НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

**Звіт**

**лабораторної роботи №1**

із дисципліни «Бази даних» на тему

«Практика роботи з запитами SQL»

*Варіант 2*

|  |  |  |
| --- | --- | --- |
| Виконала: | Керівник: |  |
| студентка групи КМ-21 | асистент Дмитренко О. А. |  |
| Челюскіна Ю.Ф. |  |  |

Київ — 2025

**Зміст**

[1 Мета 2](#_Toc191847565)

[2 Постановка задачі 2](#_Toc191847566)

[3 Хід роботи 2](#_Toc191847567)

[3.1 Запит 1: Агрегація з додатковими функціями 2](#_Toc191847568)

[3.2 Запит 2: Знайти студентів з однаковими оцінками в різних курсах 3](#_Toc191847569)

[3.3 Запит 3: Повний аналіз курсів, викладачів, студентів 4](#_Toc191847570)

[3.4 Запит 4: UNION запит з додатковими умовами 5](#_Toc191847571)

[4 Висновки 2](#_Toc191847572)

[5 Використані ресурси 3](#_Toc191847573)

# Мета

Закріпити навички роботи з SQL таблицями на прикладі бази даних з тематики освітніх курсів університету. Набути досвіду створення таблиць для конкретних запитів та очікуваних результатів.

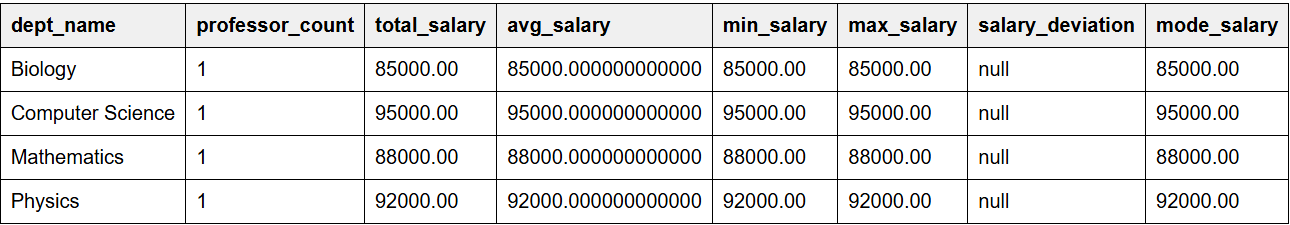
# Постановка задачі

Створити 3 таблиці(про University Courses, Student, Professor, та Department) та наповнити їх даними, щоб відповіді при виконанні наступних запитів були як очікується.

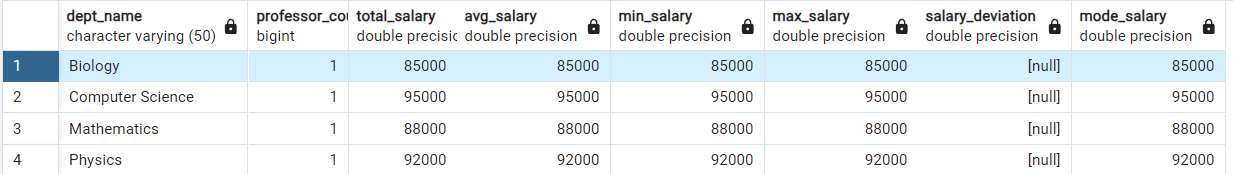
# Хід роботи

## **Запит 1: Агрегація з додатковими функціями**

**Очікуваний результат:**

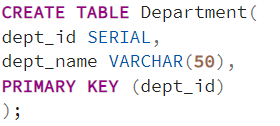
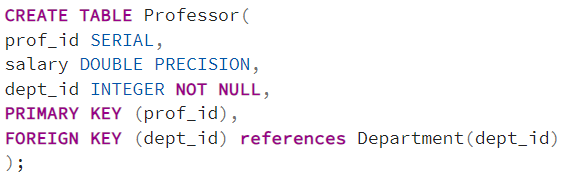


**Отриманий результат:**



По-перше було необхідно створити дві таблиці. Причому таблиця Professor залежить від Departments, адже викладачі відносяться до своїх відділів.

На цьому етапі код створення таблиць виглядає наступним чином:

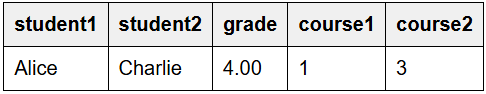
 

Це достатня база для подальшого удосконалення бази даних.

* Робота здійснюється таким чином, що в коді створюються таблиці, наповнюються необхідними даними за допомогою INSERT INTO, а після основного коду є рядки, що видаляють таблицю (DROP TABLE). Це дозволяє зручно вдосконалювати код.

## Запит 2: Знайти студентів з однаковими оцінками в різних курсах

**Очікуваний результат:**

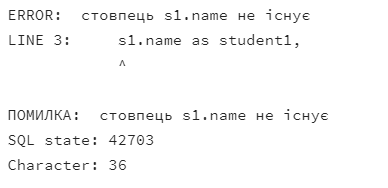


**Отриманий результат:**

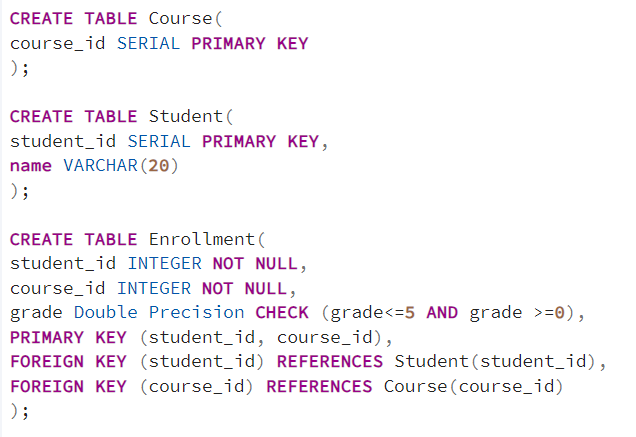


Після аналізу цього запиту було створено три нових таблиці. Це Course, Student та Enrollement, як пов’язує дві перші.

В процесі через неуважність було пропущено атрибут name для студента, що призвело до помилки:

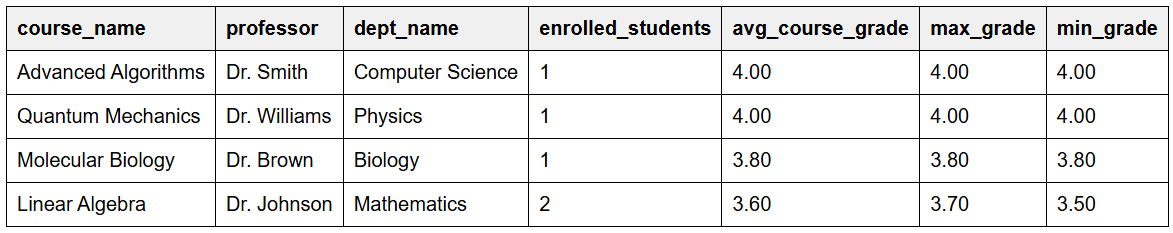


Як результат код таблиць виглядає так:

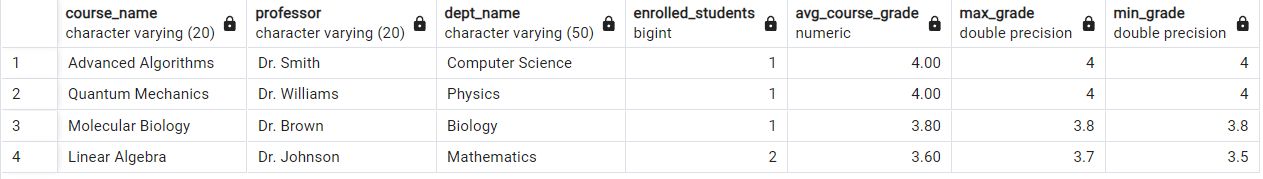


## **Запит 3: Повний аналіз курсів, викладачів, студентів**

**Очікуваний результат:**

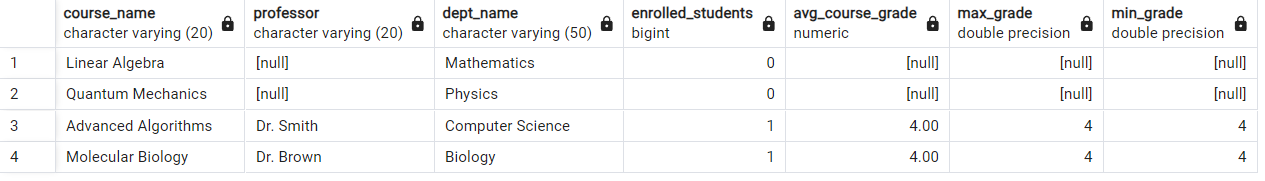


**Отриманий результат:**



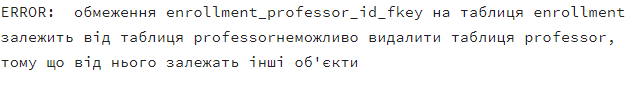
Спершу було додано атрибути назви курса, ім’я викладача, FK викладача в Enrollment та FK ід відділу в курс.

Після цих дій, а також зміни даних в таблицях отримали такий результат:



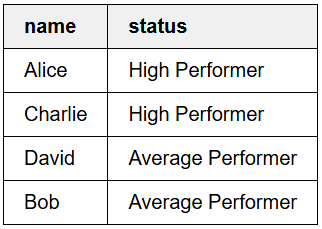
Очевидно, що в нас наразі недостатня кількість студентів, що впливає на результат.

Отже спочатку додаємо нових студентів до таблиці Students (всього має бути 5), а потім робимо відношення між ними та курсами таким чином, щоб досягти бажаного результату.

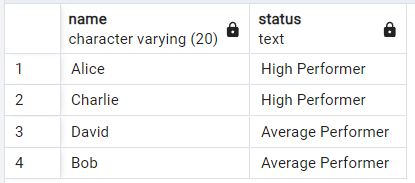
* **В процесі зміни коду та додавання нових залежностей варто звертати увагу на порядок видалення таблиць, щоб не отримати подібні помилки:**

## Запит 4: UNION запит з додатковими умовами

**Очікуваний результат:**

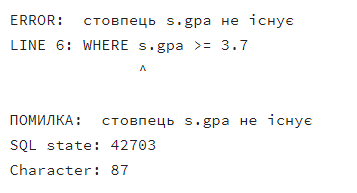


**Отриманий результат:**



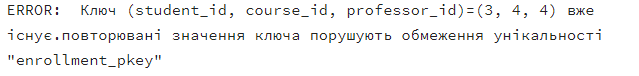
Так як в нас вже є чимало атрибутів та таблиць, простіше було перевірити що саме не працює, ніж аналізувати кожен рядок.

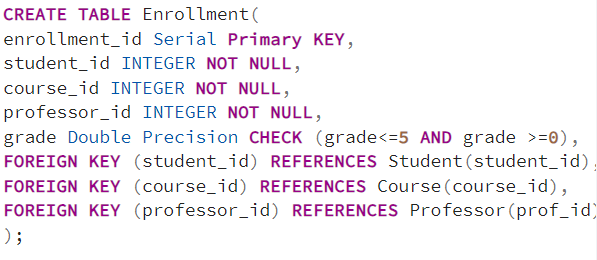
Після запуску коду отримали очікувану помилку:



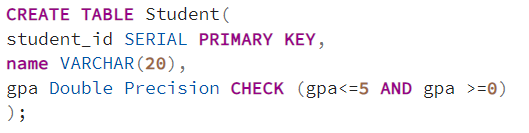
На цьому етапі стало зрозуміло, що в нас всього 4 студенти, а не 5, як здавалось, тобто один студент проходить два курси, що є логічно.

Хоча таблиця Enrollment не передбачала такого використання, тому було додано окремий первинний ключ, який не вимагає від студента, викладача чи відділу бути унікальним значенням в цій таблиці:





Також, власне повертаючись до першої помилки, було додано атрибут gpa до таблиці студентів:



# Висновки

В ході цієї лабораторної роботи було повторено та закріплено знання про створення БД в PostgreSQL.

Виконуючи завдання, вдалося набути досвіду аналізу вже створених запитів до БД. Всі запити були добре зрозумілі та логічні, тому особливих проблем не викликали. Найкраще запам'ятались команди Join та Group By, адже саме вони найбільше впливали на очікуваний результат.

Фінальний вміст БД виглядає так:

INSERT INTO Department (dept\_id, dept\_name)

VALUES

(1, 'Biology'),

(2, 'Computer Science'),

(3, 'Mathematics'),

(4, 'Physics');

INSERT INTO Professor(prof\_id, name, salary, dept\_id) VALUES

(1,'Dr. Smith', '85000.00', 1),

(2,'Dr. Williams', '95000.00', 2),

(3,'Dr. Brown', '88000.00', 3),

(4,'Dr. Johnson', '92000.00', 4);

INSERT INTO Course(course\_id, course\_name, dept\_id) VALUES

(1, 'Advanced Algorithms', 2),

(3, 'Quantum Mechanics', 4),

(2, 'Molecular Biology', 1),

(4, 'Linear Algebra', 3);

INSERT INTO Student(student\_id, name, gpa) VALUES

(1, 'Alice', 4),

(2, 'David', 3.6),

(3, 'Bob', 3.5),

(4, 'Charlie',3.8)

;

INSERT INTO Enrollment (student\_id, course\_id, professor\_id, grade) VALUES

(1, 1, 1, 4),

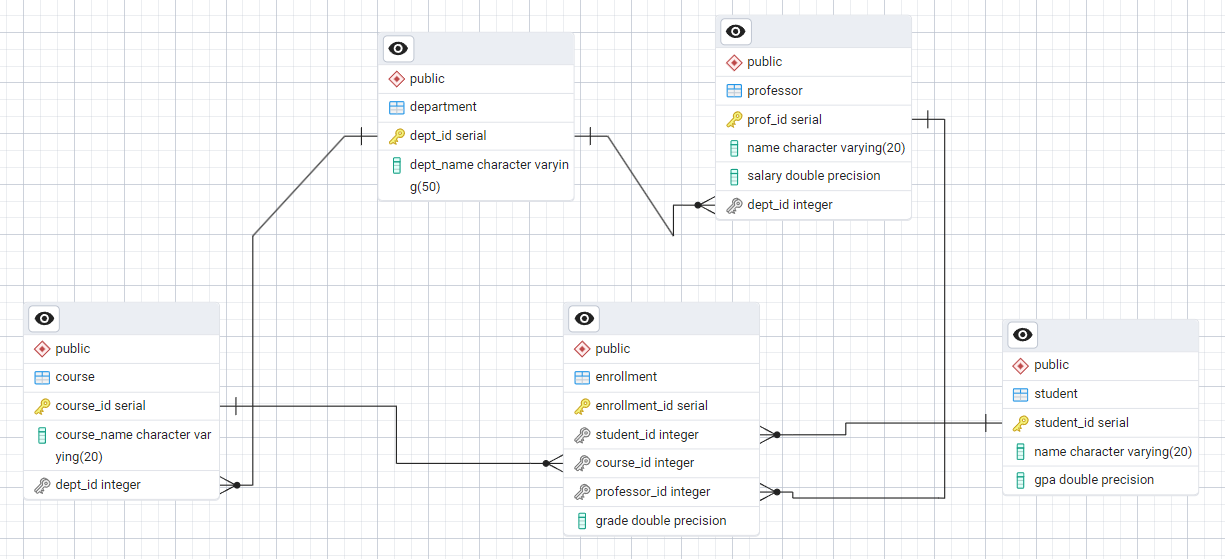
(3, 4, 4, 3.5),

(2, 4, 4, 3.7),

(4, 3, 2, 4),

(2, 2, 3, 3.8);

ERD виглядає так:



# Використані ресурси

1. Інформація по SQL запитам –

<https://www.geeksforgeeks.org/sql-tutorial/?ref=lbp>

1. Синтаксис PostgreSQL та приклади використання - <https://www.w3schools.com/postgresql/index.php>