

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

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

**Лабораторна робота №1**

з дисципліни **Бази даних і засоби управління**

*на тему: “Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL”*

Виконав: студент ІII курсу

групи КВ-93

Рудницький П.Є.

Перевірив: Павловський В.І.

Київ – 2021

**Мета роботи**

Здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

**Завдання лабораторної роботи**

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 pgAdmin4.

**Модель «сутність-зв’язок» мережі магазинів**

Обрана предметна галузь передбачає зберігання товарів в кожному окремому філіалу мережі магазинів. Згідно цієї області для побудови бази даних було виділено наступні сутності:

1. Сутність “Shop” містить такі атрибути як ID, адреса магазину та контакти управляючого філіалом. Слугує для зберігання інформації про філіал
2. Сутність “Discount” містить такі атрибути як ID, відсоток знижки та термін її дії. Слугує зберігання інформації про знижку.
3. Сутність “Product\_discount” містить такі атрибути як ID, ідентифікатор продукту та ідентифікатор знижки. Слугує для зв’язування знижки та продукту.
4. Сутність “Product” містить такі атрибути як ID, ідентифікатор відділу, до якого належить продукт та назву продукту. Слугує для зберігання інформації про конкретний продукт.

**Опис зав’язків у БД «Мережа магазинів»**

Один магазин може мати багато продуктів, але в свою чергу конкретний продукт може зберігатись тільки в одному магазині. Тому між сутностями Shop та Product зв’язок 1:M.

Оскільки один продукт може підпідати під декілька знижок(наприклад товар «Сардини» підпадає під знижку для консерв та знижку для рибних продуктів), а одна знижка може включати багато товарів, то між сутностями Product та Discount встановлено зв’язок N:M. Для того, щоб правильно встановити цей зв’язок, створена сутність Product\_discount, яка містить в собі ідентифікатор продукту та ідентифікатор знижки, яка відноситься до продукту.

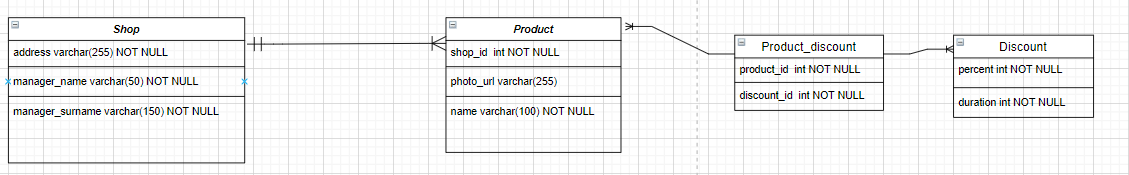


Рисунок 1 - Концептуальна модель предметної області «Мережа магазинів»

**Опис таблиць у БД «Мережа магазинів»**

Сутність «Shop» була перетворена у таблицю Shop.

Сутність «Product» була перетворена у таблицю Product, а зв'язок M:1 цієї сутності із сутністю «Shop» зумовив появу у ній зовнішнього ключа shop\_id.

Сутність «Discount» була перетворена у таблицю Discount.

Сутність «Product\_discount» була перетворена у таблицю Product\_discount, зв’зки М:1 із сутностями Product та Discount зумовили появу зовнішніх ключів product\_id та discount\_id відповідно

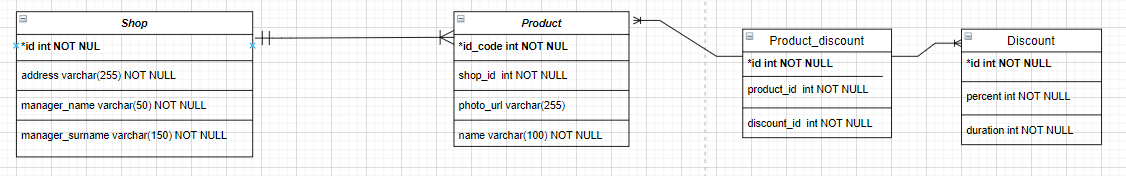
**

Рисунок 2 - Логічна модель предметної області «Мережа магазинів

***Опис структури БД «Мережа магазинів»***

|  |  |  |
| --- | --- | --- |
| Відношення | Атрибут | Розмір (тип) |
| ***Відношення “Shop”***  Вміщує інформацію про філіал магазину | **\*id** – унікальний ID філіалу магазина в БД, не допускає NULL  **address** – адреса магазину, не допускає NULL  **manager\_name** – ім’я управляючого магазином, не допускає NULL  **manager\_surname** – прізвище управляючого магазином, не допускає NULL | Числовий  Текстовий(255)  Текстовий(50)  Текстовий(150) |
| ***Відношення “Discount”***  Вміщує інформацію про знижку | **\*id** – унікальний ID знижки в БД, не допускає NULL  **percent –** відсоток знижки, не допускає NULL  **duration –** термін дії знижки, не допускає NULL | Числовий  Числовий  Числовий |
| ***Відношення “Product\_discount”***  Вміщує інформацію про продукт та знижку | **\*id** – унікальний ID таблиці в БД, не допускає NULL  **product\_id –** ідентифікатор продукту, не допускає NULL  **discount\_id –** ідентифікатор знижки, не допускає NULL | Числовий  Числовий  Числовий |
| ***Відношення “Product”***  Вміщує інформацію про конкретний продукт | **\*id\_code** – унікальний ID продукту в БД, не допускає NULL  **photo\_url** – посилання на фото продукту  **name** – назва продукту, не допускає NULL | Числовий  Текстовий(255)  Текстовий(100) |

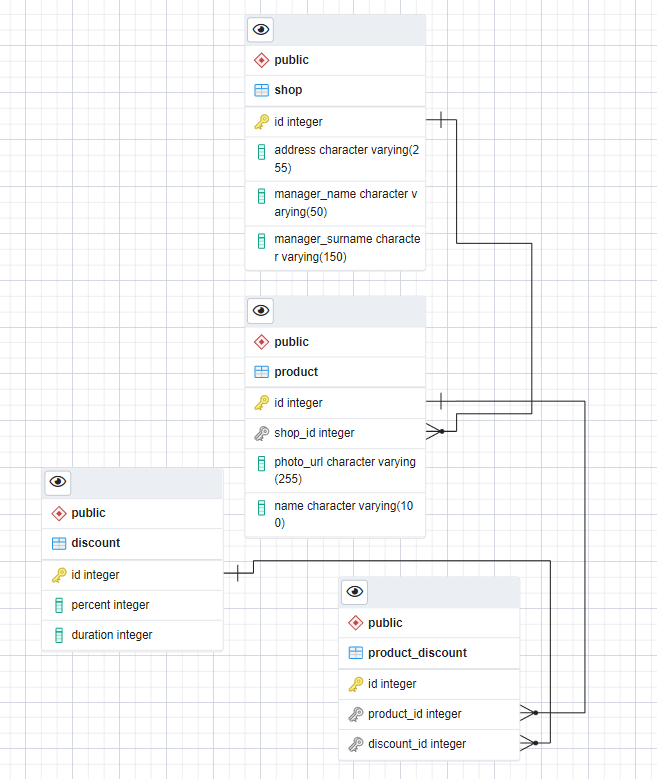


Рисунок 3 – Схема БД «Мережа магазинів» у pgAdmin 4

**Нормалізація схеми бази даних до третьої нормальної форми**

Таблиці бази даних «Мережа магазинів» відповідають першій нормальній формі оскільки в них немає повторюючих рядків та масивів даних.

Таблиці бази даних відповідають другій нормальній формі, тому що вони

відповідають першій нормальній формі та кожна таблиця має первинний ключ

(Primary key), від якого залежать всі неключові атрибути.

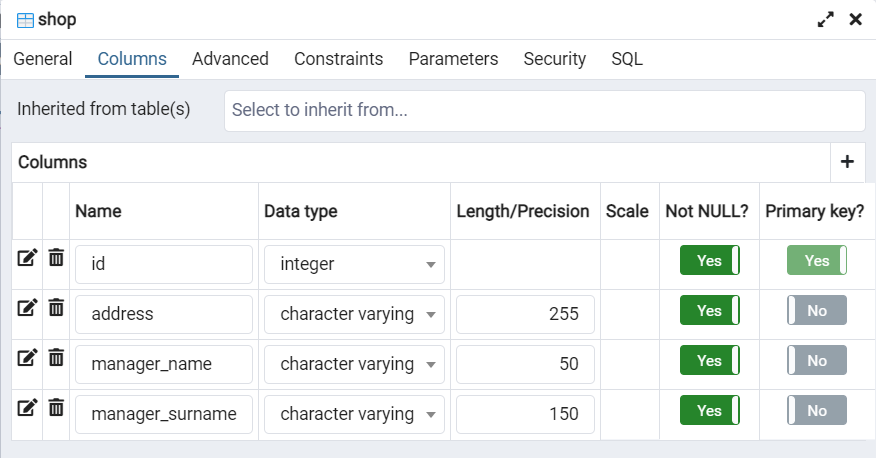
Таблиці бази даних відповідають третій нормальній формі, тому що вони

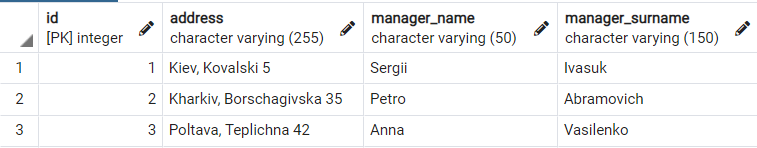
відповідають другій нормальній та не мають транзитивних залежностей для не

ключових атрибутів.

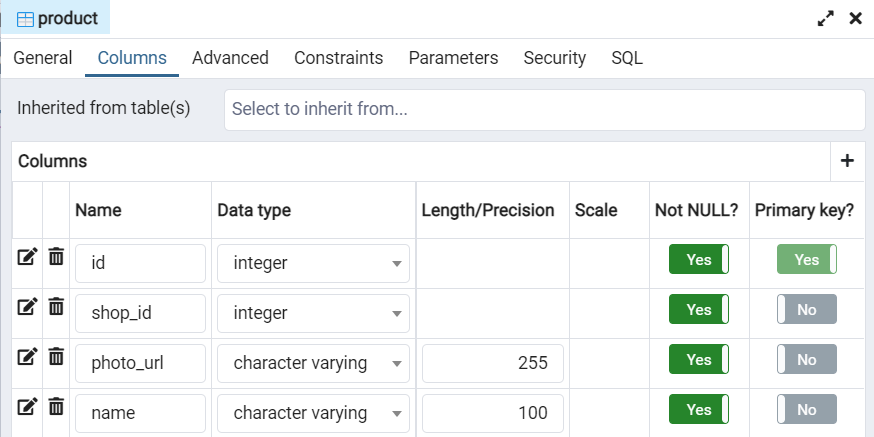
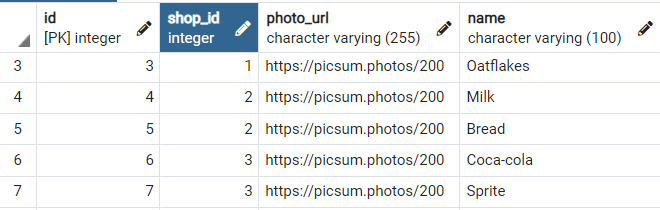
**Таблиці бази даних «Мережа магазинів» у pgAdmin4**

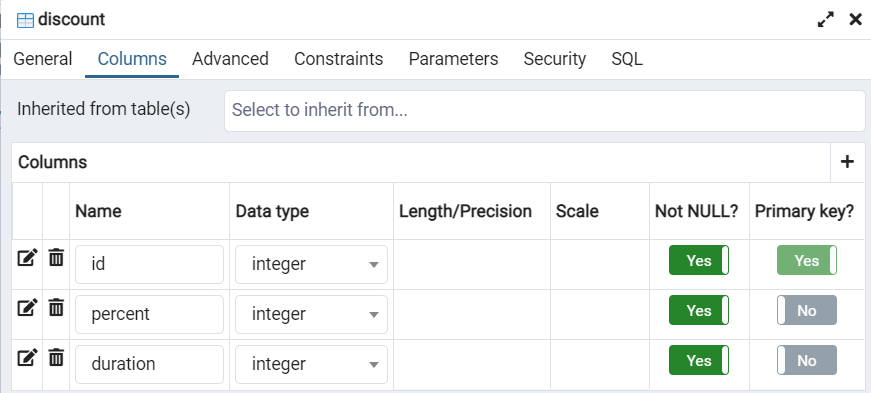
**Shop**

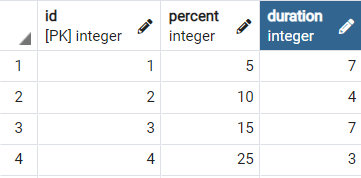




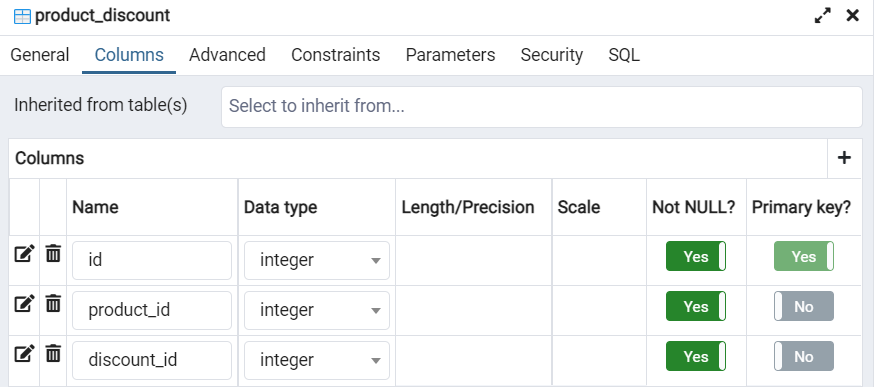
**Product**

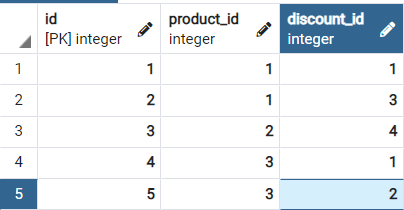
 ****

**Discount**

****

**Product\_discount**

****



**Згенерований SQL-текст**

BEGIN;

CREATE TABLE IF NOT EXISTS public.discount

(

id integer NOT NULL,

percent integer NOT NULL,

duration integer NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS public.product

(

id integer NOT NULL,

shop\_id integer NOT NULL,

photo\_url character varying(255) NOT NULL,

name character varying(100) NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS public.product\_discount

(

id integer NOT NULL,

product\_id integer NOT NULL,

discount\_id integer NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS public.shop

(

id integer NOT NULL,

address character varying(255) NOT NULL,

manager\_name character varying(50) NOT NULL,

manager\_surname character varying(150) NOT NULL,

PRIMARY KEY (id)

);

ALTER TABLE public.product

ADD FOREIGN KEY (shop\_id)

REFERENCES public.shop (id)

NOT VALID;

ALTER TABLE public.product\_discount

ADD FOREIGN KEY (discount\_id)

REFERENCES public.discount (id)

NOT VALID;

ALTER TABLE public.product\_discount

ADD FOREIGN KEY (product\_id)

REFERENCES public.product (id)

NOT VALID;

END;