НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"

Кафедра програмного забезпечення комп'ютерних систем

Лабораторна робота №1

із дисципліни «Бази даних»

на тему **«Проектування бази даних та ознайомлення з базовими операціями. СУБД PostgreSQL»**

|  |  |  |
| --- | --- | --- |
| Виконав  студент 2 курсу  групи КП-92  Фенченко Ігор Юрійович |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  к. т. н. доцент  Петрашенко Андрій Васильович |

Київ 2020

**Мета роботи:** здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

**Завдання:**

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.

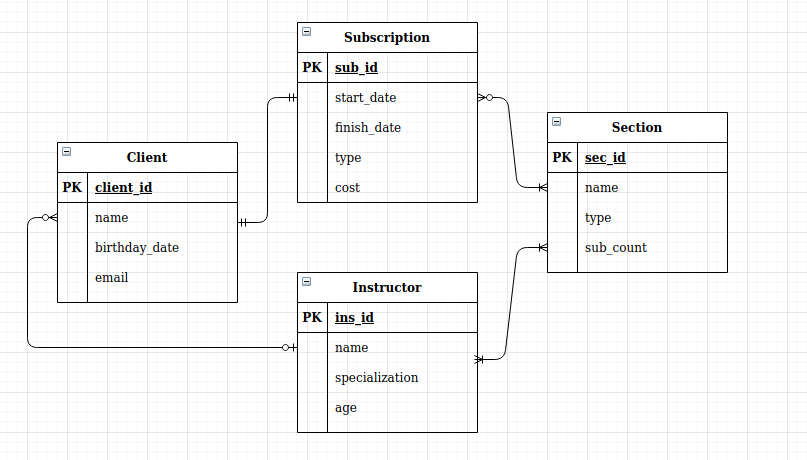
*Вимоги до ER-моделі*

1. Сутності моделі предметної галузі мають містити зв’язки типу 1:N або N:M.
2. Кількість сутностей у моделі – 3-4. Кількість атрибутів у кожній сутності: від двох до п’яти.
3. Передбачити наявність зв’язку з атрибутом.
4. Для побудови ER-діаграм використовувати одну із нотацій: Чена, “Пташиної лапки (Crow’s foot)”, UML.

**Завдання 1**

**Назва нотації:** “Пташиної лапки”.

**Графічний файл розробленої моделі:**

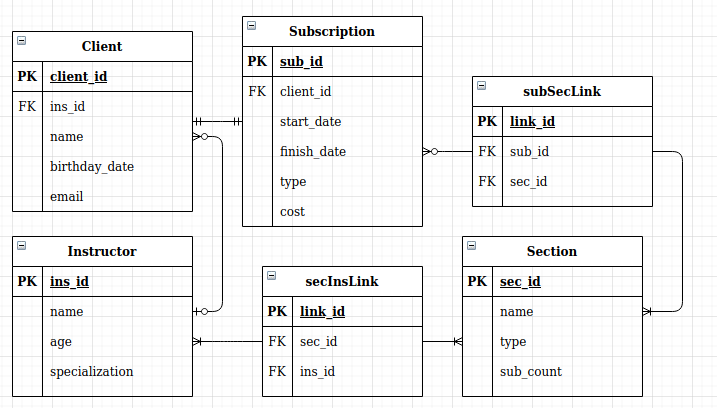


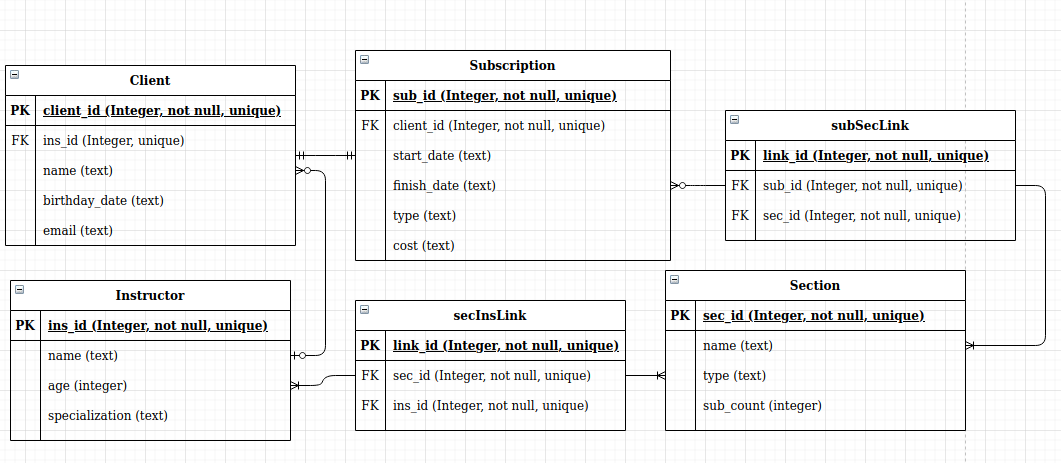
**Предметна галузь (Варіант):** Спортивний зал.

**Сутності:**

* Клієнт (Client)- відвідувач спортивного залу.
  + client\_id(PK) - унікальний ключ користувача
  + name - ФІО користувача
  + birthday\_date - дата народження користувача
  + email - електронна адреса користувача
* Абонемент (Subscription) - картка користувача
  + sub\_id(PK) - унікальний ключ абонементу
  + client(FK) - ключ власника абонементу
  + start\_date - початок дії абонементу
  + finish\_date - кінець дії абонементу
  + type - вид абонементу, що визначає можливості його власника
  + cost - вартість абонементу
* Секція спортивного залу (Section) - складова будівлі залу(поверх/приміщення/басейн)
  + sec\_id(PK) - унікальний ключ секції
  + name - назва секції
  + type - специфікація секції спортивного залу
  + sub\_count - максимальна кількість активних абонементів, що мають доступ до даної секції
* Інструктори (Instructor) - тренери спортивного залу
  + ins\_id(PK) - унікальний ключ тренера
  + name - ім’я тренера
  + specialization - область роботи тренера
  + age - вік тренера

**Завдання 2**

таблиця без типів



1. Сутність Client було перетворено у таблицю Client, а зв’язок Client➝Subscription (1:1) зумовив появу вторинного ключа (client\_id) в таблиці Subscription.
2. Сутність Subscription було перетворено у таблицю Subscription, а зв’язок Subscription➝Section (M:N) зумовив появу додаткової таблиці subSecLink, яка містить вторинні ключі sub\_id та sec\_id.
3. Сутність Section було перетворено у таблицю Section, а зв’язок Section➝Instructor (M:N) зумовив появу додаткової таблиці secInsLink, яка містить вторинні ключі ins\_id та sec\_id.
4. Сутність Instructor було перетворено у таблицю Instructor, а зв’язок Instructor➝Client (1:M) зумовив появу вторинного ключа (ins\_id) в таблиці Client.

**Завдання 3**

**Відповідність до НФ1:** Кожна таблиця повинна мати основний ключ (PK): мінімальний набір атрибутів, які ідентифікують запис, а кожен атрибут повинен мати лише одне значення, а не множину значень.

Розглянемо таблиці Client та Instructor в пункті 2. Опишемо функціональні залежності:

* ins\_id ➝ name, age, specialization (первинний ключ)
* name, age, specialization ➝ ins\_id (порушень НФ1 немає)
* client\_id ➝ ins\_id **замість** масиву client\_id в моделі Instructor (ins\_id ➝ client\_ids)

Аналогічно, всі інші таблиці в БД не мають множини значень як певний атрибут та мають первинний ключ, тобто НФ1 виконується.

**Відповідність до НФ2:** Кожна таблиця повинна відповідати вимогам першої нормальної форми та кожен не-первинний атрибут повинен повністю функціонально залежати від первинного ключа.

Розглянемо таблиці Subscription та Client в пункті 2. Опишемо функціональні залежності:

* client\_id ➝ name, birthday, email
* sub\_id ➝ start\_date, finish\_date, type, cost **замість** sub\_id ➝ name, birthday, email, finish\_date, cost… (Щоб уникнути складеного ключа, ми розбиваємо сутності клієнта та абонемента на 2 таблиці та робимо зв’язок через FK, ти самим задовольняючи НФ2)

**Відповідність до НФ3:** Таблиця повинна бути у другій нормальній формі, а дані в таблиці мають залежати винятково від первинного ключа.

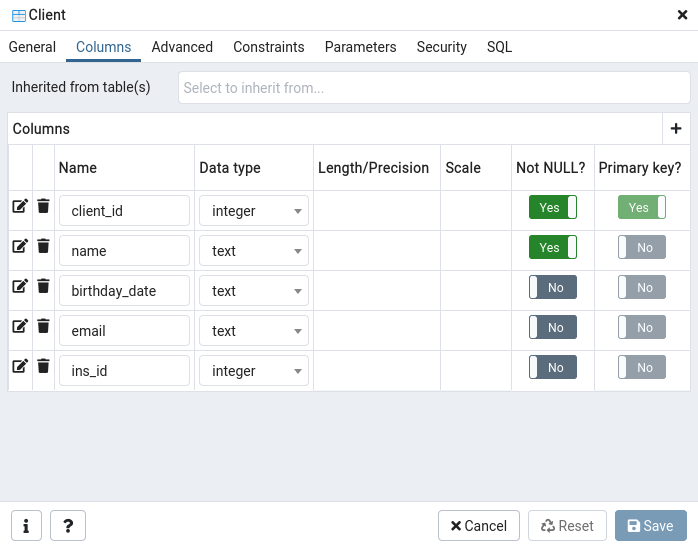
Розглянемо таблиці Section, Instructor та secIncLink в пункті 2. Опишемо функціональні залежності:

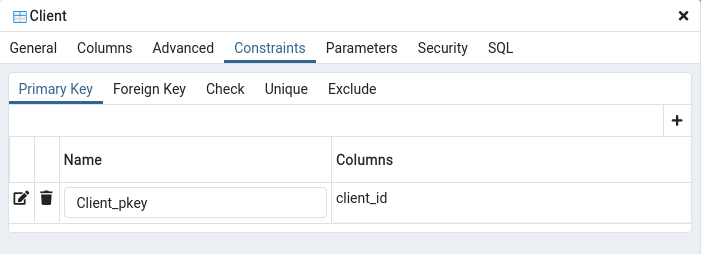
* ins\_id ➝ name, age, specialization
* sec\_id ➝ name, type, sub\_count
* link\_id ➝ sec\_id, ins\_id
* sec\_id, ins\_id ➝ link\_id **замість** ins\_id ➝ age, specialization, name, type, sub\_count (розбиваємо одну таблицю на 2 та таблицю зв’язків secIncLink, щоб первинний ключ не був складеним)

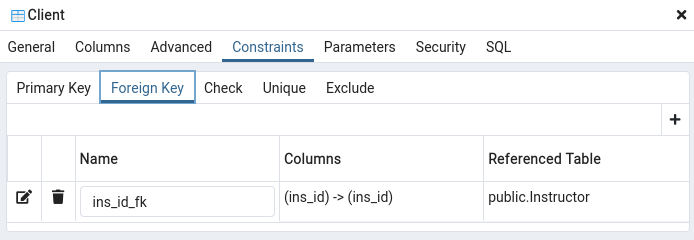
Отже, для всіх зв’язків M:N ми розробили таблиці зв’язків, щоб виконати умову НФ3.

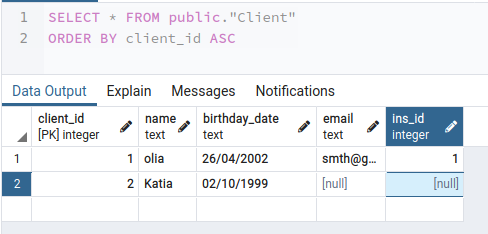
**Завдання 4**

* ***Таблиця Client***

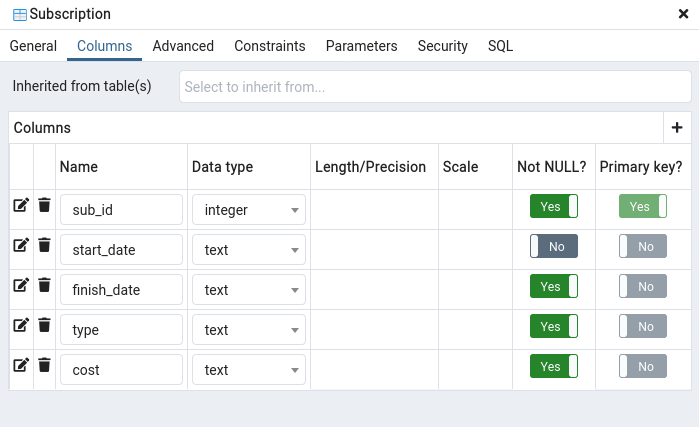


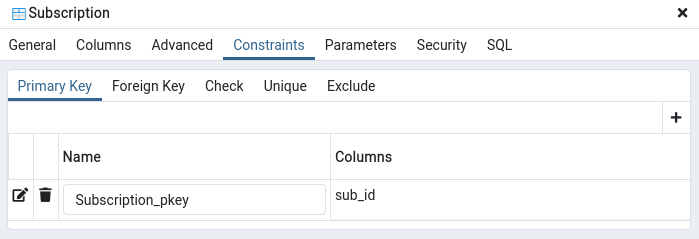


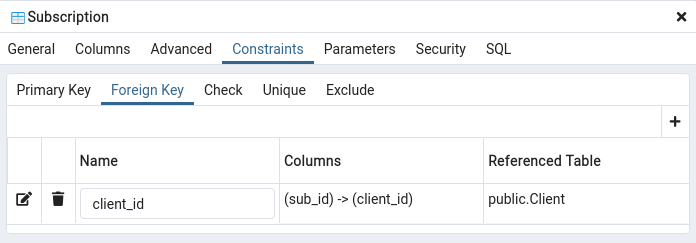


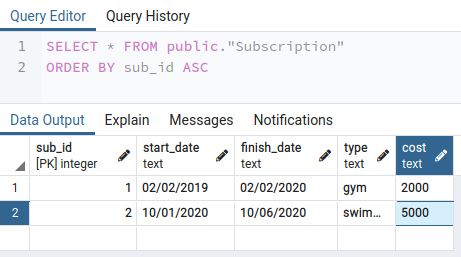


* ***Таблиця Subscriptions***

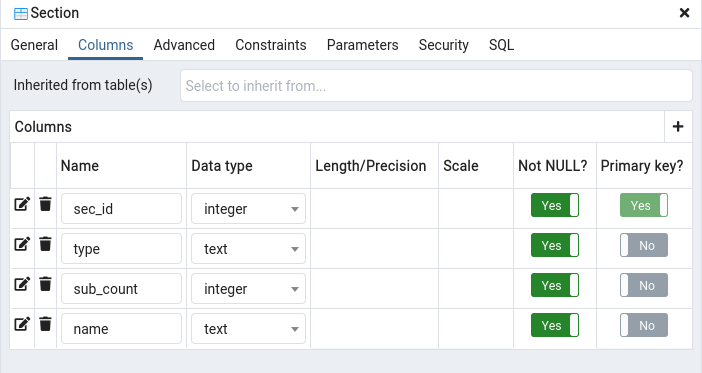


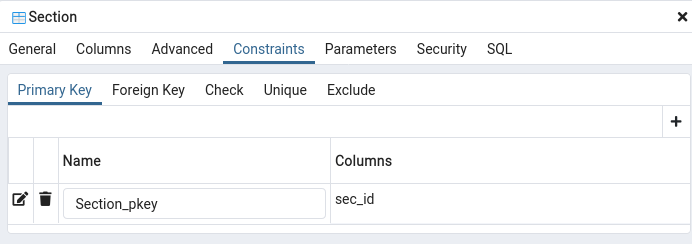


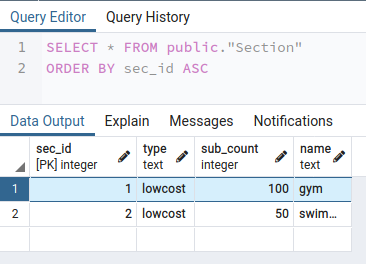




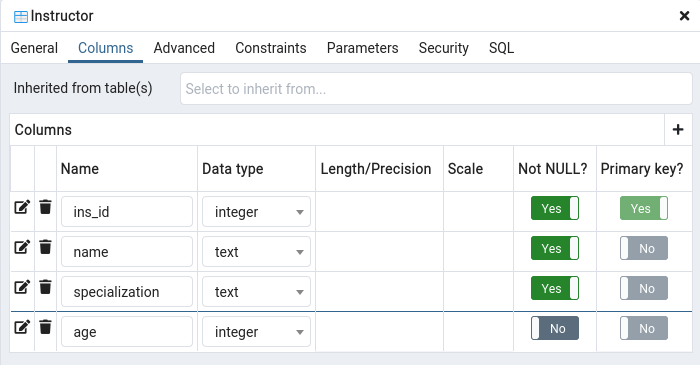
* ***Таблиця Section***

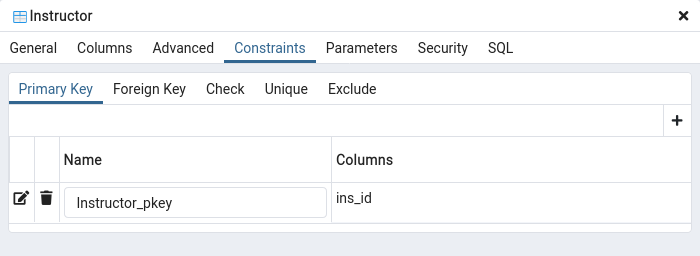


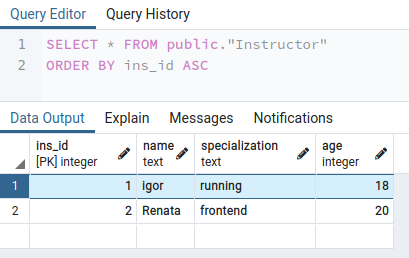




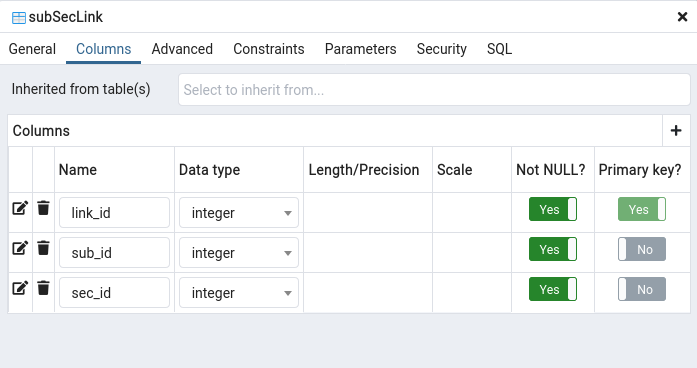
* ***Таблиця Instructor***

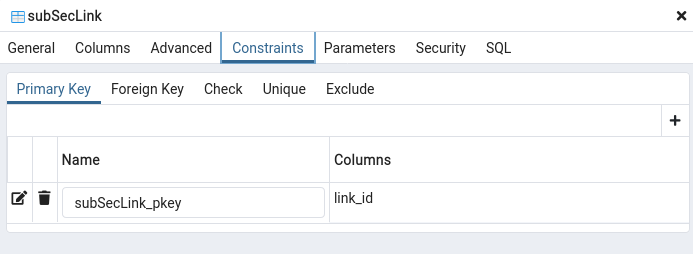


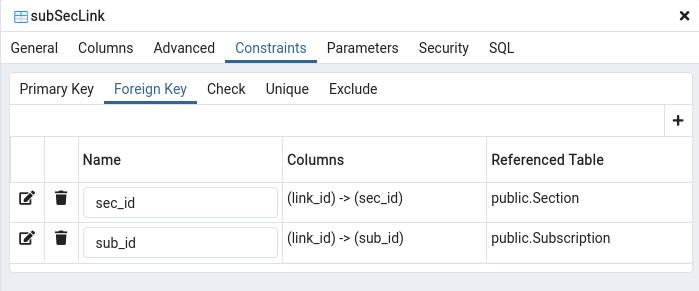


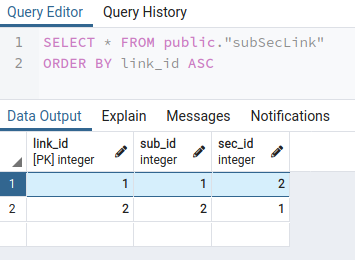


* ***Таблиця subSecLink***









* ***Таблиця secInsLink***

