

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-01

Орел І. В.

Перевірив:

Київ – 2022

*Метою роботи є* здобуття практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

*Завдання* роботи полягає у наступному:

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.

*Зміст звіту*

1. Опис проблемного середовища;
2. Концептуальна модель предметної області;
3. Логічна модель (схема) даних БД;
4. Склад СУБД PostgreSQL;
5. Список обмежень цілісності в термінах СУБД PostgreSQL;
6. Фізична модель (схема) даних БД в pgAdmin 4;
7. Приклад вмісту БД.

**Опис предметної області «Магазини»**

Обрана предметна область передбачає купівлю і продаж товарів з точки зору магазинів. Кожен магазин має хоча б одного робітника. У кожному магазині є декілька відділів. У кожному відділі в певний момент часу можуть знаходитись товари. Кожен товар може належати до багатьох покупок, і водночас покупка може містити декілька товарів.

**Опис сутностей предметної області**

Для побудови бази даних для обраної області було виділено сутності, зображені на рисунку 1:

1. Магазин (shop), з атрибутами: код магазину, назва, місце, вік. Призначена для збереження інформації про всі існуючі магазини.
2. Робітник (employee), з атрибутами: код робітника, код магазину, до якого належить робітник, прізвище, вік, номер телефону. Призначена для збереження інформації про робітників магазинів.
3. Відділ (storebranch), з атрибутами: код відділу, код магазину, назва. Призначена для збереження інформації про існуючі зали.
4. Товар (product), з атрибутами: код товару, ім’я, код відділу, ціна продажу, продано (bool). Призначена для фіксування продажу товару і збереження його параметрів.
5. Покупка (order), з атрибутами: код покупки, код магазину, до якого належить покупка, ціна покупки.

**Опис зв’язків між сутностями предметної області**

Сутність «Магазин» має зв’язок 1:N по відношенню до сутності «Робітник», тому що один робітник в певний момент часу може працювати лише в одному магазині, а в кожному магазині можуть працювати багато робітників.

Сутність «Магазин» має зв’язок 1:N по відношенню до сутності «Відділ»: кожен відділ належить лише до одного певного магазину, але магазин може мати багато відділів.

Сутність «Магазин» має зв’язок 1:N по відношенню до сутності «Покупка»: кожна покупка має буди зроблена лише в одному конкретному магазині, але в магазині може бути проведено багато покупок.

Сутність «Відділ» має зв’язок 1:N по відношенню до сутності «Товар», тому що кожен товар може належати лише єдиному відділу магазину, але один відділ може містити багато товарів.

Сутність «Товар» має зв’язок M:N по відношенню до сутності «Покупка» адже кожна покупка може містити певну кількість товарів, і кожний товар може належати до різних покупок.

**Концептуальна модель предметної області “Магазини”**

Концептуальна модель наведена на рисунку 1.

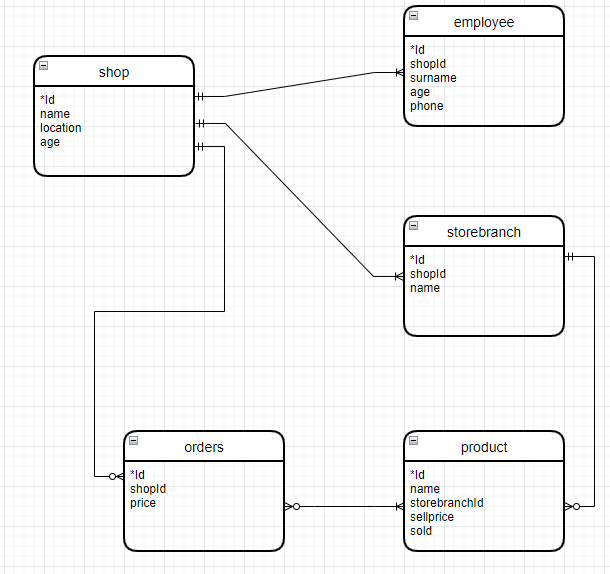


Рисунок 1 - ER-діаграма, побудована за нотацією “Пташиної лапки (Crow’s foot)” (інструмент: draw.io)

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

Сутність «Магазин» перетворено у таблицю «shop».

Сутність «Робітник» перетворено у таблицю «employee».

Сутність «Відділ» перетворено у таблицю «storebranch».

Сутність «Товар» перетворено у таблицю «product».

Сутність «Покупка» перетворено у таблицю «orders».

Зв’язок M:N між сутностями «Товар» та «Покупка» зумовив появу в ER моделі сутності «Товар-покупка», яку перетворено в таблицю «product\_orders» із зовнішніми ключами productId та ordersId.

**Логічна модель (схема) БД «Магазини»**

Логічну модель (схему бази даних) наведено на рисунку 2.

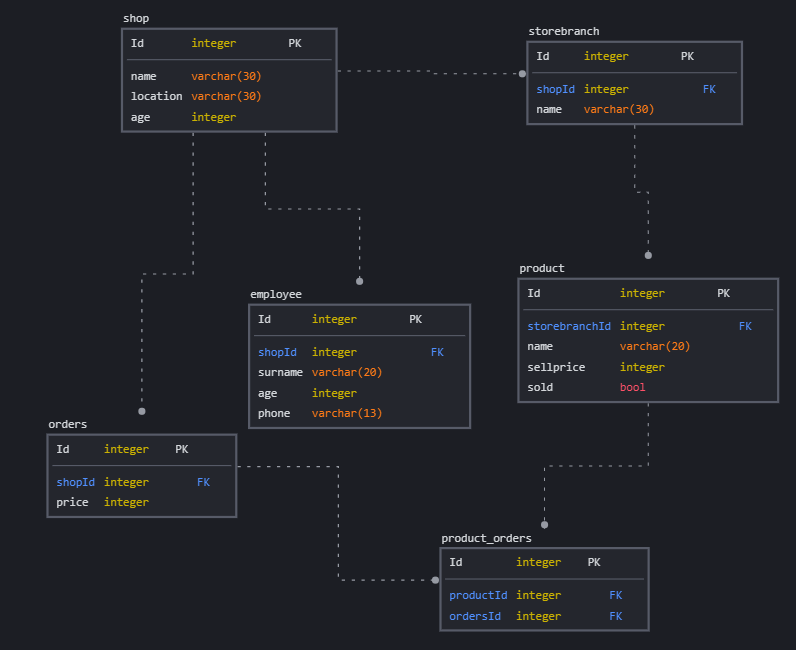


Рисунок 2 - Схема бази даних (інструмент: sqldbm.com)

Опис сутностей та їх атрибутів наведено в Таблиці 1.

Таблиця 1 – Опис структури бази даних "Магазини"

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **shop –** містить всі існуючі магазини та інформацію про них | **Id** – *унікальний ідентифікатор магазину*  **name** – *назва магазину*  **location** – *місце знаходження магазину*  **age** – *вік магазину* | **integer** *(числовий)*  **character varying** *(рядок)*  **character varying** *(рядок)*  **integer** *(числовий)* |
| **employee –** містить інформацію про всіх робітників магазинів | **Id** – *унікальний ідентифікатор робітника*  **shopId** – *унікальний ідентифікатор магазину*  **surname** – *прізвище робітника*  **age** – *вік робітника*  **phone** – *телефон робітника* | **integer** *(числовий)*  **integer** *(числовий)*  **character varying** *(рядок)*  **integer** *(числовий)*  **character varying** *(рядок)* |
| **storebranch –** містить інформацію про всі відділи магазинів | **Id** – *унікальний ідентифікатор відділу магазину*  **shopId** – *унікальний ідентифікатор магазину*  **name** – *назва відділу магазину* | **integer** *(числовий)*  **integer** *(числовий)*  **character varying** *(рядок)* |
| **product** – містить інформацію про всі товари в магазинах | **Id** – *унікальний ідентифікатор товару*  **storebranchId** – *унікальний ідентифікатор відділу магазину*  **name** – *назва товару*  **sellprice** – *ціна товару*  **sold** – *ідентифікатор продажу товару* | **integer** *(числовий)*  **integer** *(числовий)*  **character varying** *(рядок)*  **integer** *(числовий)*  **boolean** *(логічний)* |
| **orders -** містить інформацію про всі покупки товарів в магазинах | **Id** – *унікальний ідентифікатор покупки*  **shopId** – *унікальний ідентифікатор магазину*  **price** – *ціна покупки* | **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)* |
| **product\_orders -**містить унікальні ідентифікатори продуктів та покупок для їх зв’язків | **Id** – *унікальний ідентифікатор зв’язку*  **ordersId -** *унікальний ідентифікатор покупки*  **productId -** *унікальний ідентифікатор товару* | **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)* |

**Функціональні залежності для кожної таблиці**

SHOP:

Id → name, location, age

Id → name (назва магазину залежить від його коду)

Id → location (місце знаходження магазину залежить від його коду)

Id → age (вік магазину залежить від його коду)

EMPLOYEE:

Id → shopId, surname, age, phone

Id → shopId (код магазину залежить від коду робітника)

Id → surname (прізвище робітника залежить від його коду)

Id → age (вік робітника залежить від його коду)

Id → phone (телефонний номер робітника залежить від його коду)

STOREBRANCH:

Id → shopId, name

Id → shopId (код магазину залежить від коду його відділу)

Id → name (ім’я відділу залежить від його коду)

PRODUCT:

Id → storebranchId, name, sellprice, sold

Id → storebranchId (код відділу залежить від коду товару, що в ньому знаходиться)

Id → name (ім’я товару залежить від його коду)

Id → sellprice (ціна товару залежить від його коду)

Id → sold (ідентифікатор продажу товару залежить від його коду)

ORDERS:

Id →shopId, price

Id → shopId (код магазину залежить від коду покупки в ньому)

Id → price (ціна покупки залежить від її коду)

PRODUCT\_ORDERS:

Id → ordersId, productId

Id → ordersId (код покупки залежить від коду її зв’язку)

Id → productId (код продукту залежить від коду його зв’язку)

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

Схема відповідає 1НФ, тому що:

Кожна клітинка таблиць може містити лише одне значення, а кожен новий запис не може повторювати жодний з попередніх бо є унікальним.

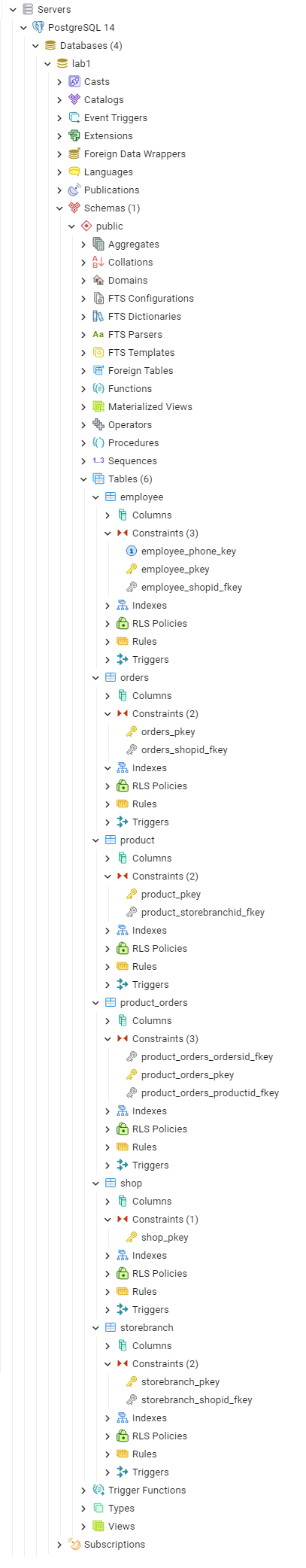
Схема відповідає 2НФ, тому що:

Схема відповідає 1НФ і в схемі відсутні композитні первинні ключі, а отже первинні ключі складаються з одного атрибуту.

Схема відповідає 3НФ, тому що:

Схема відповідає 2НФ і кожен атрибут, що не являє собою первинний ключ, є функціонально залежним від первинного ключа.

**Фізична модель БД «Магазини» у pgAdmin4**

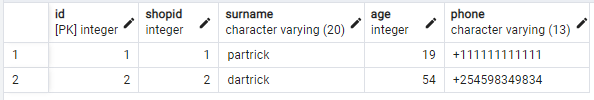


**Фотографії вмісту таблиць**

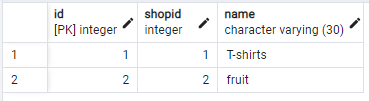
**shop**

****

**employee**

****

**storebranch**

****

**product**



**orders**

****

**product\_orders**



**SQL-текст опису БД «Магазини»**

CREATE TABLE shop

(

Id SERIAL PRIMARY KEY,

name CHARACTER VARYING(30) NOT NULL,

location CHARACTER VARYING(30) DEFAULT 'Kyiv' NOT NULL,

age INTEGER

);

CREATE TABLE employee

(

Id SERIAL PRIMARY KEY,

shopId INTEGER NOT NULL,

surname CHARACTER VARYING(20) NOT NULL,

age INTEGER NOT NULL,

phone CHARACTER VARYING(13) UNIQUE NOT NULL,

FOREIGN KEY (shopId) REFERENCES shop(Id) ON DELETE CASCADE

);

CREATE TABLE storebranch

(

Id SERIAL PRIMARY KEY,

shopId INTEGER NOT NULL,

name CHARACTER VARYING(30) NOT NULL,

FOREIGN KEY (shopId) REFERENCES shop(Id) ON DELETE CASCADE

);

CREATE TABLE product

(

Id SERIAL PRIMARY KEY,

name CHARACTER VARYING(20) NOT NULL,

storebranchId INTEGER NOT NULL,

sellprice INTEGER NOT NULL,

sold BOOL DEFAULT FALSE,

FOREIGN KEY (storebranchId) REFERENCES storebranch(Id) ON DELETE CASCADE

);

CREATE TABLE orders

(

Id SERIAL PRIMARY KEY,

shopId INTEGER NOT NULL,

price INTEGER NOT NULL,

FOREIGN KEY (shopId) REFERENCES shop(Id) ON DELETE CASCADE

);

CREATE TABLE product\_orders

(

Id SERIAL PRIMARY KEY,

ordersId INTEGER NOT NULL,

productId INTEGER NOT NULL,

FOREIGN KEY (ordersId) REFERENCES orders(Id) ON DELETE CASCADE,

FOREIGN KEY (productId) REFERENCES product(Id) ON DELETE CASCADE

);