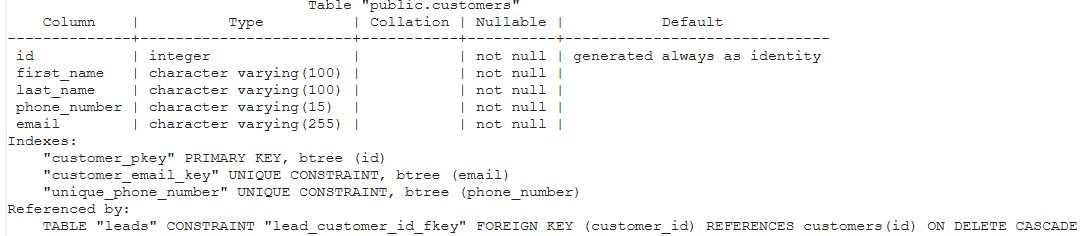
Бачимо, що транзитивні залежності від неключових атрибутів відсутні.

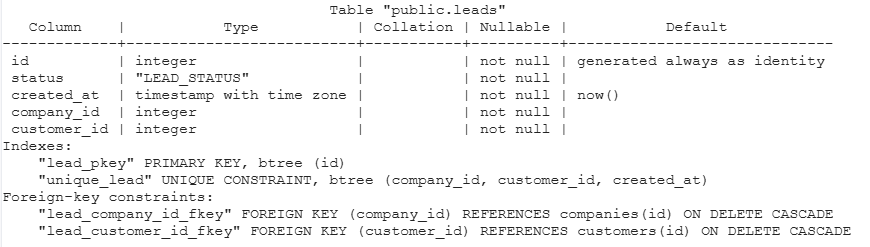
Висновок: Таблиці нормалізовані до 3НФ.

**Робота з pgAdmin 4**

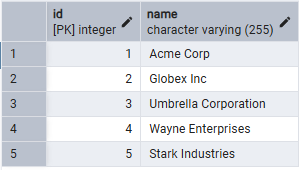
****



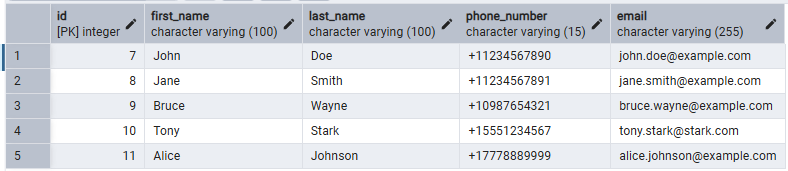




Companies:



Customers:



Leads:

