Изображение выглядит как текст

Автоматически созданное описание

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”

Виконала: студентка 3 курсу

групи КВ-93

Баранова Є.В.

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.

**Завдання 1**

Модель «сутність-зв’язок» галузі продажу напоїв в кав’ярні.

Модель має чотири сутності: order, drink, drink\_category, category.

Drink: містить інформацію про напій. Кожен напій має власний id, назву і ціну.

Order: містить інформацію про клієнта і обраний ним напій. Кожне замовлення має власне id, напій і власника.

Drink\_category: містить інформацію про конкретній напій. Містить id напою і id категорії. Слугує для зв’язування напою та категорії.

Category: містить інформацію про категорії напоїв. Кожна категорія має власний id та назву.

**Опис зв’язків**

Замовлення може бути лише одне, але в замовленні може бути багато напоїв, тому між сутностями drink та order існує зв’язок 1: М.

В одній категорії може бути багато напоїв (холодним напоєм з цукром може бути і американо і лимонад) і один напій може відноситися до декількох категорій (американо може бути холодним напоєм без цукру і гарячим напоєм з цукром), тому між сутностями drink та category існує зв’язок M:N. Для того, щоб правильно встановити цей зв'язок, створена сутність Drink\_category, яка містить в собі ідентифікатор категорії та ідентифікатор напою, який відноситься до цієї категорії.

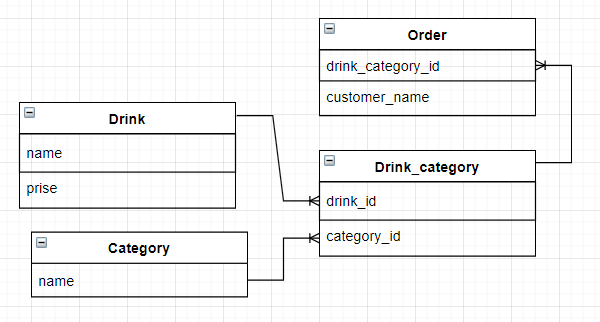


Рисунок 1 – Концептуальна модель

**Завдання 2**

**Опис таблиць**

Сутність Drink\_category була перетворена у таблицю Drink\_category, а зв’язки М:1 із сутностями Category та Drink зумовили появу у ній зовнішніх ключів category\_id та drink\_id.

Сутність Order була перетворена у таблицю Order, а зв’язок M:1 цієї сутності з сутністю Drink\_category зумовив появу у ній зовнішнього ключа drink\_category\_id.

Сутність Drink була перетворена у таблицю Drink.

Сутність Category була перетворена у таблицю Category.

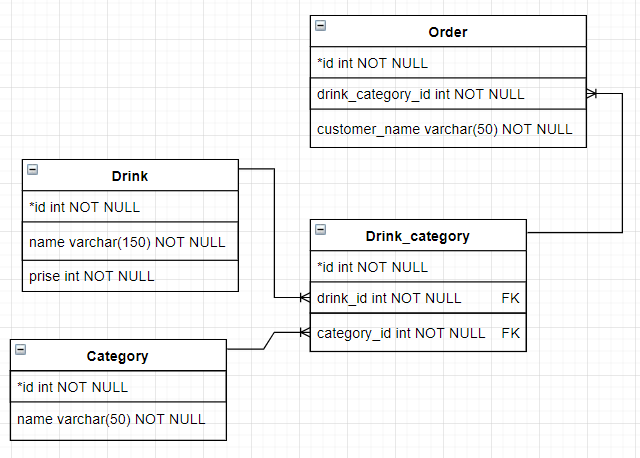


Рисунок 2 – Логічна модель

Таблиця 1 – Опис структури

|  |  |  |
| --- | --- | --- |
| **Відношення** | **Атрибут (кожен атрибут не допускає NULL)** | **Тип** |
| Drink  (Інформація про конкретний напій) | id – унікальний ID напою в БД  name – назва напою  price – ціна напою | Числовий  Текстовий (150)  Числовий |
| Order  (Інформація про замовлення) | id – унікальний ID замовлення в БД  drink\_category\_id – ID конкретного напою  customer\_name – ім’я клієнта | Числовий  Числовий  Текстовий (50) |
| Drink\_category  (Інформація про категорію та напій) | id – унікальний ID клієнта в БД  drink \_id – ID напою  category\_id – ID категорії | Числовий  Числовий  Числовий |
| Category  (Інформація про категорії напоїв) | id – унікальний ID категорії напою в БД  name – назва типу напою | Числовий  Текстовий (50) |

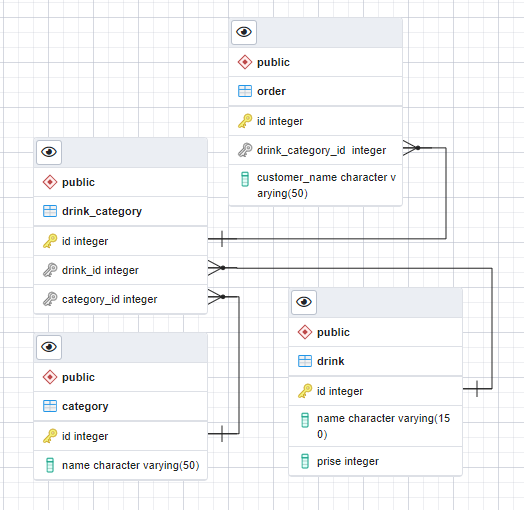


Рисунок 3 – Схема бази даних

**Завдання 3**

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

Схема бази даних задовольняє умовам першої нормальної форми, оскільки атрибути у всіх таблицях є атомарними, відсутні багатозначні та композитні атрибути і в них немає повторюючихся рядків даних.

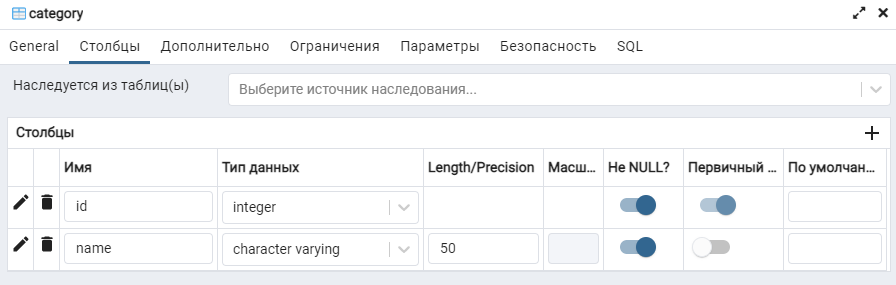
Схема бази даних задовольняє умовам другої нормальної форми, так як вона знаходиться в першій нормальній формі, та кожний не ключовий атрибут залежить від єдиного ключа.

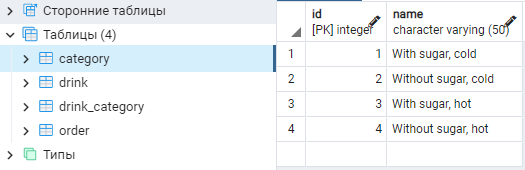
Схема бази даних задовольняє умовам третьої нормальної форми, оскільки вона задовольняє умовам 2НФ та неключові атрибути не залежать від інших атрибутів, які не є первинними ключами, тобто немає транзитивних функціональних залежностей не ключових атрибутів від ключових.

**Завдання 4**

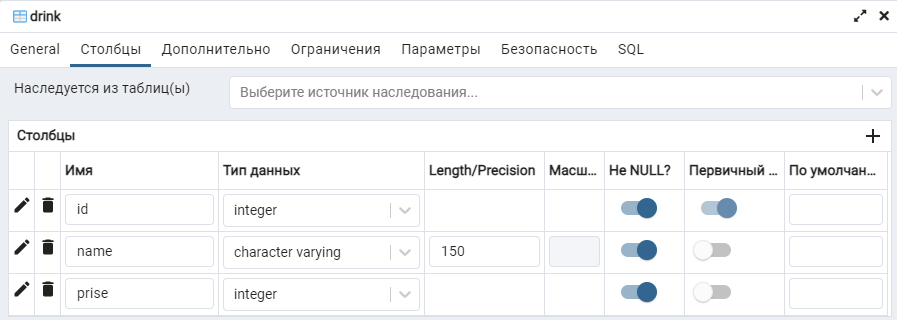
**Копії екрану з pgAdmin4, що відображають назви та типи стовпців та обмеження на стовпці**

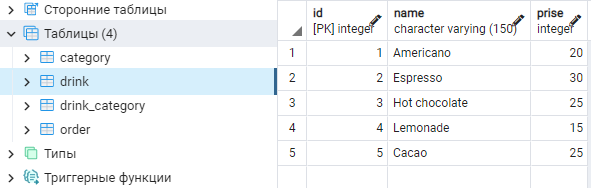
**Category**



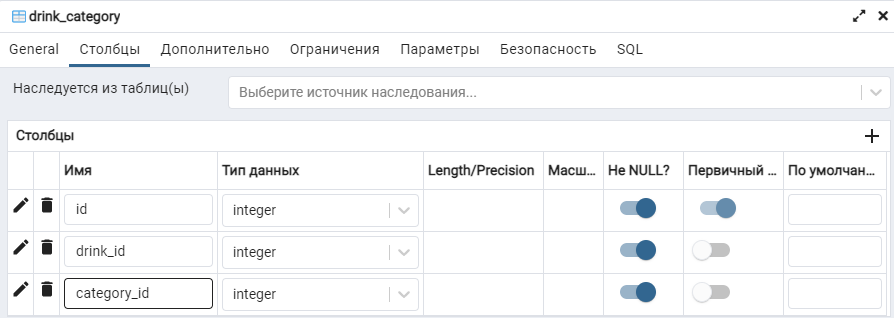


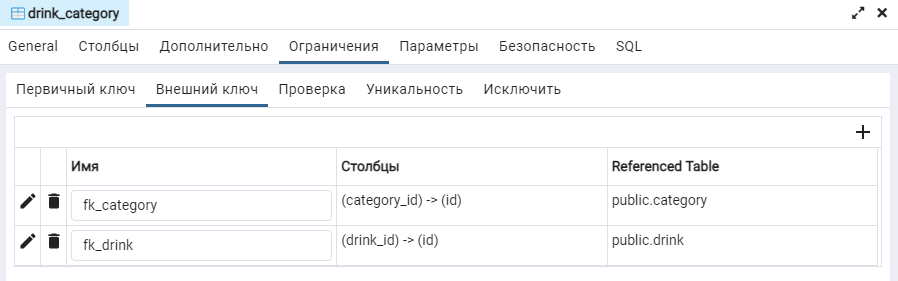
**Drink**

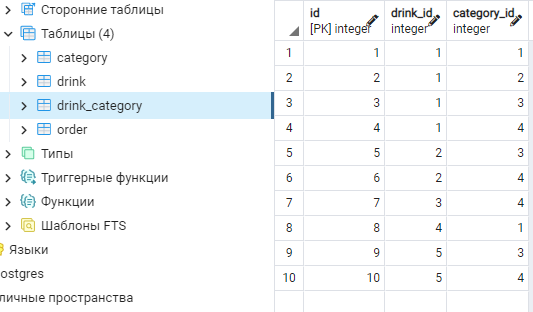




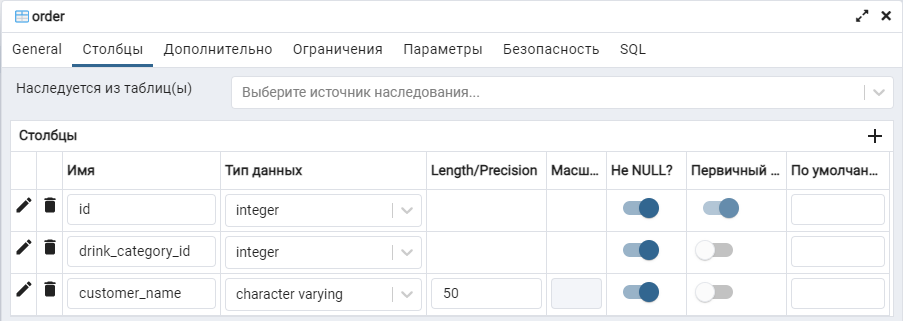
**Drink\_category**

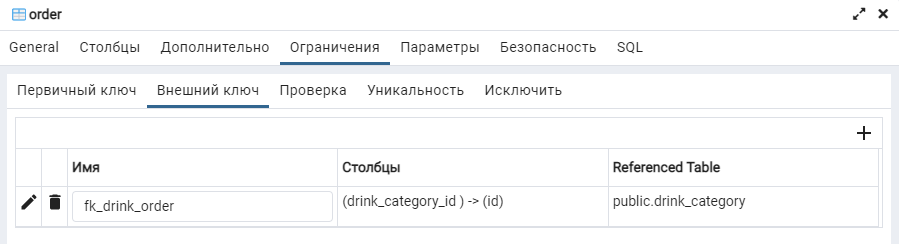


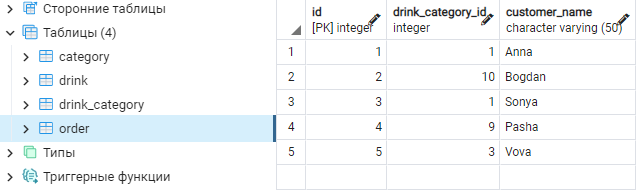




**Order**







**Згенерований SQL-текст**

BEGIN;

CREATE TABLE IF NOT EXISTS public.category

(

id integer NOT NULL,

name character varying(50) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT category\_pkey PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS public.drink

(

id integer NOT NULL,

name character varying(150) COLLATE pg\_catalog."default" NOT NULL,

prise integer NOT NULL,

CONSTRAINT drink\_pkey PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS public.drink\_category

(

id integer NOT NULL,

drink\_id integer NOT NULL,

category\_id integer NOT NULL,

CONSTRAINT "customer \_pkey" PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS public."order"

(

id integer NOT NULL,

"drink\_category\_id " integer NOT NULL,

customer\_name character varying(50) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT order\_pkey PRIMARY KEY (id)

);

ALTER TABLE IF EXISTS public.drink\_category

ADD CONSTRAINT fk\_category FOREIGN KEY (category\_id)

REFERENCES public.category (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public.drink\_category

ADD CONSTRAINT fk\_drink FOREIGN KEY (drink\_id)

REFERENCES public.drink (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public."order"

ADD CONSTRAINT fk\_drink\_order FOREIGN KEY ("drink\_category\_id ")

REFERENCES public.drink\_category (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

END;