

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»**

Тема: «***Створити БД "Магазин електроніки" в СУБД PostgreSQL   
з допомогою конструктора PgAdmin 4***»

Виконав: студент 3 курсу

ФПМ групи КВ-83

Гончаров В.С.

Перевірив: Павловський В.І.

Київ – 2020

## **Лабораторна робота №1.**

***Створити БД "Магазин електроніки" в СУБД PostgreSQL   
з допомогою конструктора PgAdmin 4***

***Мета роботи:*** створити БД Магазин електроніки та сформувати обмеження цілісності на значення даних.

***Порядок виконання роботи***

1. Розробити концептуальну модель вибраного предметного середовища. Концептуальну модель предметного середовища Магазин електроніки наводиться в Додатку А до лабораторної роботи;
2. Розробити логічну модель (схему) даних БД. Логічна модель (схема) даних БД Магазин електроніки наводиться в Додатку до лабораторної роботи;
3. Вивчити склад та правила роботи з СУБД PostgreSQL;
4. Створити в СУБД PostgreSQL БД Магазин електроніки, використовуючи конструктори таблиць та стовпчиків (краще колонок). Схема даних БД Магазин електроніки наводиться в Додатку Б до лабораторної роботи. Перелік атрибутів наводиться в Додатку В до лабораторної роботи;
5. Сформувати обмеження цілісності.
6. Фізична модель (схема) даних БД Магазин електроніки наводиться в Додатку до лабораторної роботи;
7. Заповнити створену БД даними (порядку 5-10 записів в кожній таблиці).

###### ***Зміст звіту***

1. Склад СУБД PostgreSQL.
2. Концептуальна модель предметної області.
3. Логічна модель (схема) БД.
4. Склад обмежень цілісності в термінах СУБД PostgreSQL.
5. Фізична модель БД в термінах СУБД PostgreSQL.
6. Представлення БД в pgAdmin 4

**Додаток А. Концептуальна модель предметної області**

**"Магазин"**

В концептуальній моделі предметної області "Магазин" виділяються наступні сутності та зв'язки між ними.

Сутність «**Магазин**» з атрибутами: id, місто, вулиця, будинок;

Сутність «**Продавець**» з атрибутами: id, ім’я, id магазину, часова мітка;

Сутність «**Заказ**» з атрибутами: id, статус, id продавця, часова мітка;

Сутність «**Товар**» з атрибутами: id, ціна, категорія.

Сутність «**Паспорт**» з атрибутами: id, дата народження, місце народження.

Між сутностями «**Магазин**» та «**Продавець**» зв’язок R(1:N), тому що один магазин може мати тільки декілька продавців.

Між сутностями «**Прдавець**» та «**Замовлення**» зв’язок R(1:N), тому що один продавець може оформити декілька заказів.

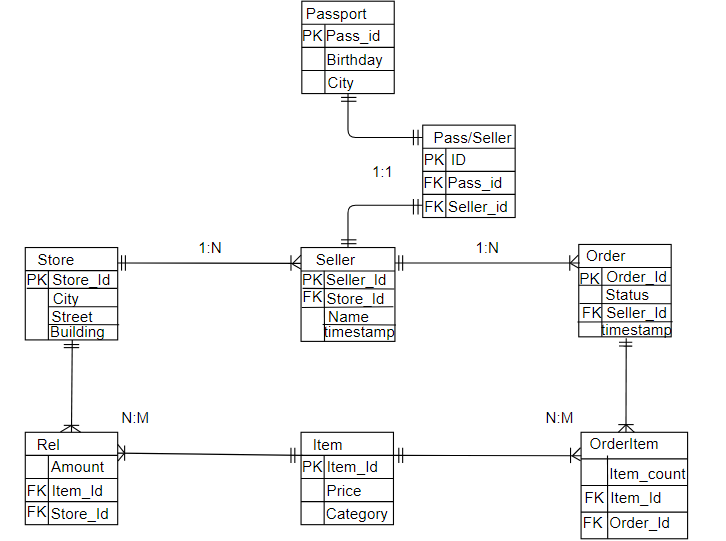
Між сутностями «**Замовлення**» та «**Товар**» зв’язок R(N:M), тому що як один заказ може містити декілька видів товарів, так і один вид товару може бути в декількох заказах.

Між сутностями «**Магазин**» та «**Товар**» зв’язок R(N:M), тому що як один магазин може продавати декілька видів товарів, так і певний вид товару може продаватися в декількох магазинах.

Між сутностям «**Продавець**» та «**Паспорт**» зв’язок R(1:1), тому що кожен продавець може мати лише один паспорт.

## 

Концептуальна модель предметної області



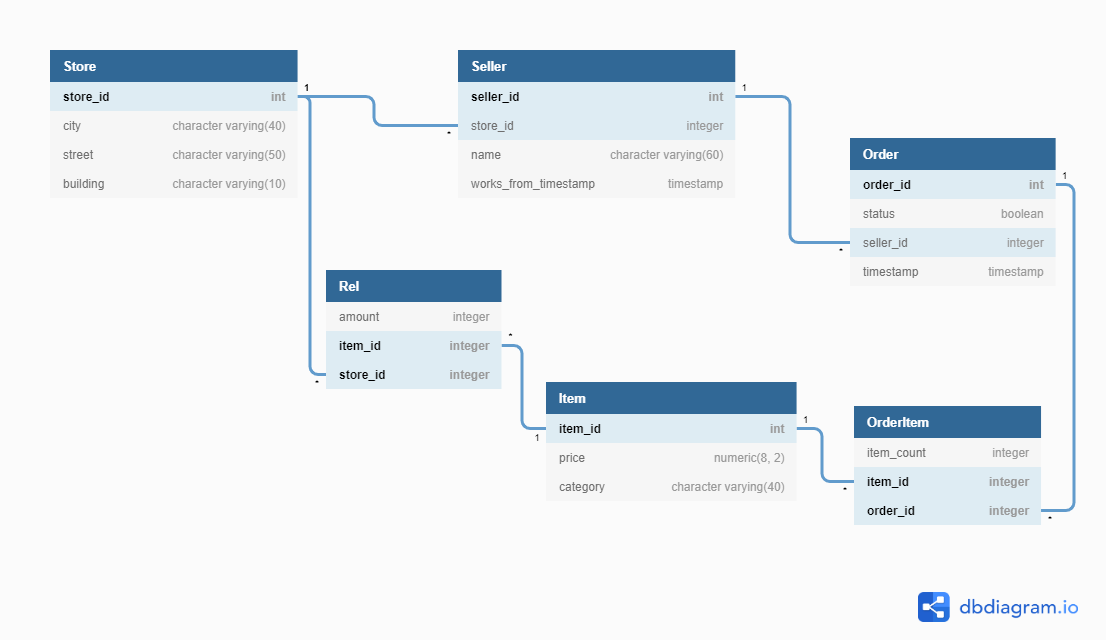
Логічна модель предметної області

## **База даних**«**Магазин електроніки»**

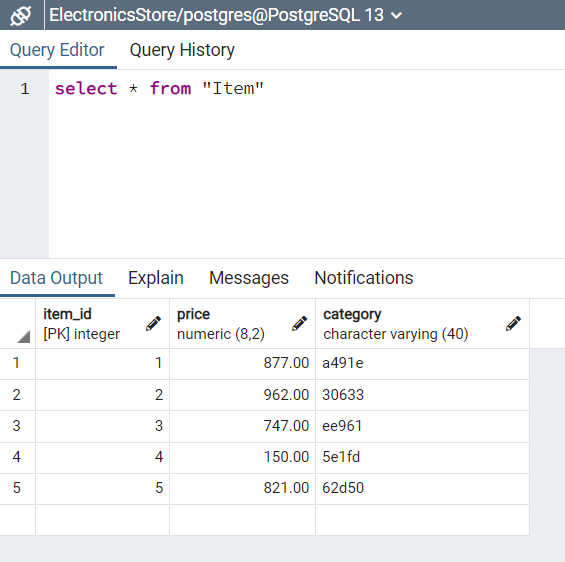
## База даних складається з 6 таблиць (4 сутностей, 2 таблиць-зв'язків)



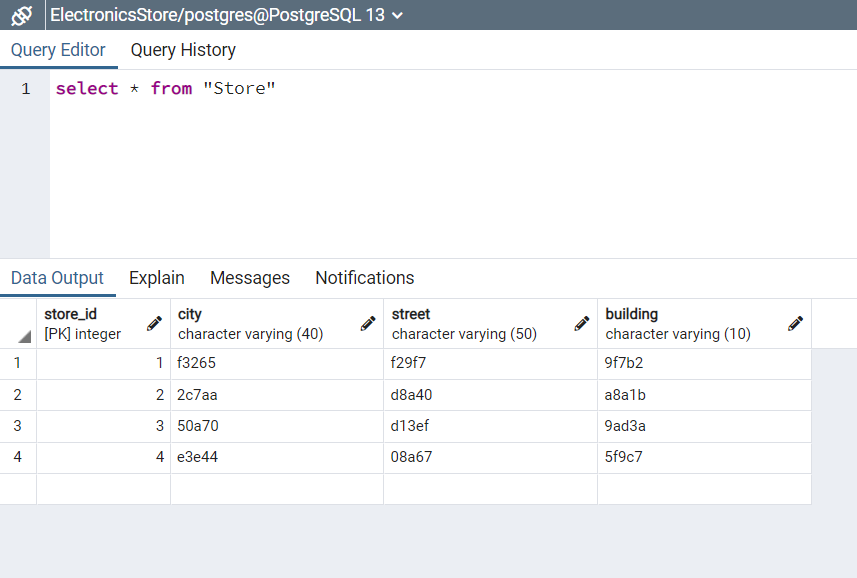
## **UML-діаграма**



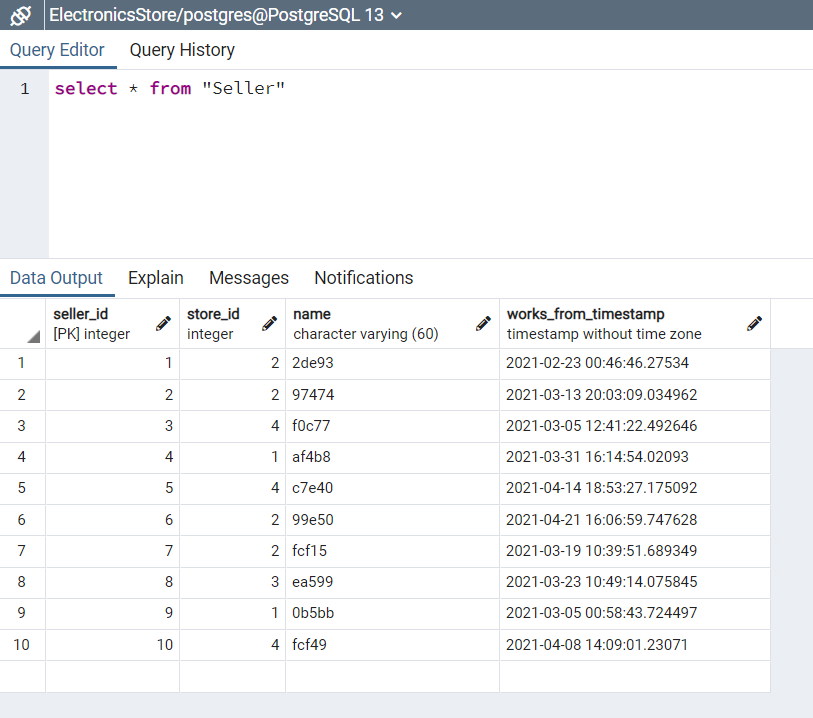
**Вміст таблиці Item**



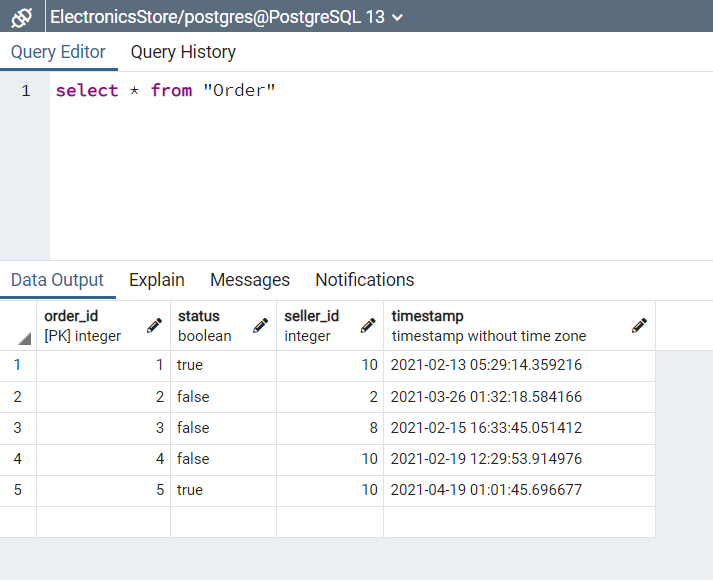
## **Вміст таблиці** **Store**



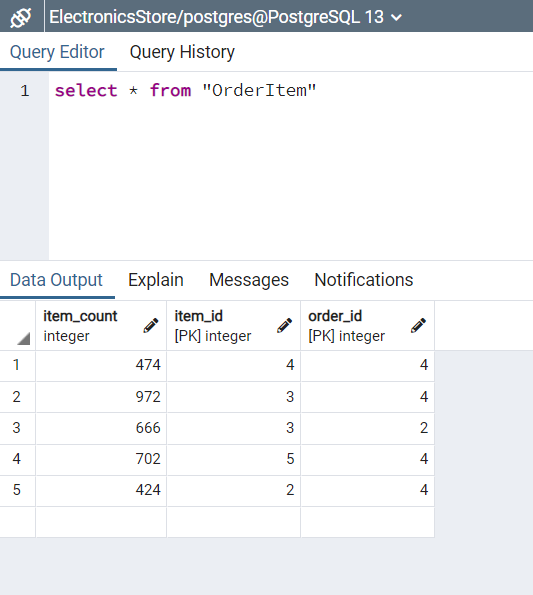
## **Вміст таблиці Seller**



## **Вміст таблиці Order**



**Вміст таблиці OrderItem**



## **Вміст таблиці Rel**

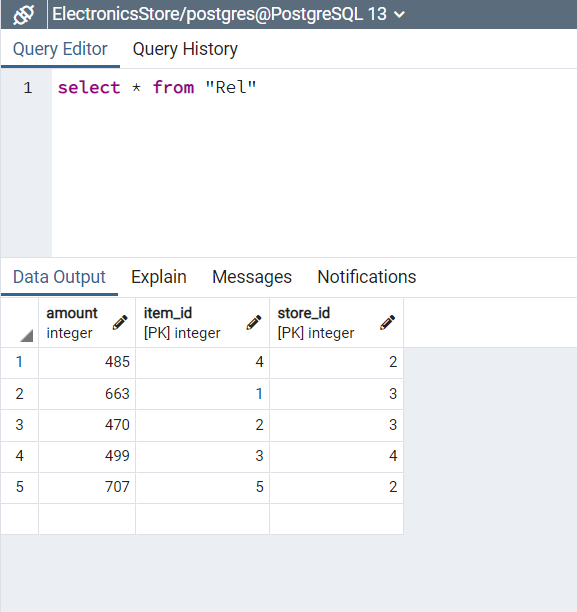


Схема відповідає 1НФ, а саме:

1. Кожна таблиця повинна мати основний ключ (Primary key, PK): мінімальний набір атрибутів, які ідентифікують запис;
2. Кожен атрибут повинен мати лише одне значення, а не множину значень.

Схема відповідає 2НФ, а саме:

1. Схема повинна відповідати вимогам 1НФ;
2. Атрибути повинні повністю залежати від обох первинних атрибутів, а не від якогось окремого.

Схема відповідає 3НФ, а саме:

1. Схема повинна відповідати вимогам 2НФ;
2. Дані в таблиці повинні залежати винятково від основного ключа.

## **Код SQL**

## **"Item" table (Info about electronic devices)**

CREATE TABLE public."Item"

(

item\_id SERIAL,

price numeric(8,2),

category character varying(40) COLLATE pg\_catalog."default",

CONSTRAINT "Item\_pkey" PRIMARY KEY (item\_id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."Item"

OWNER to postgres;

## **"Store" table (Stores location info)**

CREATE TABLE public."Store"

(

store\_id SERIAL,

city character varying(40) COLLATE pg\_catalog."default",

street character varying(50) COLLATE pg\_catalog."default",

building character varying(10) COLLATE pg\_catalog."default",

CONSTRAINT "Store\_pkey" PRIMARY KEY (store\_id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."Store"

OWNER to postgres;

## **"Seller" table (Basic info about sellers)**

CREATE TABLE public."Seller"

(

seller\_id SERIAL,

store\_id integer,

name character varying(60) COLLATE pg\_catalog."default",

works\_from\_timestamp timestamp without time zone,

CONSTRAINT "Seller\_pkey" PRIMARY KEY (seller\_id),

CONSTRAINT "FK\_StoreSeller" FOREIGN KEY (store\_id)

REFERENCES public."Store" (store\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."Seller"

OWNER to postgres;

## **"Order" table (Info about transactions)**

CREATE TABLE public."Order"

(

order\_id SERIAL,

status boolean,

seller\_id integer,

"timestamp" timestamp without time zone,

CONSTRAINT "Order\_pkey" PRIMARY KEY (order\_id),

CONSTRAINT "FK\_SellerOrder" FOREIGN KEY (seller\_id)

REFERENCES public."Seller" (seller\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."Order"

OWNER to postgres;

## "OrderItem" table (Links **"Item"** and **"Order")**

CREATE TABLE public."OrderItem"

(

item\_count integer,

item\_id integer NOT NULL,

order\_id integer NOT NULL,

CONSTRAINT "OrderItem\_pkey" PRIMARY KEY (item\_id, order\_id),

CONSTRAINT "FK\_ItemOrderItem" FOREIGN KEY (item\_id)

REFERENCES public."Item" (item\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

NOT VALID,

CONSTRAINT "FK\_OrderOrderItem" FOREIGN KEY (order\_id)

REFERENCES public."Order" (order\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."OrderItem"

OWNER to postgres;

## **"Rel" table (Links "Item"** **and "Store")**

CREATE TABLE public."Rel"

(

amount integer,

item\_id integer NOT NULL,

store\_id integer NOT NULL,

CONSTRAINT "Rel\_pkey" PRIMARY KEY (item\_id, store\_id),

CONSTRAINT "FK\_ItemRel" FOREIGN KEY (item\_id)

REFERENCES public."Item" (item\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

NOT VALID,

CONSTRAINT "FK\_StoreRel" FOREIGN KEY (store\_id)

REFERENCES public."Store" (store\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE CASCADE

NOT VALID

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public."Rel"

OWNER to postgres;