Катрич Владислав Сергійович

КП-93

**Лабораторна робота № 1.**

*Завдання* роботи полягає у наступному:

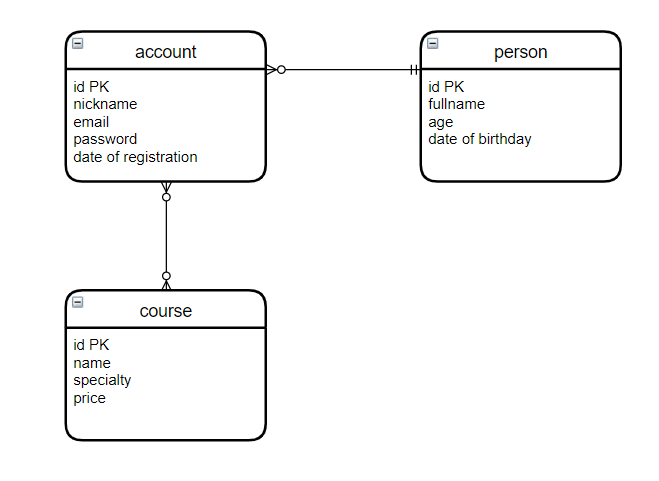
1. Розробити модель «сутність-зв’язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

**1)**

Сутність людина, людина може мати декілька акаунтів.

Сутність акаунт, у акаунта мусить бути користувач та можуть бути курси.

Сутність курс, у курсу можуть бути акаунти(ті, які вирішили підписатись на цей курс).



ER модель.

Назва нотації: crow’s feet.

**2)**

Оскільки у одного акаунту може бути декілька курсів, та у одного курса можуть бути декілька користувачів, то виникає зв’язок many to many і відповідно було створено нову таблицю account\_course\_links для того, щоб зберігати зв’язки між цими таблицями.

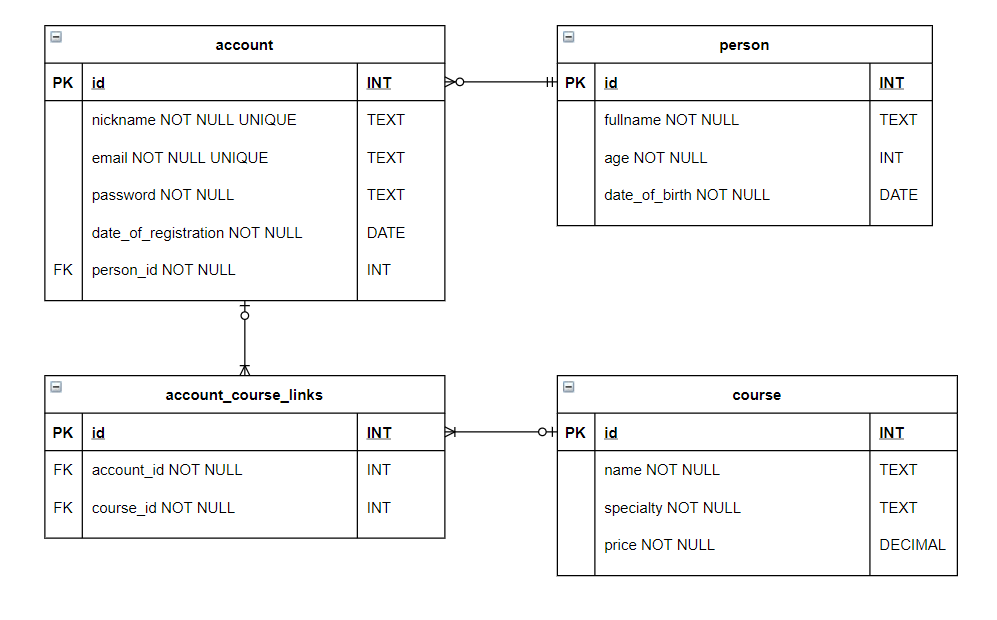


Схема бази даних

**3)**

Схема бд відповідає 1 нормальній формі, так як кожне поле містить лише одне значення.

2 нормальній формі, так як кожна таблиця має id, і інші поля таблиць ніяк від нього не залежать.

3 нормальній формі, так як всі дані, які можуть мати різні таблиці винесені в окремі таблиці.

INSERT INTO person (fullname, age, date\_of\_birth, account\_id) VALUES ('gfrrgr', 22, '24/04/2020', 1);

-- INSERT INTO account (nickname, email, password, date\_of\_registration) VALUES ('nick', 'gmail@gmail.com', '777', '24/04/1111');

**4)**

Створення таблиць:

CREATE TABLE person

(

id INT GENERATED ALWAYS AS IDENTITY,

fullname TEXT NOT NULL,

age INT NOT NULL,

date\_of\_birth DATE NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE account (

id INT GENERATED ALWAYS AS IDENTITY,

nickname TEXT NOT NULL,

email TEXT NOT NULL,

password TEXT NOT NULL,

date\_of\_registration DATE NOT NULL,

person\_id INT NOT NULL,

PRIMARY KEY(id),

CONSTRAINT account\_email\_nickname\_key UNIQUE (email, nickname),

CONSTRAINT fk\_person FOREIGN KEY(person\_id) REFERENCES person(id)

);

CREATE TABLE course (

id INT GENERATED ALWAYS AS IDENTITY,

name TEXT NOT NULL,

specialty TEXT NOT NULL,

price DECIMAL NOT NULL,

PRIMARY KEY(id)

);

CREATE TABLE account\_course\_links (

id INT GENERATED ALWAYS AS IDENTITY,

account\_id INT NOT NULL,

course\_id INT NOT NULL,

PRIMARY KEY(id),

CONSTRAINT fk\_account

FOREIGN KEY(account\_id)

REFERENCES account(id),

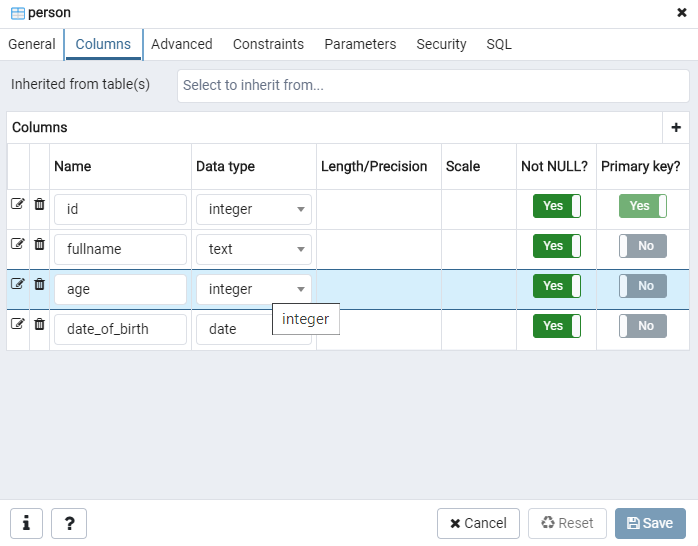
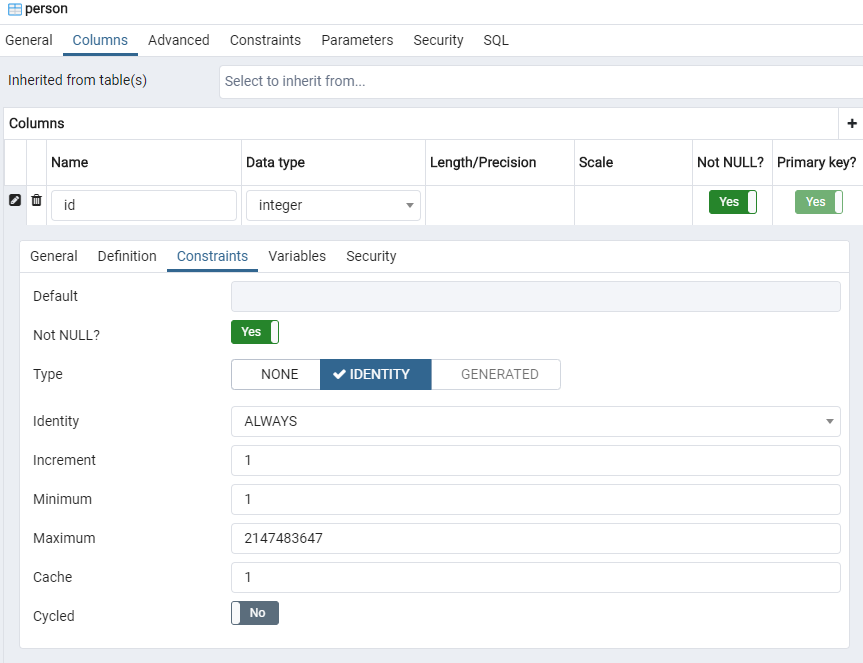
CONSTRAINT fk\_course

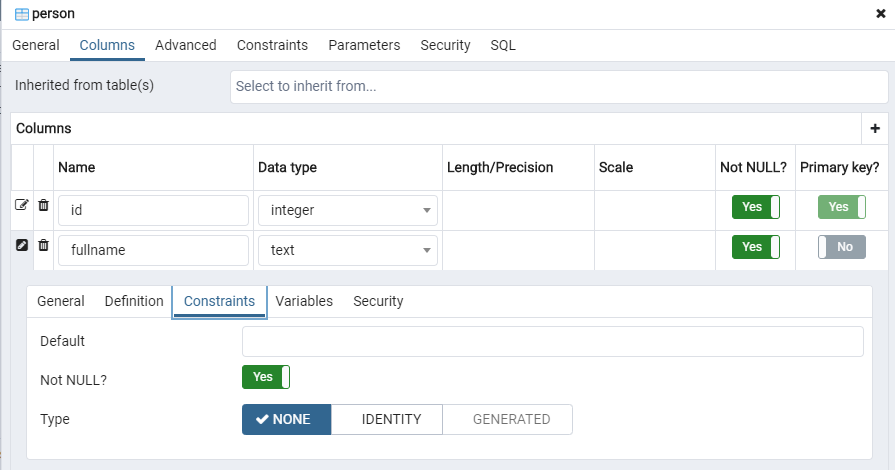
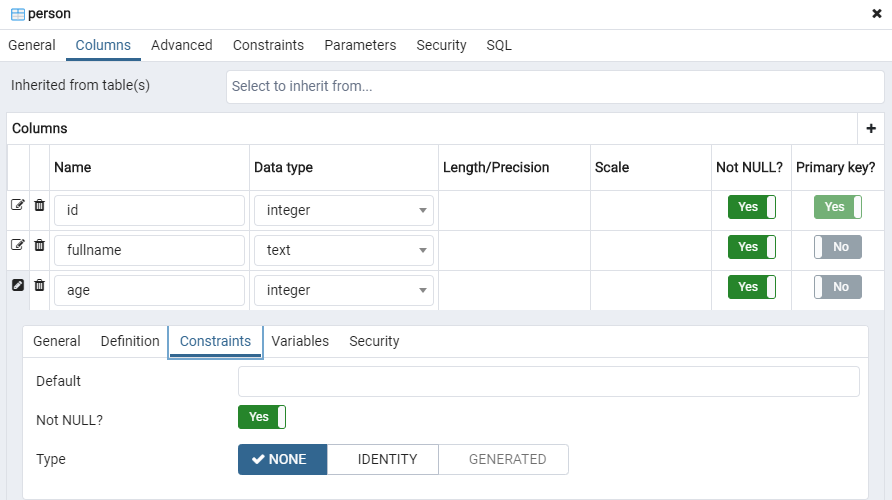
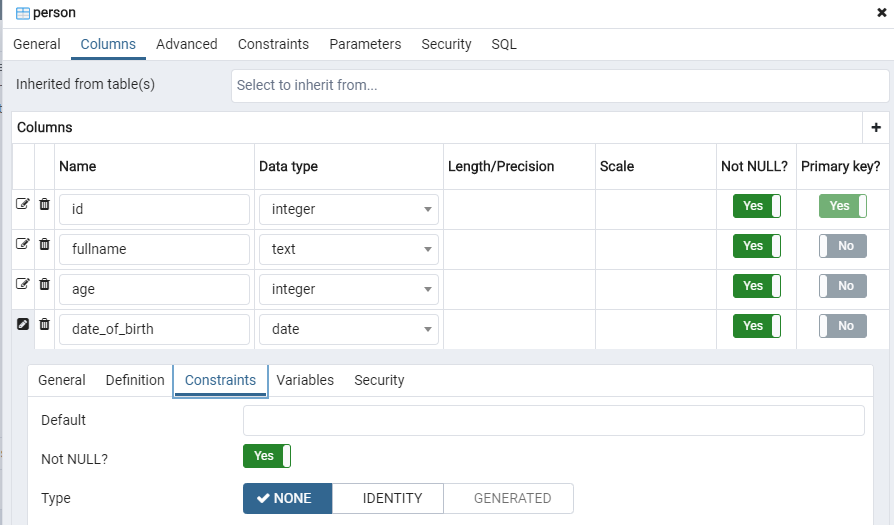
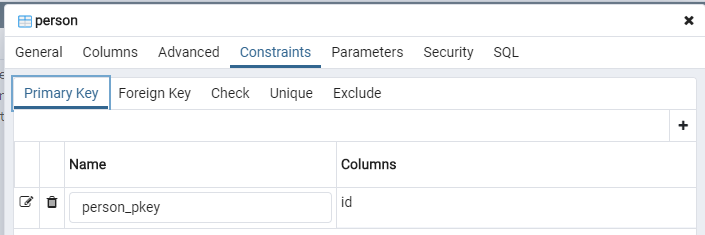
FOREIGN KEY(course\_id)

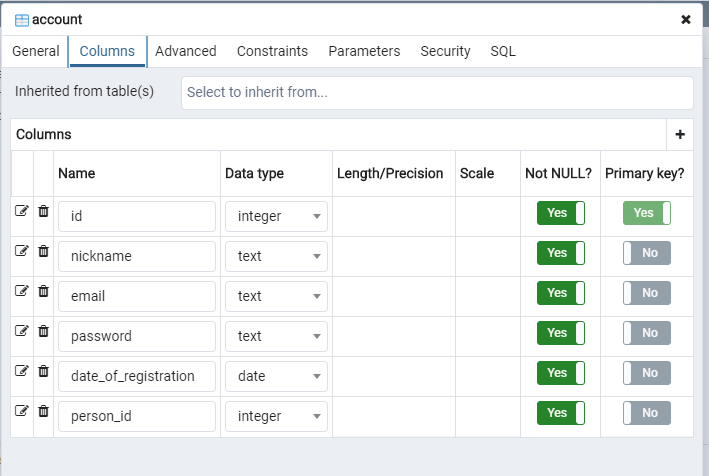
REFERENCES course(id)

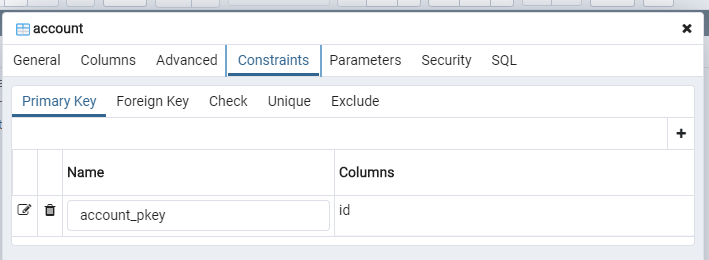
);

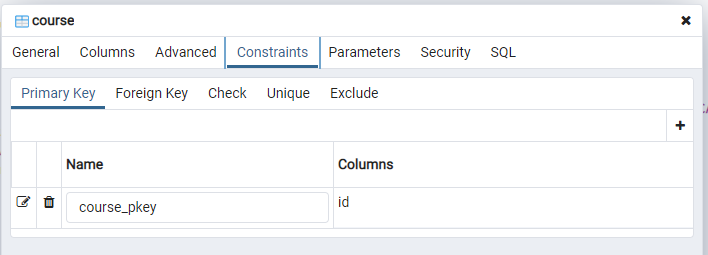
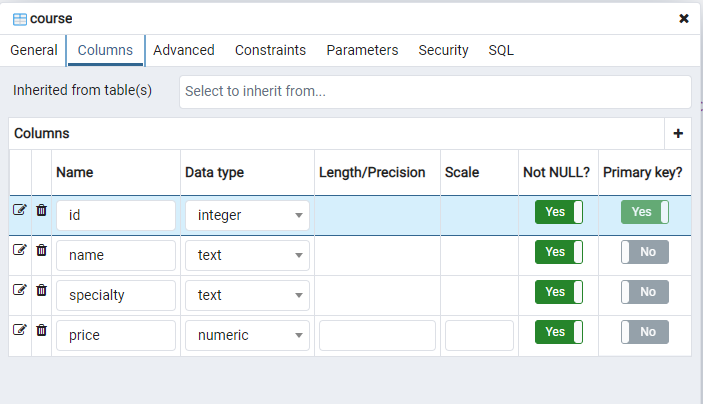
Скріншоти з полями таблиць

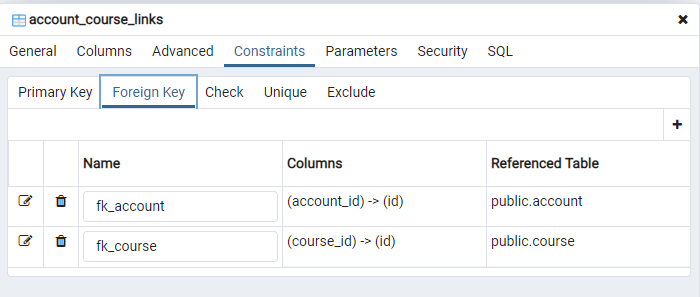
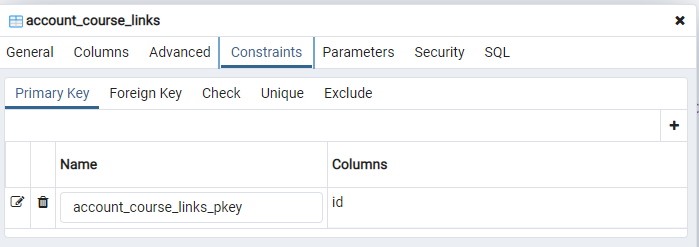
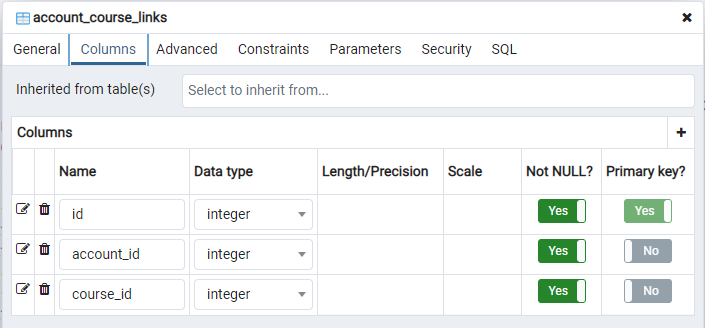
 

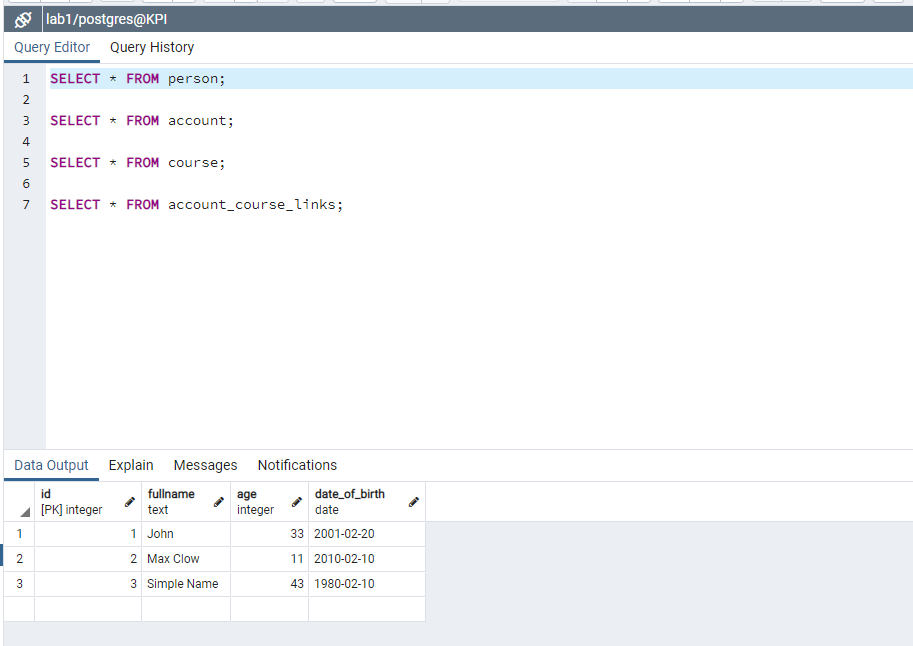


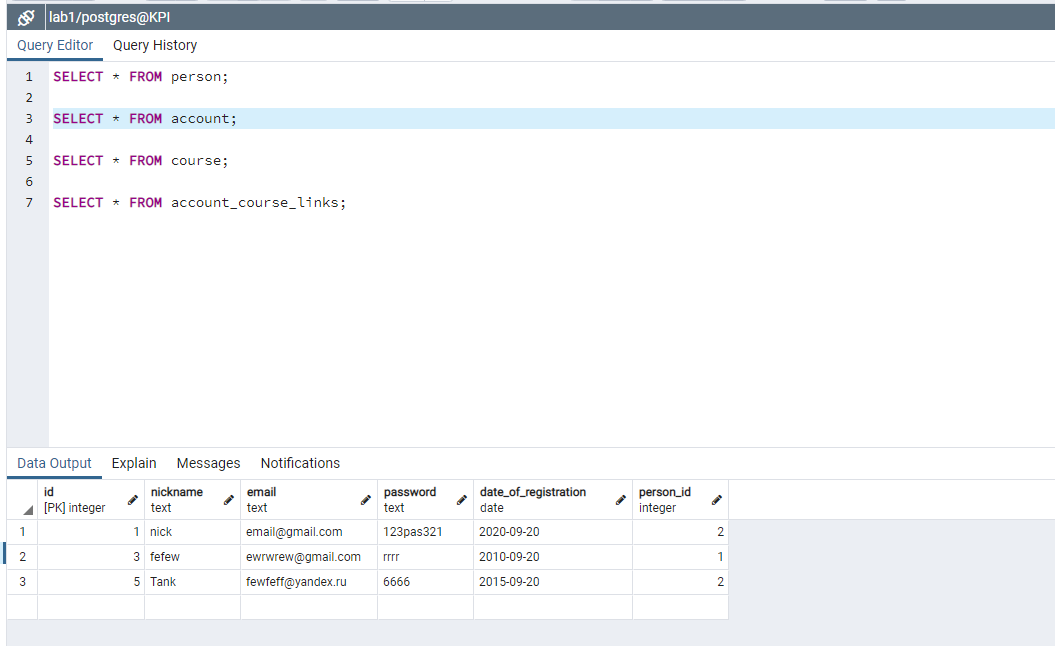


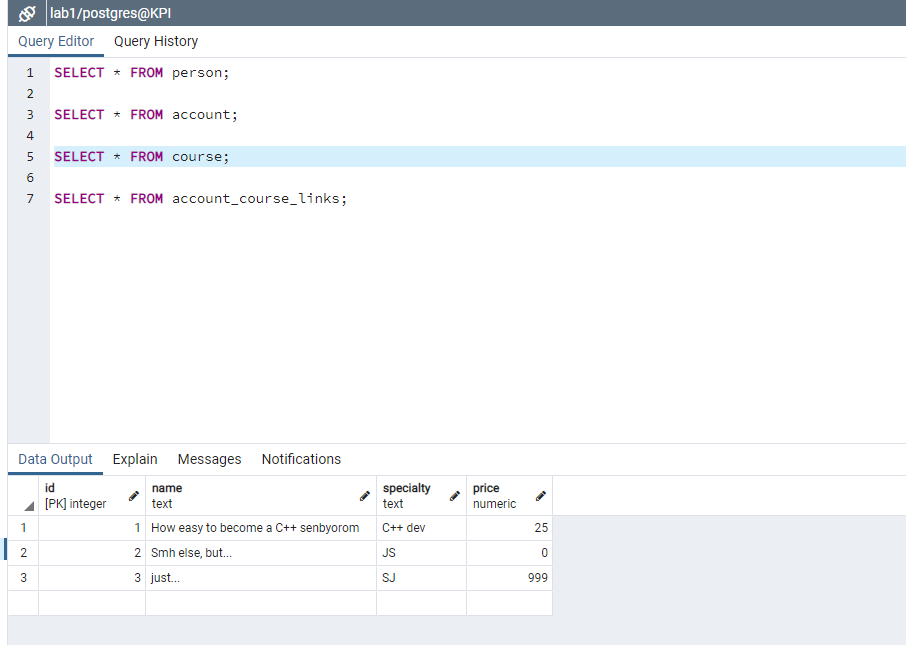


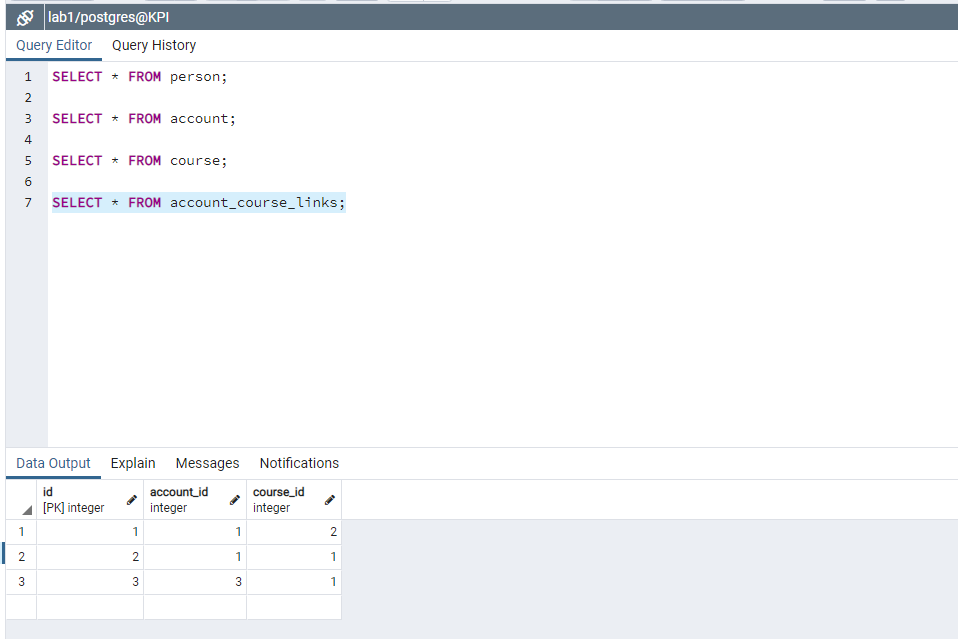


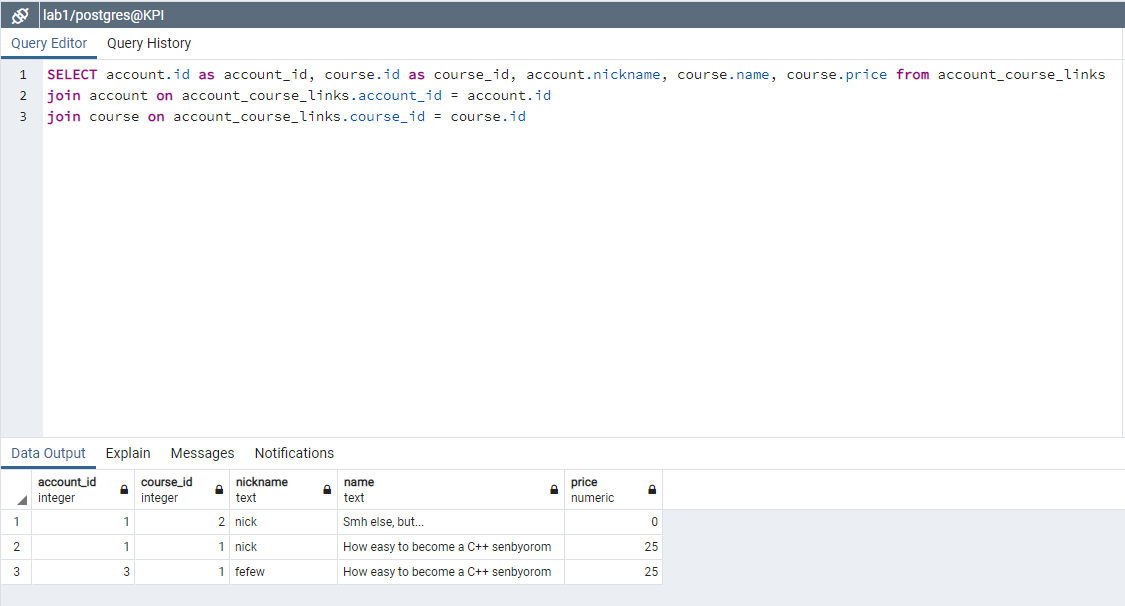
Скріншоти з даними в таблицях:











Відповіді на запитання:

1. **Сформулювати призначення діаграм типу «сутність-зв’язок».**

Схематично змоделювати сутності бази даних та зв’язки між ними.

1. **Назвати основні об’єкти схеми PostgreSQL.**

Таблиці, функції, тригери, процедури, представлення, типи.

1. **Навести приклади різних типів зв’язків у базах даних (1:1, 1:N, N:M).**

1:1 - наприклад у нас є користувач, у нього є багато полів, ми можемо розбити інформацію про нього у декілька окремих таблиць і між цими таблицями буде зв’язок 1:1.

1:N - наприклад у соціальній мережі користувач може ставити багато лайків.

N:M - наприклад книга і автор (у одного автора може бути багато книг, а у книги - багато авторів).