**Лабораторна робота № 1**

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

***з дисципліни***

**“БАЗИ ДАНИХ ТА ЗАСОБИ УПРАВЛІННЯ”**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Група: КВ-01**

**Виконала: Бондарчук Марія**

**Лабораторна робота № 1**

**Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL**

*Метою роботи* є здобуття вмінь проектування бази даних та 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.

**Опис предметної області «Пекарня»**

Обрана предметна область забезпечує адміністрування та функціонування пекарні. Заклад має декілька пекарів з визначеними зарплатами. Пекарі працюють з кількома рецептами. Різні рецепти передбачають різний час на їх приготування.

**Опис сутностей предметної області:**

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

1. Власник (Owner), з атрибутами:

* Ідентифікаційний номер
* Ім’я

1. Пекарня (Bakery), з атрибутами:

* Ідентифікаційний номер
* Ідентифікаційний номер власника
* Назва
* Адреса

1. Пекар (Baker), з атрибутами:

* Ідентифікаційний номер
* Ідентифікаційний номер пекарні
* Ім’я
* Зарплатня

1. Рецепт (Recipe), з атрибутами:

* Ідентифікаційний номер
* Ідентифікаційний номер пекарні
* Ім’я
* Інгредієнти

1. Продукт (Product), з атрибутами:

* Ідентифікаційний номер продукту
* Ідентифікаційний номер пекарні
* Ідентифікаційний номер рецепту

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

1) Сутність **Bakery** було перетворено в таблицю з назвою **Bakery,** яка має зв’язок 1 : N з **Owner**, а також 1 : N з таблицями **Baker** та **Recipe**, для всіх є головною.

2)Сутність **Owner** було перетворено в таблицю з назвою **Owner,** яка має зв’язок 1 : N з таблицею **Bakery**, є залежною від **Bakery**.

3)Сутність **Recipe** було перетворено в таблицю з назвою **Recipe,** яка має зв’язок 1 : N з таблицею **Bakery**, є залежною від **Bakery** та реалізує зв’язок N : M з таблицею **Baker**, за допомогою таблиці **Product**.

4)Сутність **Baker** було перетворено в таблицю з назвою **Baker,** яка має зв’язок 1 : N з таблицею **Bakery**, є залежною від **Bakery** а також реалізує зв’язок N : M з таблицею **Recipe**, за допомогою таблиці **Product**.

5)Сутність **Product** було перетворено в таблицю з назвою **Product,** яка має зв’язок 1 : N з таблицями **Baker** та **Recipe** і є залежною від них.

**Концептуальна модель предметної області “Пекарня”**

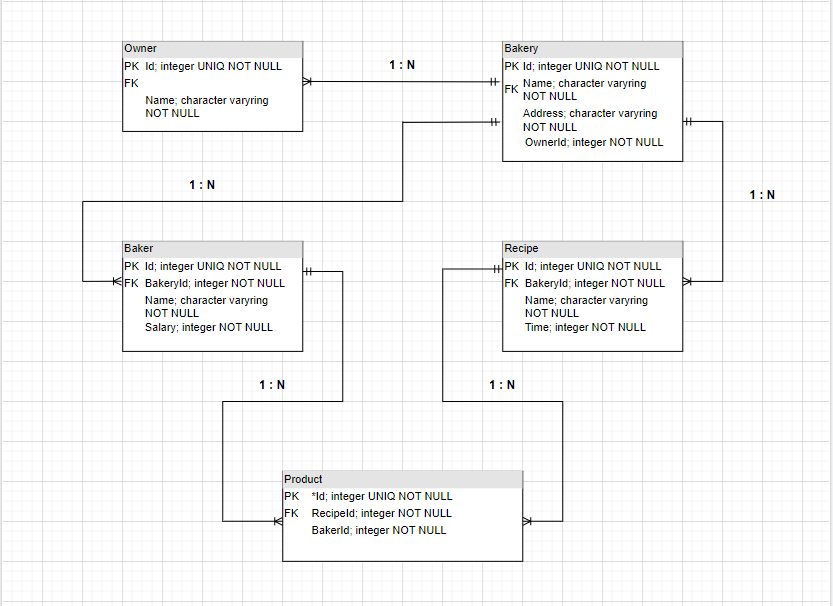


Рисунок 1 - ER-діаграма, Нотація Пташиної лапки (Crow’s foot)”

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

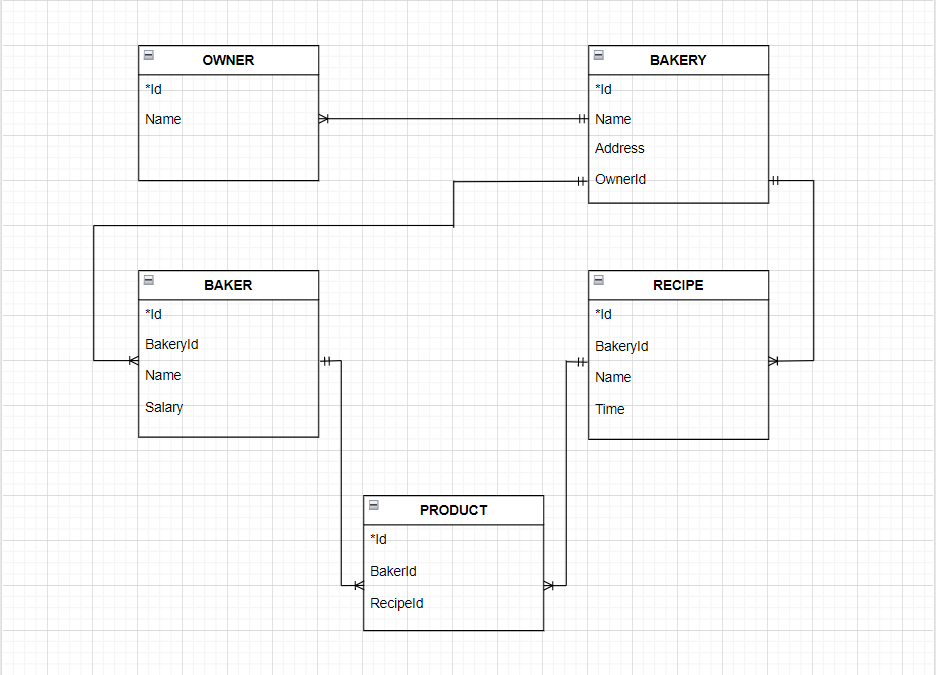
Сутність «Власник» перетворено у таблицю «Owner».

Сутність «Пекарня» перетворено у таблицю «Bakery», а зв’язок 1:N з сутністю «Власник» зумовив появу зовнішнього ключа OwnerId

Сутність «Пекар» перетворено у таблицю «Baker», а зв’язок 1:N з сутністю «Пекарня» зумовив появу зовнішнього ключа BakeryId

Сутність «Рецепт» перетворено у таблицю «Recipe» », а зв’язок 1:N з сутністю «Пекарня» зумовив появу зовнішнього ключа BakeryId

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

  
Рисунок 2 - Схема бази даних

**Опис об’єктів бази даних у вигляді таблиці**

|  |  |  |
| --- | --- | --- |
| **СУТНІСТЬ** | **АТРИБУТ** | **ТИП АТРИБУТУ** |
| **OWNER** - містить інформацію про власника | **Id** - унікальний ідентифікатор власника  **Name** – ім’я | **integer**  **character varyring** |
| **BAKERY** - містить інформацію про пекарню | **Id** - унікальний ідентифікатор пекарні  **OwnerId** – ідентифікатор власника  **Name** - назва  **Address** - адреса пекарні | **integer**  **integer**  **character varyring**  **character varyring** |
| **BAKER** - містить інформацію про пекаря | **Id** - унікальний ідентифікатор пекаря  **BakeryId** - ідентифікатор пекарні  **Name** - ім’я  **Salary** - зарплатня пекаря | **integer**  **integer**  **character varyring**  **integer** |
| **RECIPE** - містить інформацію про рецепт | **Id** - унікальний ідентифікатор рецепту  **BakeryId** - ідентифікатор пекарні  **Name** - назва рецепту  **Time** – час на приготування | **integer**  **integer**  **character varyring**  **integer** |
| **PRODUCT** - містить інформацію про готовий продукт | **Id** - унікальний ідентифікатор продукту  **BakerId** - ідентифікатор пекаря  **RecipeId** - ідентифікатор рецепту | **integer**  **integer**  **integer** |

**Відповідність схеми бази даних нормальним формам:**

**НФ1:**

* кожна «комірка» відношення містить одне значення.
* кожен запис є унікальним.

**НФ2:**

* таблиця в НФ1
* кожен неключовий атрибут функціонально залежати від усього ключа, а не від його частини

**НФ3:**

* таблиця в НФ2
* всі атрибути залежать від первинного ключа, а не від інших атрибутів

**Функціональні залежності:**

**OWNER:**

Id → Name

Id → Name (ім’я власника зв’язане з унікальним ідентифікатором власника)

**BAKERY:**

Id → Name, Address

Id → Name (ім’я пекарні зв’язане з її унікальним ідентифікатором)

Id → Address (адреса пекарні зв’язана з її унікальним ідентифікатором)

Name → Address

Id → Name→ Address (всі атрибути залежать від первинного ключа, а не від інших атрибутів)

Address → Name

Id → Address→ Name (всі атрибути залежать від первинного ключа, а не від інших атрибутів)

**BAKER:**  
Id → Name, Salary

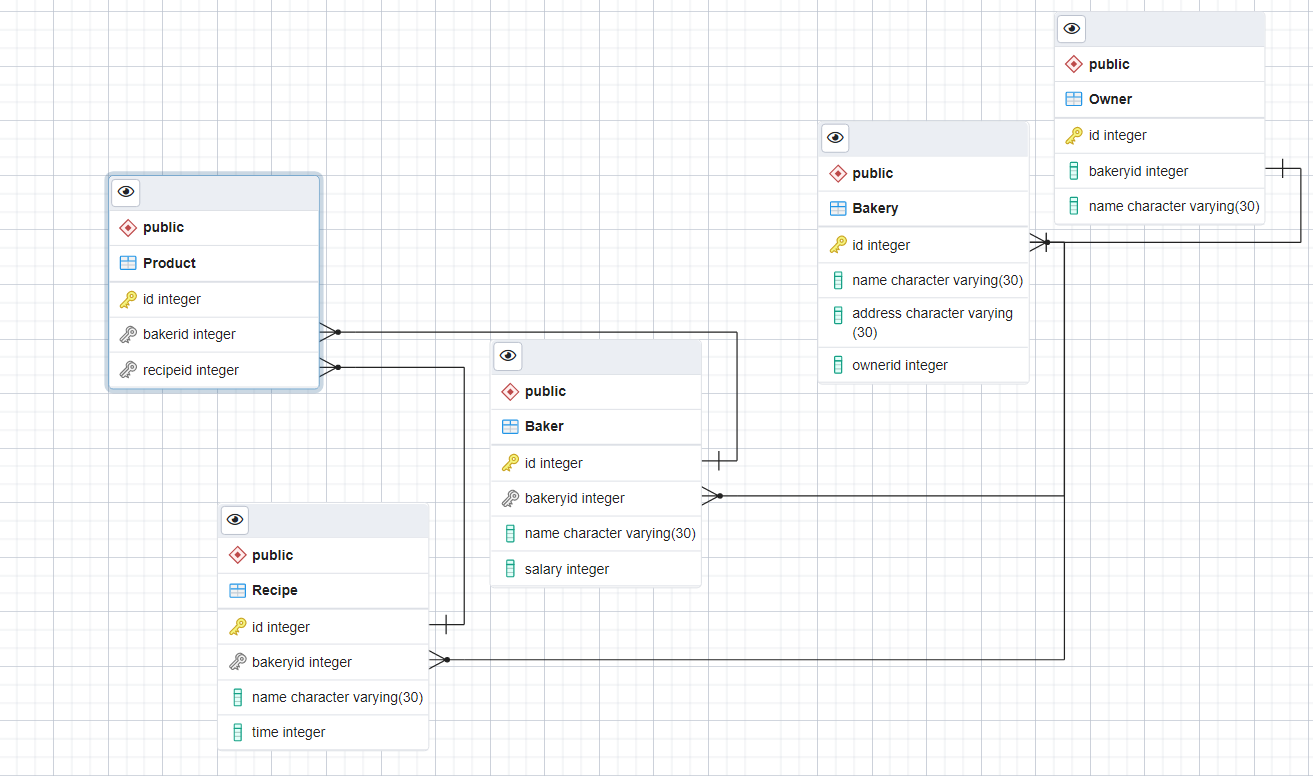
Id → Name (ім’я пекаря зв’язане з його унікальним ідентифікатором)

Id → Salary (зарплата пекаря зв’язана з її унікальним ідентифікатором)

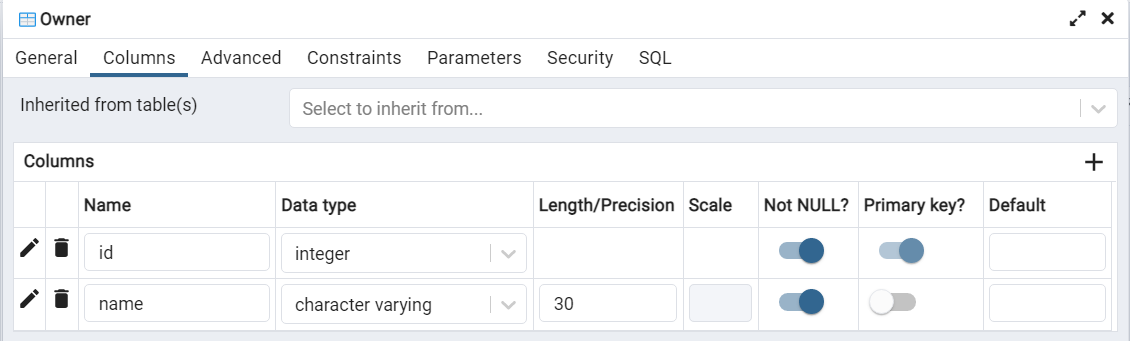
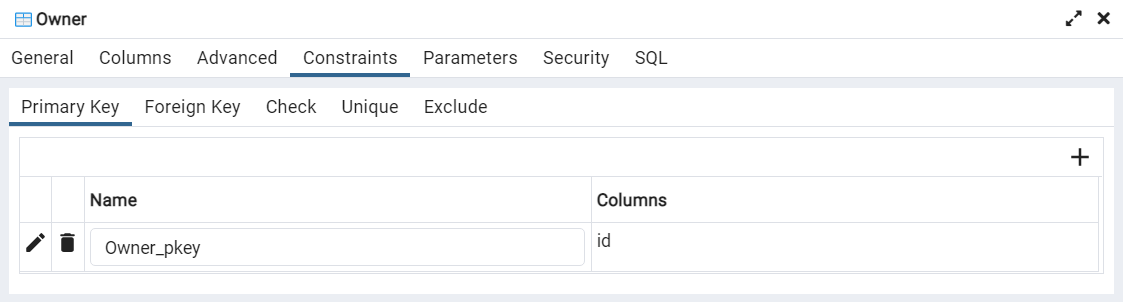
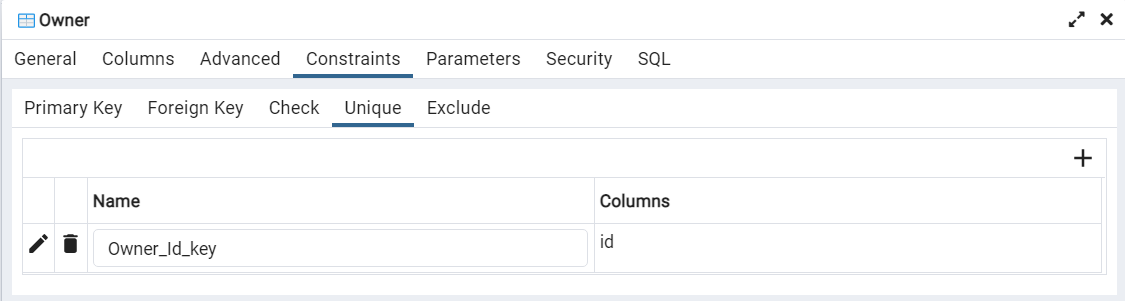
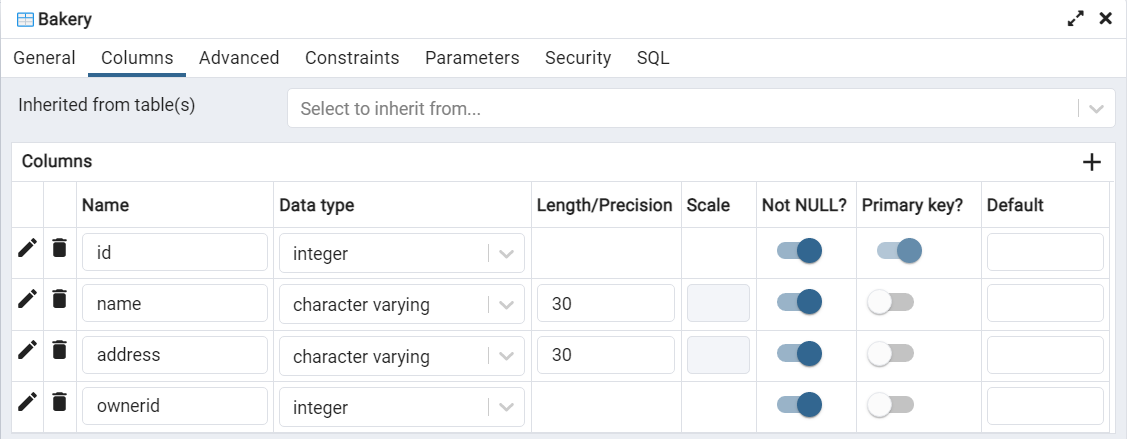
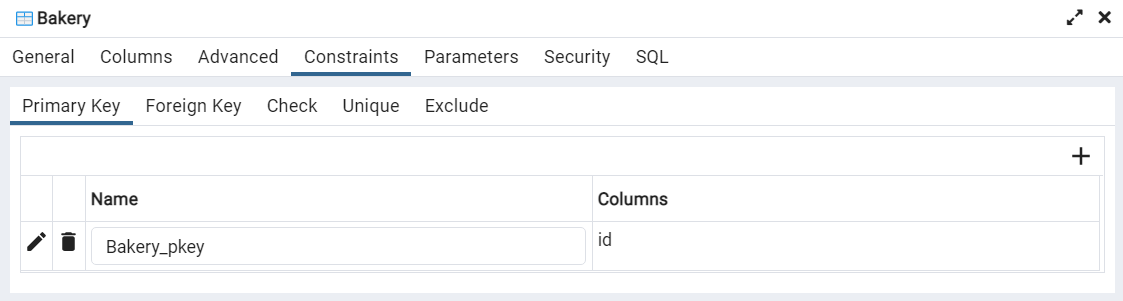
**RECIPE:**

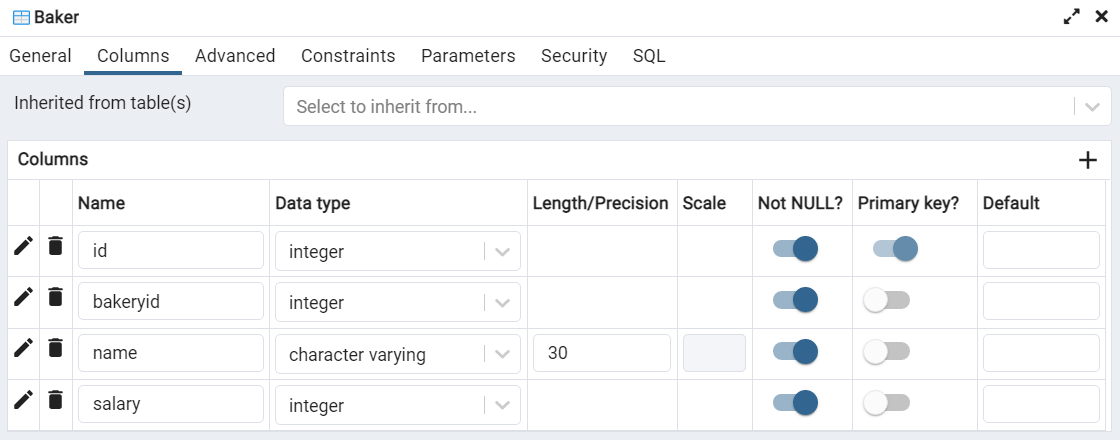
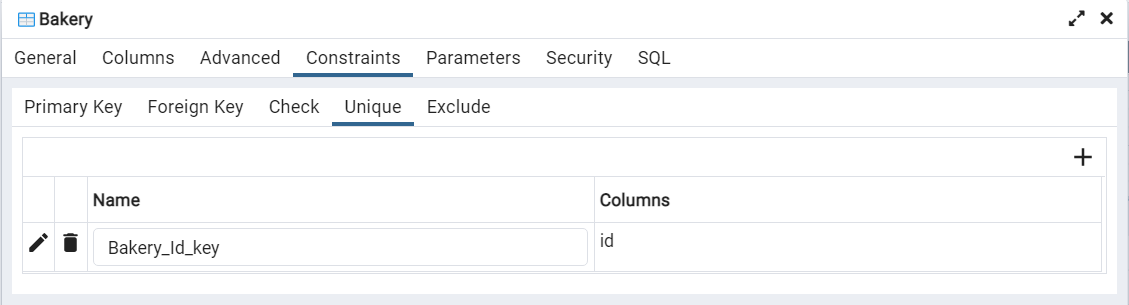
Id → Name, Time

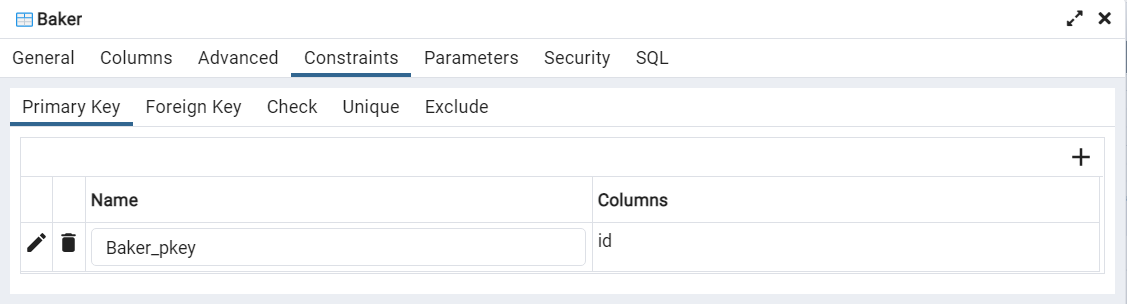
Id → Name (назва рецепту зв’язане з його унікальним ідентифікатором)

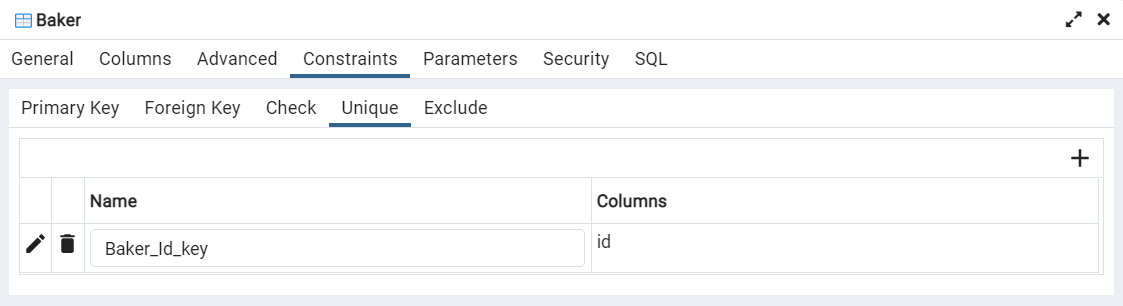
Id → Time (час на виконання рецепту зв’язаний з його унікальним ідентифікатором)  
  
  
  
  
  
**Фізична модель БД «Пекарня» у pgAdmin4**

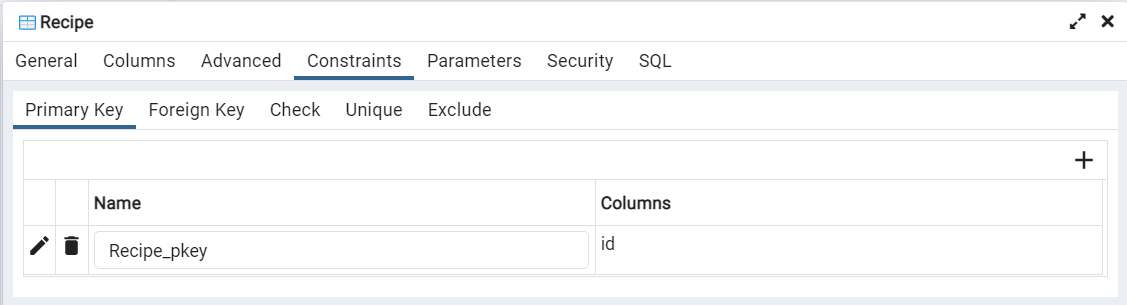
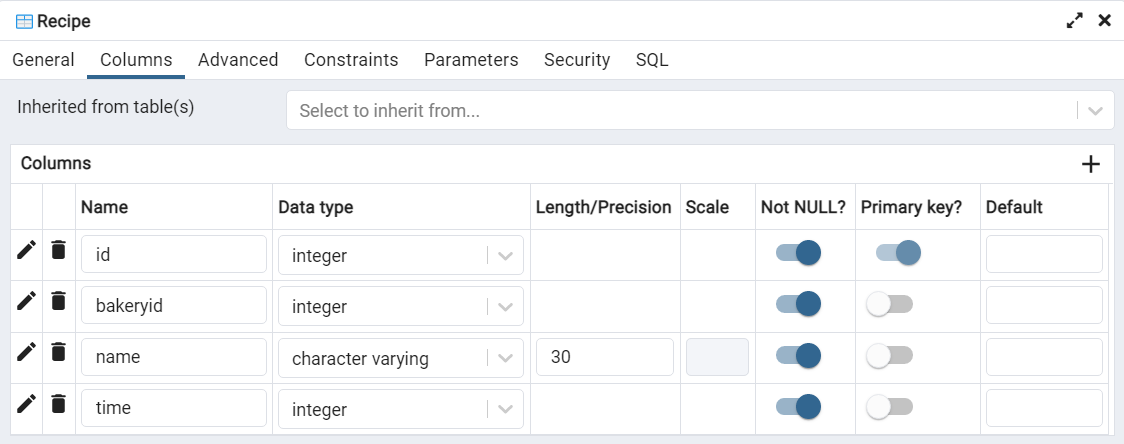
**Скріншоти з pgAdmin4:**

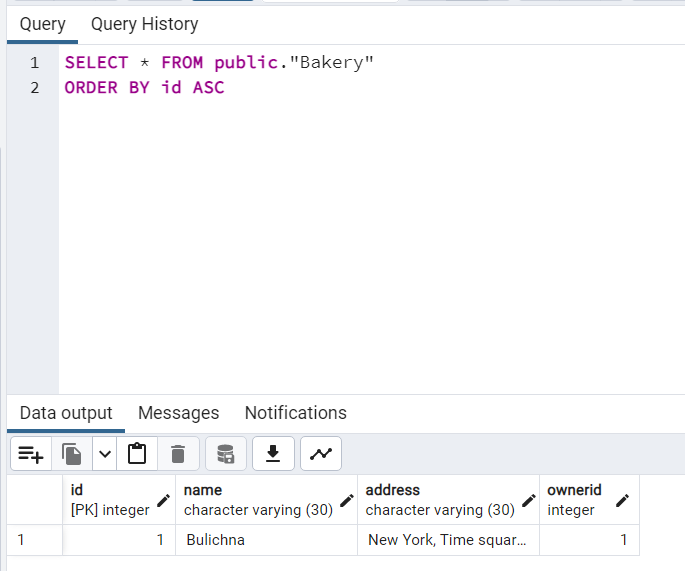
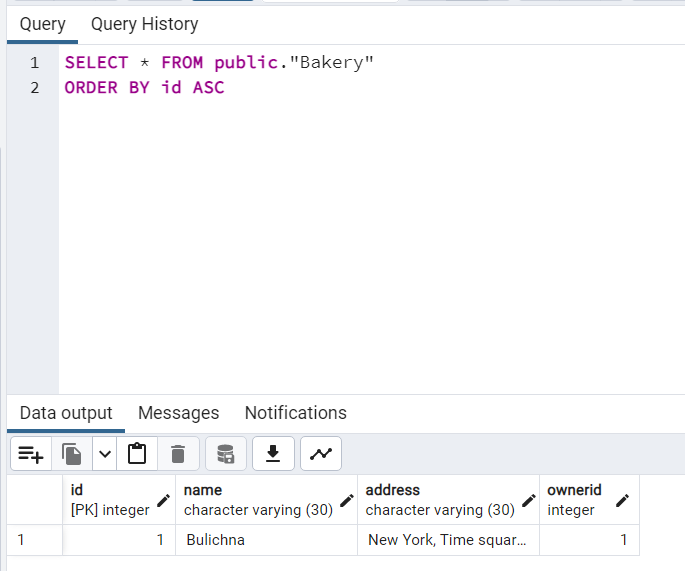
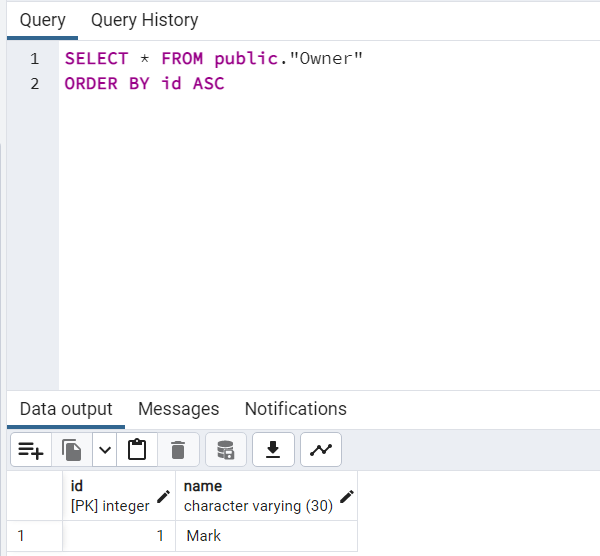
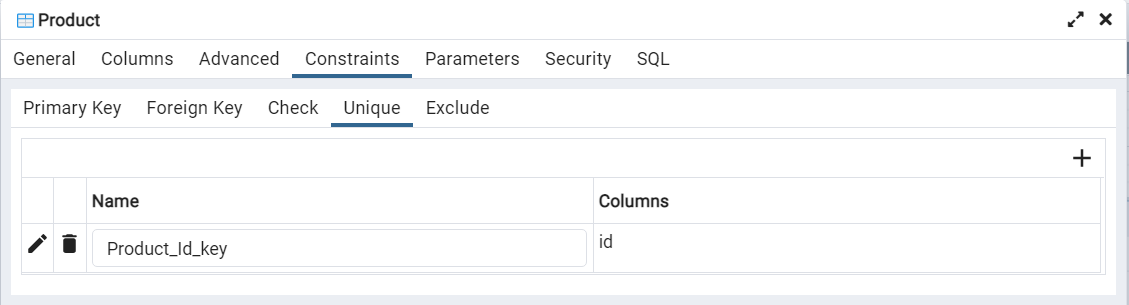
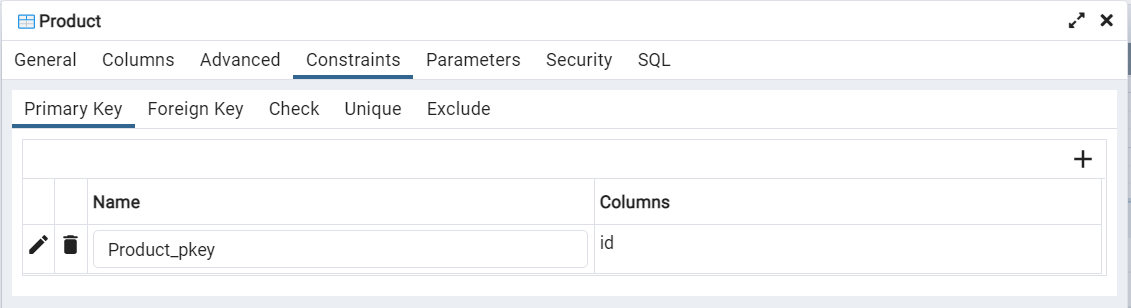
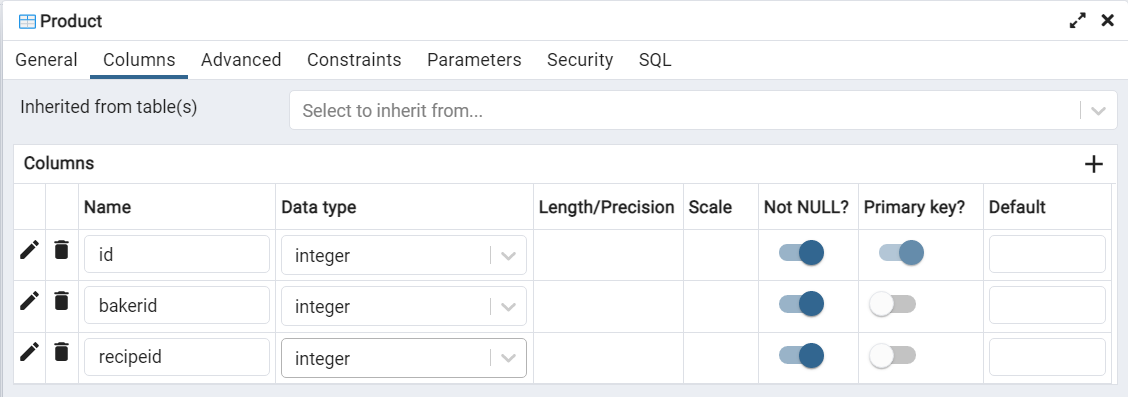
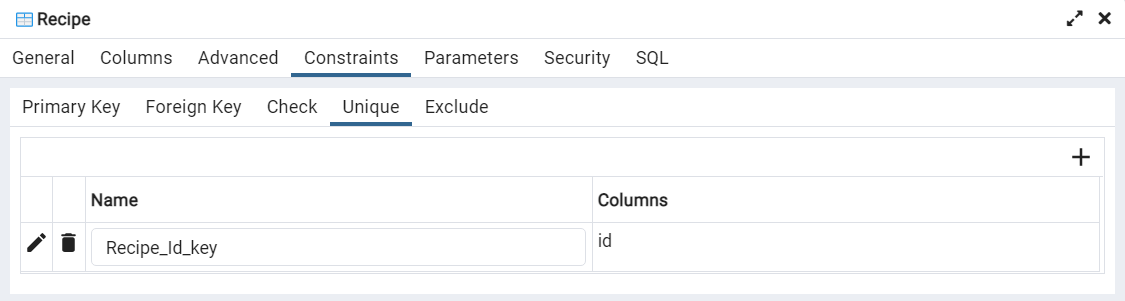
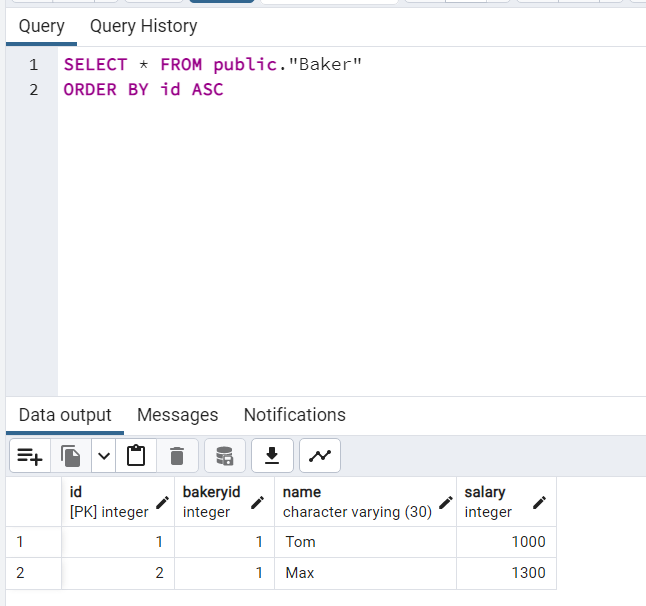
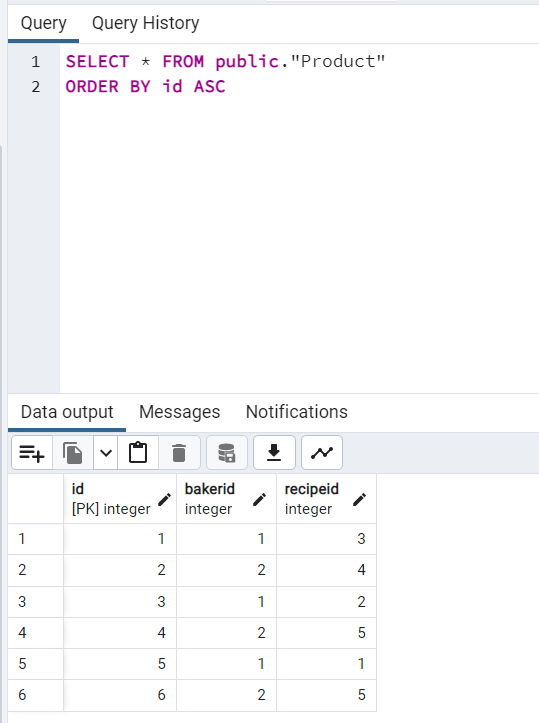
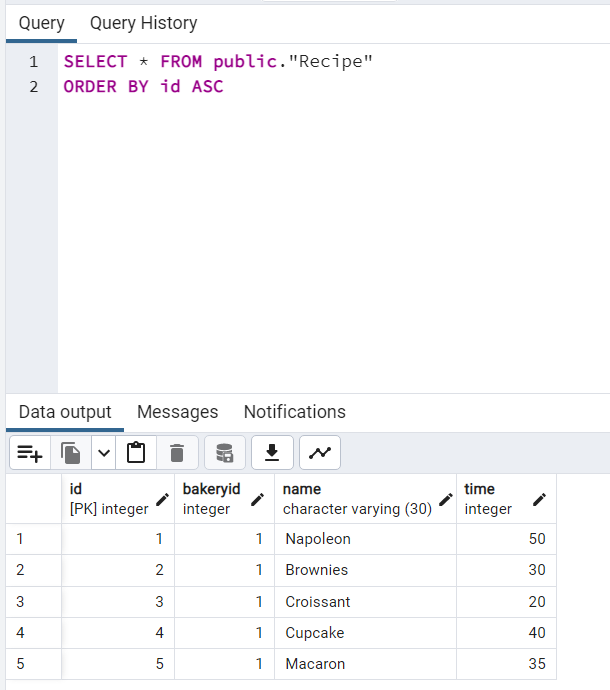
  
  
  
  
  
  
  
  
  
  
  
  










**SQL-текст опису БД «Пекарня»**

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

CREATE EXTENSION IF NOT EXISTS adminpack WITH SCHEMA pg\_catalog;

COMMENT ON EXTENSION adminpack IS 'administrative functions for PostgreSQL';

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

CREATE TABLE public."Baker" (

id integer NOT NULL,

bakeryid integer NOT NULL,

name character varying(30) NOT NULL,

salary integer NOT NULL

);

ALTER TABLE public."Baker" OWNER TO postgres;

CREATE TABLE public."Bakery" (

id integer NOT NULL,

name character varying(30) NOT NULL,

address character varying(30) NOT NULL,

ownerid integer NOT NULL

);

ALTER TABLE public."Bakery" OWNER TO postgres;

CREATE TABLE public."Owner" (

id integer NOT NULL,

name character varying(30) NOT NULL

);

ALTER TABLE public."Owner" OWNER TO postgres;

CREATE TABLE public."Product" (

id integer NOT NULL,

bakerid integer NOT NULL,

recipeid integer NOT NULL

);

ALTER TABLE public."Product" OWNER TO postgres;

CREATE TABLE public."Recipe" (

id integer NOT NULL,

bakeryid integer NOT NULL,

name character varying(30) NOT NULL,

"time" integer NOT NULL

);

ALTER TABLE public."Recipe" OWNER TO postgres;

ALTER TABLE ONLY public."Baker"

ADD CONSTRAINT "Baker\_Id\_key" UNIQUE (id);

ALTER TABLE ONLY public."Baker"

ADD CONSTRAINT "Baker\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Bakery"

ADD CONSTRAINT "Bakery\_Id\_key" UNIQUE (id);

ALTER TABLE ONLY public."Bakery"

ADD CONSTRAINT "Bakery\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Owner"

ADD CONSTRAINT "Owner\_Id\_key" UNIQUE (id);

ALTER TABLE ONLY public."Owner"

ADD CONSTRAINT "Owner\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Product"

ADD CONSTRAINT "Product\_Id\_key" UNIQUE (id);

ALTER TABLE ONLY public."Product"

ADD CONSTRAINT "Product\_pkey" PRIMARY KEY (id);

ALTER TABLE ONLY public."Recipe"

ADD CONSTRAINT "Recipe\_Id\_key" UNIQUE (id);

ALTER TABLE ONLY public."Recipe"

ADD CONSTRAINT "Recipe\_pkey" PRIMARY KEY (id);

**Посилання на GitHub:**

[**https://github.com/mariabondarchuk/DB\_Lab\_1**](#_top)