

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»**

**на тему** «Ознайомлення з базовими операціями СУБД PostgreSQL»

Виконав: студент ІII курсу

ФПМ групи КВ-11

Лабазов Володимир Володимирович

Перевірив: Петрашенко А.В.

Київ – 2023

**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.

*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.

*Вимоги до інструментарію*

1. Створення ER-діаграм: Google Docs (Drawing) або <https://www.draw.io/> або [https://www.lucidchart.com](https://www.lucidchart.com/)
2. Середовище для створення таблиць відлагодження SQL-запитів до бази даних – pgAdmin 4.
3. СУБД - PostgreSQL 10, 11 або 12.

*Вимоги до оформлення лабораторної роботи у електронному вигляді*

Опис лабораторної роботи у  **репозиторії** **GitHub** включає: назву лабораторної роботи, варіант студента (опис обраної предметної галузі) та вимоги до звітування щодо пунктів 1-4 завдання, які наведено нижче:

*У звіті щодо пункту №1 завдання має бути:*

* перелік сутностей з описом їх призначення;
* графічний файл розробленої моделі «сутність-зв’язок»;
* назва нотації.

*У звіті щодо пункту №2 завдання має бути:*

* опис процесу перетворення (наприклад, “сутність А було перетворено у таблицю А, а зв’язок R (M:N) зумовив появу додаткової таблиці R1 тощо);
* схему бази даних у графічному вигляді **з назвами таблиць (!) та зв’язками між ними, а також  необхідно намалювати перетворену ER-діаграму у ТАБЛИЦІ БД! Це означає, що тут не може бути зв'язку N:M, мають бути позначені первинні та зовнішні ключі, обмеження NOT NULL та UNIQUE і бажано внести типи даних атрибутів.**

*У звіті щодо пункту №3 завдання має бути:*

* пояснення (**обґрунтування!**) щодо відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3. Пояснення ***полягає у наведенні функціональних залежностей***, що демонструють висновки. У випадку невідповідності надати опис необхідних змін у схемі;
* У випадку проведення змін у схемі бази даних надати оновлену версію схеми, інакше - не наводити схему.

*У звіті щодо пункту №4 завдання має бути:*

* навести копії екрану з pgAdmin4, що відображають назви, типи та обмеження на стовпці (доступне у закладці “Columns” та “Constraints” властивостей “Properties” таблиць дерева об’єктів у pgAdmin4);
* навести копії екрану з pgAdmin4, що відображають вміст таблиць бази даних у PostgreSQL. Таблиці на зображенні обов'язково **повинні мати назву**!

# Опис предметної галузі

При проектуванні бази даних «Система обліку екзаменаційних балів студентів» я виділив наступні сутності: Студент (Student), Група (Group), Предмет (Discipline), Оцінка (Mark).

Група може містити багато студентів , але студент може знаходитись в одній групі (зв'язок 1:N).

Кожен студент має багато оцінок (зв'язок 1:N).

Кожен студент ходить на не одну дисципліну , і на не одну дисципліну ходять не один студент (зв'язок N:M).

Кожна оцінка має один предмет з якого вона була отримана , і кожен предмет має одну оцінку за екзамен (зв'язок 1:1).

# Таблиця сутностей з описом їх призначення

|  |  |  |
| --- | --- | --- |
| **Сутність** | **Атрибут** | **Тип (розмір)** |
| Сутність «Student» містить інформацію про студента | id (PK) – унікальний id студента  name – ім’я студента  group\_id(FK) – унікальний id власника | Числовий  Текстовий (50)  Числовий |
| Сутність «Group» містить інформацію про групу | id (PK) – унікальний id групи  name – назва групи | Числовий  Текстовий (5) |
| Сутність «Discipline» містить інформацію про предмет | id (PK) – унікальний id дисципліни  name – назва дисципліни  teacher\_name – ім’я викладача дисципліни | Числовий  Текстовий (50)  Текстовий (50) |
| Сутність «Mark» містить інформацію про екзаменаційну оцінку | value – оцінка в балах  dicipline\_id(FK) – id дисципліни з якої була отримана оцінка  student\_id(FK) – id студента , що отримав оцінку  when\_received – унікальний id головного тренера | Числовий  Числовий  Числовий  Дата |

# Концептуальна модель предметної області “Система обліку екзаменаційних балів студентів”

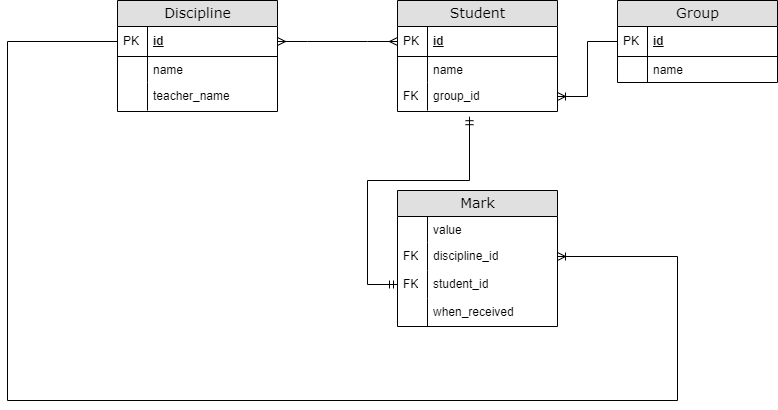


Рисунок 1 – Концептуальна модель предметної області «Система обліку екзаменаційних балів студентів».

**Нотація:** «UML». Модель побудована засобами програми draw.io

# Опис процесу перетворення

Сутності «Student», «Group», «Discipline», «Mark» було перетворено у таблиці. Зв'язок між студентом та предметом (зв'язок багато до багатьох) зумовив появі додаткової таблиці «student\_disciplines», яка містить унікальні id студента та дисципліни.

# 

# Відповідність схеми БД нормальним формам

Схема БД відповідає першій нормальній формі тому, що схема передбачає один елемент в кожній комірці. Кожна таблиця передбачує рівно одне значення в одну комірку (у студента лише одне ім’я, у групи лише одна назва, у предмета лише одна назва і предмет може читати лише один викладач, у оцінки є одна дата її отримання).

Схема БД відповідає другій нормальній формі тому, що вона, по-перше, відповідає першій нормальній формі та, по-друге, не включає в собі залежності від декількох потенційних ключів, тобто залежить лише від одного ключа(відношення між всіма таблицями відбуваються через поля id, що є унікальними)

Схема БД відповідає третій нормальній формі тому, що вона, по-перше, відповідає другій нормальній формі та, по-друге, не містить транзитивних функціональних залежностей не ключових атрибутів від ключових(всі поля логічно розділені і таблиці з’єднані через поля id, за якими далі вже можна отримати більшу інформацію).

# Скріншоти з DBeaver

# students

# 

# groups

# 

# disciplines

# 

# marks

# 

# student\_disciplines

