A black and white drawing of a building

Description automatically generated

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

Виконала:

студентка ІІI курсу

групи 34

Попов Єгор

Перевірив:

Павловский В. І.

Київ – 2025

**Мета:** 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.

**Опис предметної області**

Дана предметна область — система управління процесом навчання на курсах. Вона включає інструменти для організації, моніторингу та збереження інформації про курси, студентів і викладачів, забезпечує реєстрацію слухачів, контроль статусу проходження навчання, призначення викладачів до курсів, а також формування звітності. Система дозволяє ефективно керувати навчальним процесом, відстежувати результати студентів та оптимізувати взаємодію між усіма учасниками освітнього процесу.

**Опис сутностей**

Для побудови концептуальної моделі обраної предметної галузі, були виділені такі сутності:

1. Курс(Course)

Атрибути: індефікатор проекту; назва проекту; опис проекту.

Призначення: збереження інформації про різні курси.

1. Студент(Student)

Атрибути: індефікатор студента; ПІБ студента; група студента.

Призначення: збереження інформації про різних студентів на курсах.

1. Професор(Professor)

Атрибути: індефікатор професора; ПІБ професора.

Призначення: збереження інформації про різних професорів на курсах.

**Опис зв’язків між сутностями**

Зв’язок-сутність “Студент” – “Курс” є зв'язок N:M. Один студент може бути зареєстрований на кілька курсів, і один курс має багато студентів.

Зв’язок-сутність “Курс” – “Викладач” зв'язок 1:N. Один викладач може викладати багато курсів, але курс має лише одного викладача.

Графічне подання концептуальної моделі «Сутність-зв’язок» зображено на рисунку 1.

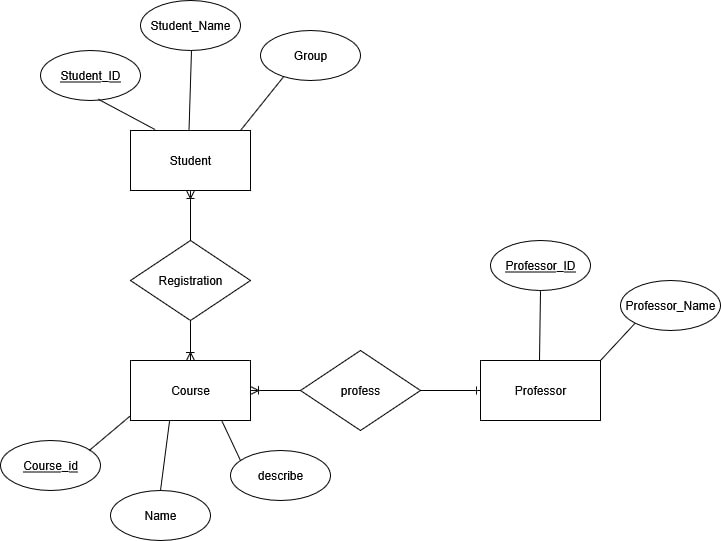


Рисунок 1 – ER-діаграма, побудована за нотацією Чена

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

Сутність “Professor” перетворено в таблицю “Professor” з первиним ключем Professor\_ID та атрибутом Professor\_Name.

Сутність “Course” перетворено в таблицю “Course” з первиним ключем Course\_ID, атрибутами Name і Describe, та зовнышнім ключем Professor\_ID.

Сутність “Student” перетворено в таблицю “Student” з первиним ключем Student\_ID, атрибутами Student\_Name та Group.

Оскільки в логічній моделі безпосередній зв’язок N:M є неможливим а в концептуальній моделі він існує між сутностями Course і Student, то для його реалізації було створено таблицю Registration( регістрація на проект), з первинним ключем Registration\_ID, атрибутами Status і Date, та зовнішніми ключами Student\_ID та Course\_ID

Графічне подання логічної моделі «Сутність-зв’язок» зображено на рисунку 2.

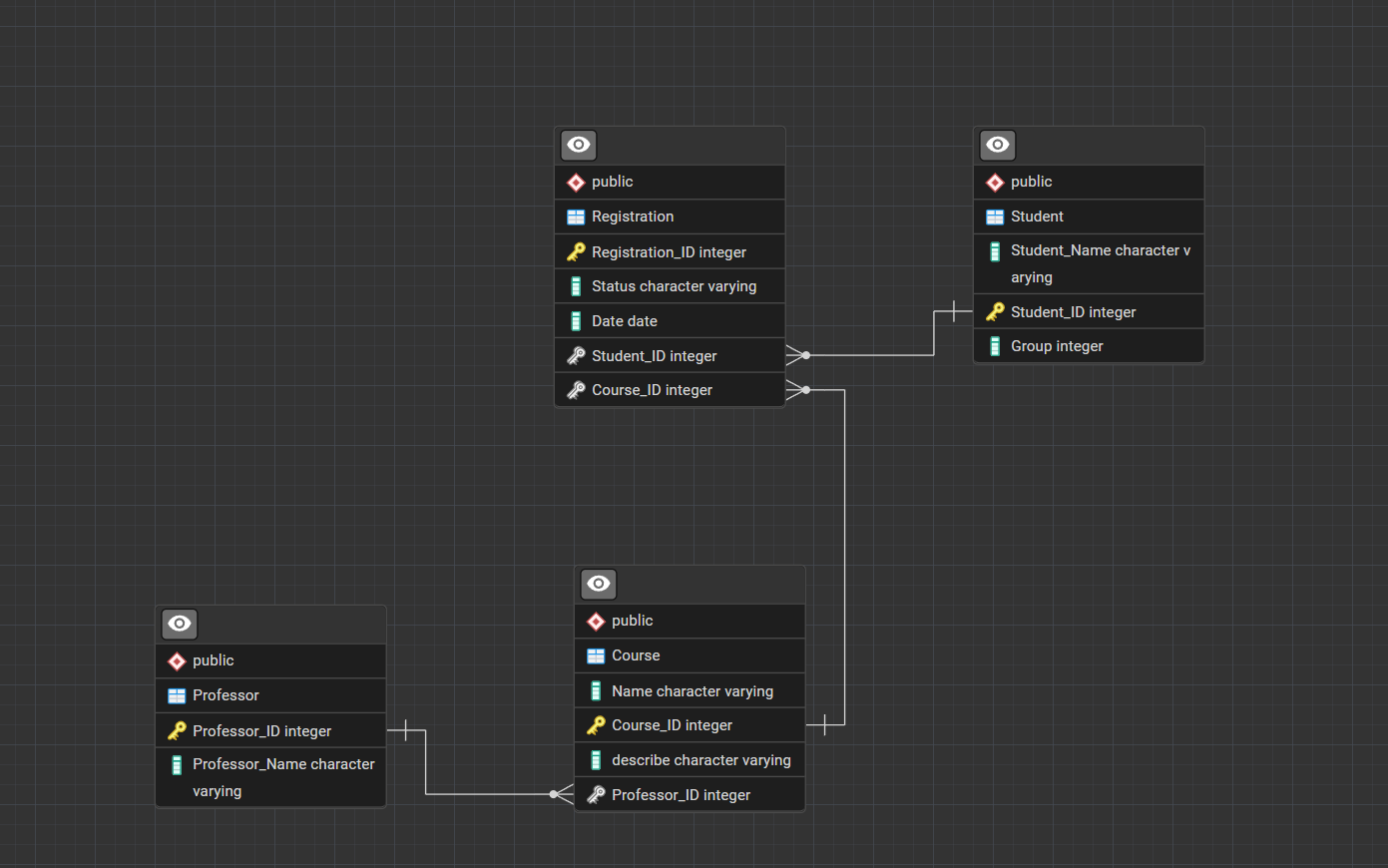


Рисунок 2 – Схема бази даних

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої.

Таблиця 1 – Опис об’єктів бази даних

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| Professor – містить дані про професора | Professor\_ID - унікальний індефікатор професора | Integer(число) |
| Professor\_Name – ПІБ професора | Character varying(рядок) |
| Course – містить дані про курс | Course\_ID - унікальний індефікатор курсу | Integer(число) |
| Name – назва курсу | Character varying(рядок) |
| Describe – опис курсу | Character varying(рядок) |
| Professor\_Id - унікальний індефікатор професора | Integer(число) |
| Student – містить дані про студента | Student\_ID - унікальний індефікатор студента | Integer(число) |
| Student\_Name – ПІБ студента | Character varying(рядок) |
| Group – номер групи | Integer(число) |
| Registration – містить дані про регістрацію | Registration\_ID - унікальний індефікатор регістрації | Integer(число) |
| Status – статус проходження курсу | Character varying(рядок) |
| Date – початок курсу | Date(дата) |
| Student\_ID - унікальний індефікатор студента | Integer(число) |
| Course\_ID - унікальний індефікатор курсу | Integer(число) |

**Функціональні залежності для кожної таблиці**

1. Course:
2. Course\_ID – {Name, Describe}
3. Describe – {Name}
4. Student:
5. Student\_ID – {Student\_Name, Group}
6. Group – {Student\_Name}
7. Professor:
8. Professor\_ID – {Professor\_Name}
9. Registration:
10. Registration\_ID – { Student\_ID, Course\_ID, date, status}
11. Status – { Student\_ID }

**Відповідність схеми нормальним формам**

Вона відповідає ІНФ, оскільки всі атрибути є атомарними (недільними). та мають просту структуру, не містять складних об'єктів або масивів.

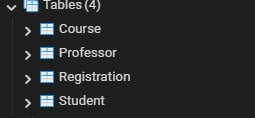
Вона відповідає 2НФ, оскільки всі неключові атрибути залежать від цілісних ключів цих таблиць. Наприклад, атрибут "Name" в таблиці "Course". залежить від ідентифікатора "Course\_ID".

Вона також відповідає ЗНФ, оскільки усі неключові атрибути мають бути функціонально залежними від первинного ключа, тобто не може бути взаємозалежностей між неключовими атрибутами. Кожен атрибут безпосередньо пов'язаний з ключами, від яких він залежить, що і виконується у нас.

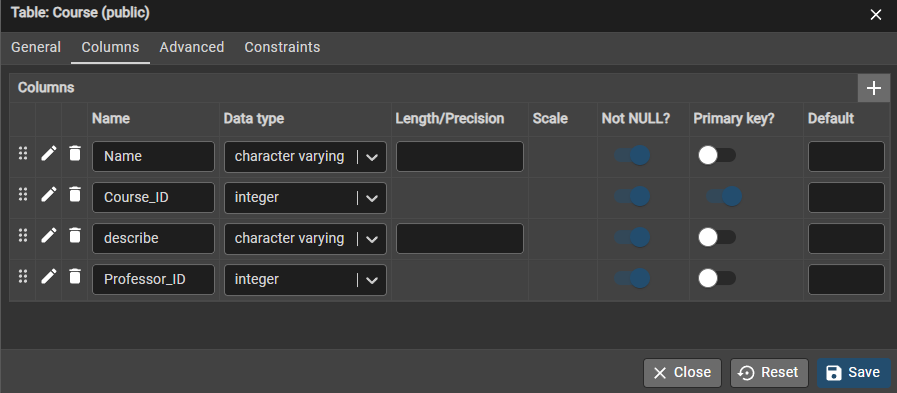
Загальний висновок полягає в тому, що ця база даних добре. структурована і нормалізована, що робить її ефективною та надійною для зберігання і обробки даних.

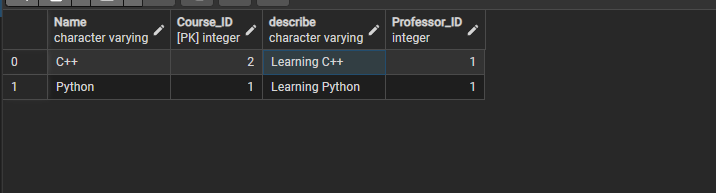
**Таблиці бази даних у pgAdmin4**

Таблиці у pgAdmin4

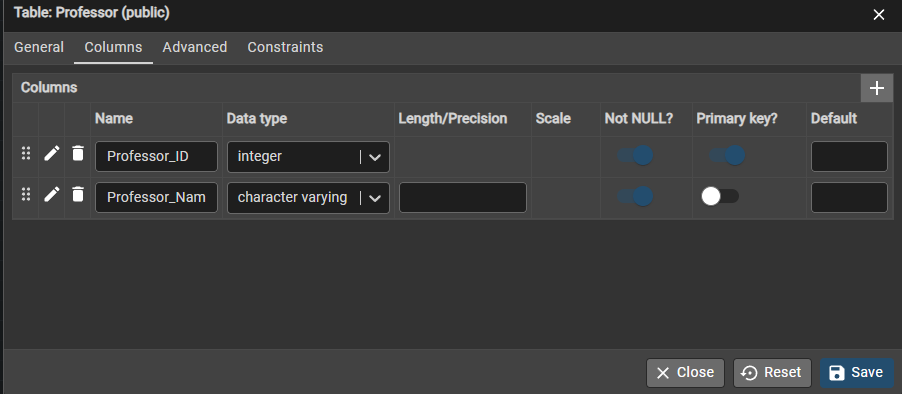


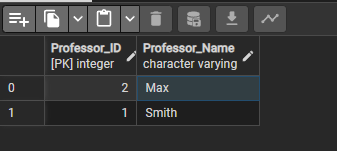
Таблиця Course



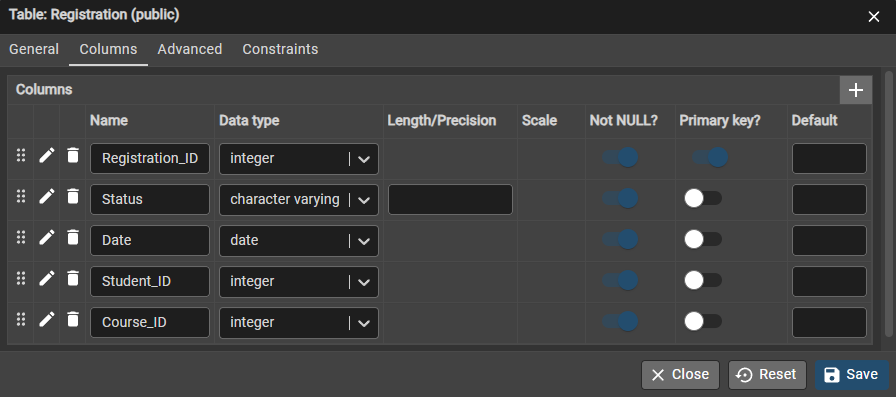


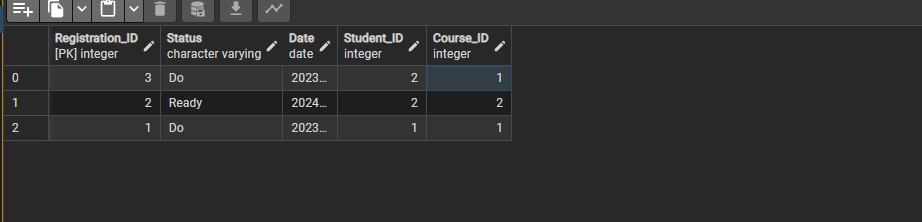
Таблиця Professor



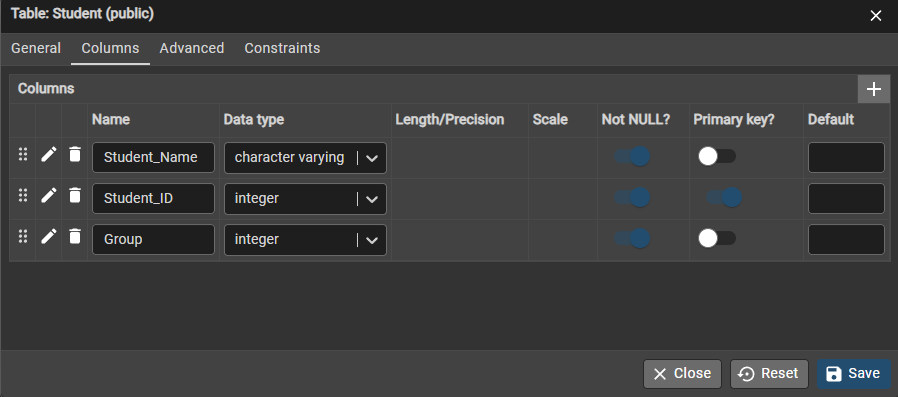


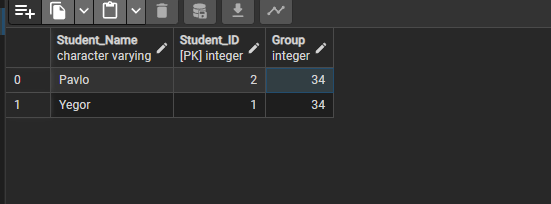
Таблиця Registrarion





Таблиця Student





**Таблиці в коді SQL**

-- This script was generated by the ERD tool in pgAdmin 4.

-- Please log an issue at https://github.com/pgadmin-org/pgadmin4/issues/new/choose if you find any bugs, including reproduction steps.

BEGIN;

CREATE TABLE IF NOT EXISTS public."Course"

(

    "Name" character varying,

    "Course\_ID" integer NOT NULL,

    describe character varying,

    "Professor\_ID" integer NOT NULL,

    PRIMARY KEY ("Course\_ID")

);

CREATE TABLE IF NOT EXISTS public."Student"

(

    "Student\_Name" character varying,

    "Student\_ID" integer,

    "Group" integer,

    PRIMARY KEY ("Student\_ID")

);

CREATE TABLE IF NOT EXISTS public."Professor"

(

    "Professor\_ID" integer,

    "Professor\_Name" character varying,

    PRIMARY KEY ("Professor\_ID")

);

CREATE TABLE IF NOT EXISTS public."Registration"

(

    "Registration\_ID" integer,

    "Status" character varying,

    "Date" date,

    "Student\_ID" integer NOT NULL,

    "Course\_ID" integer NOT NULL,

    PRIMARY KEY ("Registration\_ID")

);

ALTER TABLE IF EXISTS public."Course"

    ADD FOREIGN KEY ("Professor\_ID")

    REFERENCES public."Professor" ("Professor\_ID") MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public."Registration"

    ADD FOREIGN KEY ("Student\_ID")

    REFERENCES public."Student" ("Student\_ID") MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

ALTER TABLE IF EXISTS public."Registration"

    ADD FOREIGN KEY ("Course\_ID")

    REFERENCES public."Course" ("Course\_ID") MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID;

END;

**Висновок**

Під час виконання лабораторної роботи була створена база даних для системи управління процесом начання на курсах. Ця база даних містить інформацію про професорів, студентів, курсів та їх взаємозв'язки.

Діаграма сутностей і взаємозв'язків (ER-діаграма) була використана для графічного представлення структури бази даних. Ми використовували нотацію Чена для позначення зв'язків і атрибутів.

Схема бази даних була перевірена на відповідність нормальним формам ΗΦΙ, ΗΦ2-і- НФЗ. Усі таблиці відповідають цим нормальним формам, що гарантує надійне та ефективне зберігання даних.

Також була представлена оновлена версія схеми бази даних, де була включена таблиця "Registrarion" для відображення взаємозв'язків між курсами та студентами, які їх проходять .

У кінці були надані знімки екрана з pgAdmin4, що демонструють властивості стовпців та обмеження, а також вміст таблиць бази даних PostgreSQL.

Загалом, розроблена база даних відповідає всім вимогам та завданням і готова до подальшого використання.

**Контакти:**

Посилання на репозиторій GitHub - https://github.com/PacMaN312/Database\_Popov\_KV-34.git