

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”*

Виконав:

студент ІII курсу

групи КВ-02

Шпот І.Р.

Перевірив:

Київ – 2022

*Метою роботи є* здобуття 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.

*Зміст звіту*

1. Опис проблемного середовища;
2. Концептуальна модель предметної області;
3. Логічна модель (схема) даних БД;
4. Склад СУБД PostgreSQL;
5. Список обмежень цілісності в термінах СУБД PostgreSQL;
6. Фізична модель (схема) даних БД в pgAdmin ----;
7. Приклад вмісту БД.

**Опис предметної області «Школа»**

Обрана предметна область передбачає зберігання та обробку даних про викладачів, предмети, учнів та їх оцінки. Предмети можуть бути з різних дисциплін. Викладачі можуть ввести декілька предметів. Учні можуть вивчати декілька предметів. Розклад є індивідуальним для кожного викладача чи учня і може містити предмети відповідні викладачу, що його викладає, та учню, що його вивчає. Оцінки виставляються викладачем учню з якогось певного предмету.

**Опис сутностей предметної області**

* + - 1. Предмети (Subjects), з атрибутами: назва , id предмету;
      2. Викладачі (Teachers), з атрибутами: ім’я , id викладача, адреса, номер телефону, електронна пошта. Призначення: збереження інформації про викладача;
      3. Учні (Students), з атрибутами: ім’я, id учня, адреса, номер телефону, електронна пошта. Призначення: збереження інформації про учня;
      4. Розклад (Schedule), з атрибутами: id розкладу, id предмету, id викладача, id учня, день тижня. Призначення: утворення зв’язку між предметами, викладачами, що викладають ці предмети, та учнями, що вивчають ці предмети, також для збереження інформації про індивідуальний розклад та днем тижня цього розкладу

**Опис зв’язків між сутностями предметної області**

Сутності «Предмети», «Викладачі» та «Учні» мають зв’язок M:N між собою, оскільки у нас є багато предметів, які вивчають багато учнів, а у них є багато викладачів. Для реалізації цього зв’язку ми використовуємо сутність «Розклад», що дозволяє нам зберігати інформації про те, які та коли предмети викладаються викладачами учням.

Сутність «Предмети» має зв’язок 1:N по відношенню до сутності «Розклад», оскільки один предмет може бути в різних розкладах, але розклад може містити в певного часу лише один предмет для викладача, що його викладає, і для учня, що його вивчає.

Сутність «Викладачі» має зв’язок 1:N по відношенню до сутності «Розклад», оскільки один викладач може бути в різних розкладах, але розклад може мати в певний час лише одного викладача, що викладає певний предмет для певного учня.

Сутність «Учні» має зв’язок 1:N по відношенню до сутності «Розклад», оскільки один учень може мати декілька різних розкладів для різних предметів та викладачів, що введуть ці предмети, але ці розклади можуть мати лише одного учня до якого ці розклади відносяться.

**Концептуальна модель предметної області «»**

Концептуальна модель наведена на Рисунок 1.

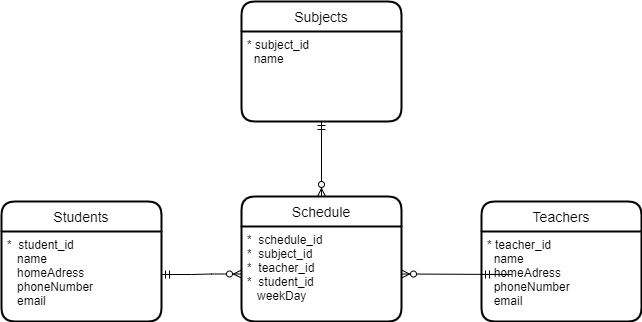


Рисунок 1 – ER-діаграма предметної галузі «Школа», побудована за нотацією Crow’s foot (інструмент draw.io)

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

Сутність «Предмети» перетворено у таблицю «students».

Сутність «Викладачі» перетворено у таблицю «teachers».

Сутність «Учні» перетворено у таблицю «students».

Зв’язки M:N між сутностями «Предмети» і «Викладачі», «Викладачі» і «Учні» та «Учні» і «Предмети» було реалізовано через сутність «Розклад», яку перетворено в таблицю «schedule» із зовнішніми ключами subject\_id, teacher\_id та student\_id.

**Логічна модель (схема) БД «Школа»**

Логічна модель (схему бази даних) наведено на Рисунок 2.

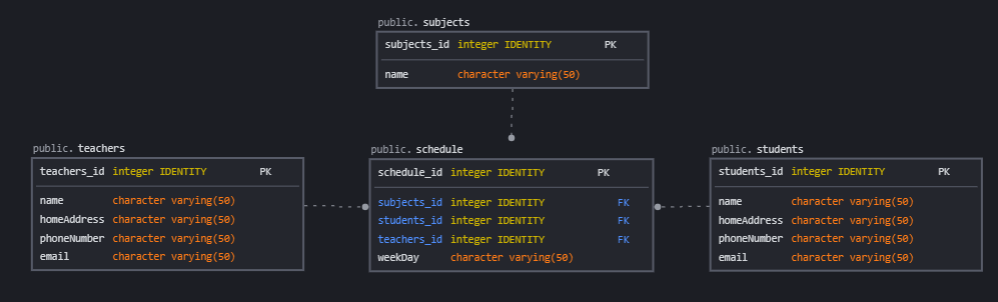


Рисунок 2 – Схема бази даних (інструмент sqldmb.com)

**Опис об’єктів бази даних у вигляді таблиці**

Опис сутностей та їх атрибутів наведено в Таблиця 1.

Таблиця 1 – опис структури бази даних «Школа».

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **subjects** – *містить усі фільм, що є у навчальній програмі* | **subjects\_id** – *унікальний ідентифікатор предмета. Не допускає NULL*  **name** – *назва предмету. Не допускає NULL* | **integer** (*числовий*)  **character varying** (*рядок*) |
| **teachers** – *містить інформацію про викладача* | **teachers\_id** – *унікальний ідентифікатор викладача. Не допускає NULL*  **name** – *ім’я викладача. Не допускає NULL*  **homeAddress** – *адреса викладача. Не допускає NULL*  **phoneNumber** – *номер телефону викладача. Не допускає NULL*  **email** – *електронна пошта викладача* | **integer** (*числовий*)  **character varying** (*рядок*)  **character varying** (*рядок*)  **character varying** (*рядок*)  **character varying** (*рядок*) |
| **students** – *містить інформацію про учня* | **students\_id** – *унікальний ідентифікатор учня. Не допускає NULL*  **name** – *ім’я викладача. Не допускає NULL*  **homeAddress** – *адреса викладача. Не допускає NULL*  **phoneNumber** – *номер телефону викладача. Не допускає NULL*  **email** – *електронна пошта викладача* | **integer** (*числовий*)  **character varying** (*рядок*)  **character varying** (*рядок*)  **character varying** (*рядок*)  **character varying** (*рядок*) |
| **schedule –** *містить інформацію про розклад* | **schedule\_id** – *унікальний ідентифікатор розкладу. Не допускає NULL*  **subjects\_id** – *унікальний ідентифікатор предмета. Не допускає NULL*  **teachers\_id** – *унікальний ідентифікатор викладача. Не допускає NULL*  **students\_id** – *унікальний ідентифікатор учня. Не допускає NULL*  **weekDay** ­ *день тижня розкладу. Не допускає NULL* | **integer** (*числовий*)  **integer** (*числовий*)  **integer** (*числовий*)  **integer** (*числовий*)  **character varying** (*рядок*) |

**Функціональні залежності для кожної таблиці**

SUBJECTS

subjects\_id → name (назва залежить від id предмету)

TEACHERS

teachers\_id → name (назва залежить від id викладача)

teachers\_id → homeAddress (домашня адреса залежить від id викладача)

teachers\_id → phoneNumber (номер телефону залежить від id викладача)

teachers\_id → email (електронна пошта залежить від id викладача)

STUDENTS

students\_id → name (назва залежить від id учня)

students\_id → homeAddress (домашня адреса залежить від id учня)

students\_id → phoneNumber (номер телефону залежить від id учня)

students\_id → email (електронна пошта залежить від id учня)

SCHEDULE

schedule\_id → subjects\_id (id предмету залежить від id розкладу)

schedule\_id → teachers\_id (id викладача залежить від id розкладу)

schedule\_id → students\_id (id учня залежить від id розкладу)

schedule\_id → weekDay (день тижня залежить від id розкладу)

**Відповідність схеми бази даних до третьої нормальної форми**

Схема відповідає 1НФ, тому що всі комірки таблиці зберігають атомарні значення і кожен рядок є унікальний

Схема відповідає 2НФ, тому що вона відповідає 1НФ і всі неключові атрибути функціонально залежать від цілого ключа, а не його частини.

Схема відповідає 3НФ, тому що вона відповідає 2НФ і всі неключові атрибути таблиці функціонально залежать від первинного ключа

**Фізична модель БД «Школа» у pgAdmin4**

Фізичну модель бази даних наведено на Рисунок 3.

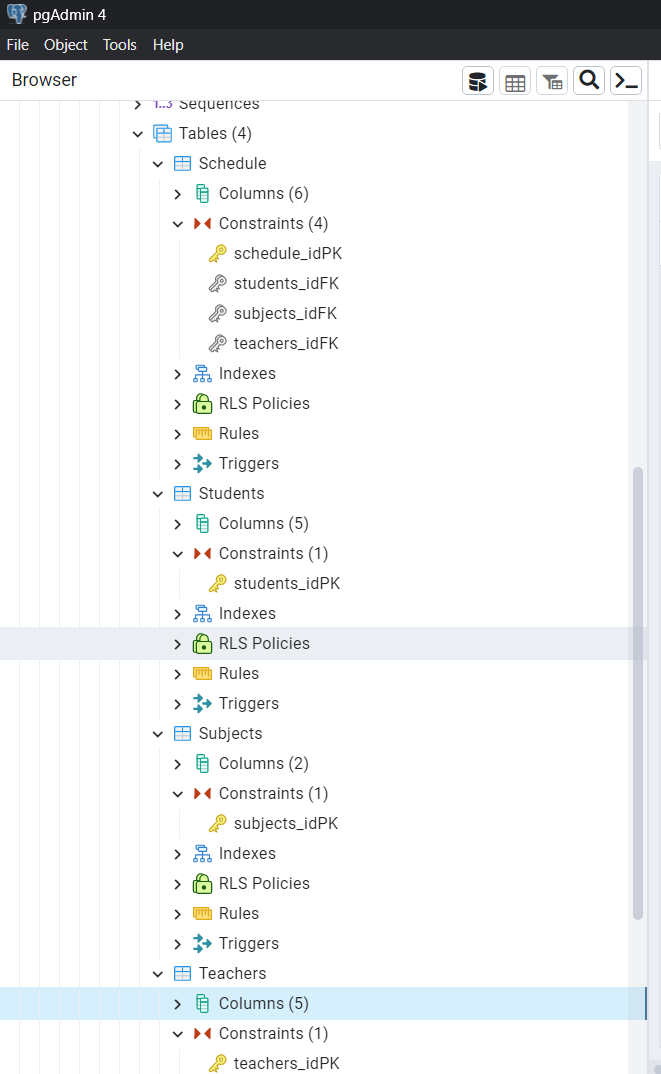


Рисунок 3 – Схема бази даних (інструмент pgAdmin4)

**Фотографії вмісту таблиць**

Вміст таблиць бази даних зображено на Рисунок 4

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 4.1 – вміст таблиці «Subjects»

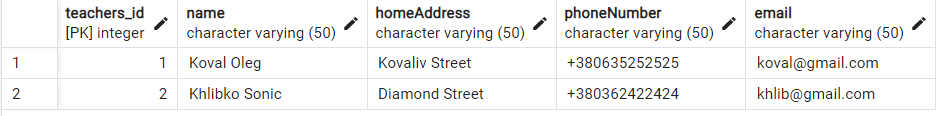


Рисунок 4.2 – вміст таблиці «Teachers»

Зображення, що містить текст

Автоматично згенерований опис

Рисунок 4.3 – вміст таблиці «Students»

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 4.4 – вміст таблиці «Schedule»

**SQL-текст опису БД «Школа»**

1. School (Database)

*-- Database: School*

*-- DROP DATABASE IF EXISTS "School";*

**CREATE** **DATABASE** "School"

**WITH**

**OWNER** = postgres

**ENCODING** = 'UTF8'

LC\_COLLATE = 'Ukrainian\_Ukraine.1251'

LC\_CTYPE = 'Ukrainian\_Ukraine.1251'

TABLESPACE = pg\_default

**CONNECTION** **LIMIT** = -1

IS\_TEMPLATE = **False**;

1. Subjects (Table)

*-- Table: public.Subjects*

*-- DROP TABLE IF EXISTS public."Subjects";*

**CREATE** **TABLE** IF **NOT** **EXISTS** **public**."Subjects"

(

subjects\_id integer **NOT** **NULL** **DEFAULT** 'nextval('"Subjects\_subjects\_id\_seq"'::regclass)',

name character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

**CONSTRAINT** "subjects\_idPK" **PRIMARY** **KEY** (subjects\_id)

)

TABLESPACE pg\_default;

**ALTER** **TABLE** IF **EXISTS** **public**."Subjects"

**OWNER** **to** postgres

1. Teachers (Table)

*-- Table: public.Teachers*

*-- DROP TABLE IF EXISTS public."Teachers";*

**CREATE** **TABLE** IF **NOT** **EXISTS** **public**."Teachers"

(

teachers\_id integer **NOT** **NULL** **DEFAULT** 'nextval('"Teachers\_teachers\_id\_seq"'::regclass)',

name character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

"homeAddress" character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

"phoneNumber" character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

email character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

**CONSTRAINT** "teachers\_idPK" **PRIMARY** **KEY** (teachers\_id)

)

TABLESPACE pg\_default;

**ALTER** **TABLE** IF **EXISTS** **public**."Teachers"

**OWNER** **to** postgres;

1. Students (Table)

*-- Table: public.Students*

*-- DROP TABLE IF EXISTS public."Students";*

**CREATE** **TABLE** IF **NOT** **EXISTS** **public**."Students"

(

students\_id integer **NOT** **NULL** **DEFAULT** 'nextval('"Students\_students\_id\_seq"'::regclass)',

name character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

"homeAddress" character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

"phoneNumber" character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

email character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

**CONSTRAINT** "students\_idPK" **PRIMARY** **KEY** (students\_id)

)

TABLESPACE pg\_default;

**ALTER** **TABLE** IF **EXISTS** **public**."Students"

**OWNER** **to** postgres;

1. Schedule (Table)

*-- Table: public.Schedule*

*-- DROP TABLE IF EXISTS public."Schedule";*

**CREATE** **TABLE** IF **NOT** **EXISTS** **public**."Schedule"

(

schedule\_id integer **NOT** **NULL** **DEFAULT** 'nextval('"Schedule\_schedule\_id\_seq"'::regclass)',

subjects\_id integer **NOT** **NULL**,

teachers\_id integer **NOT** **NULL**,

students\_id integer **NOT** **NULL**,

"weekDay" character varying(50) **COLLATE** pg\_catalog."default" **NOT** **NULL**,

**CONSTRAINT** "schedule\_idPK" **PRIMARY** **KEY** (schedule\_id),

**CONSTRAINT** "students\_idFK" **FOREIGN** **KEY** (students\_id)

**REFERENCES** **public**."Students" (students\_id) **MATCH** **FULL**

**ON** **UPDATE** **CASCADE**

**ON** **DELETE** **CASCADE**,

**CONSTRAINT** "subjects\_idFK" **FOREIGN** **KEY** (subjects\_id)

**REFERENCES** **public**."Subjects" (subjects\_id) **MATCH** **FULL**

**ON** **UPDATE** **CASCADE**

**ON** **DELETE** **CASCADE**,

**CONSTRAINT** "teachers\_idFK" **FOREIGN** **KEY** (teachers\_id)

**REFERENCES** **public**."Teachers" (teachers\_id) **MATCH** **FULL**

**ON** **UPDATE** **CASCADE**

**ON** **DELETE** **CASCADE**

)

TABLESPACE pg\_default;

**ALTER** **TABLE** IF **EXISTS** **public**."Schedule"

**OWNER** **to** postgres;

*-- Index: fki\_subject\_idFK*

*-- DROP INDEX IF EXISTS public."fki\_subject\_idFK";*

**CREATE** **INDEX** IF **NOT** **EXISTS** "fki\_subject\_idFK"

**ON** **public**."Schedule" **USING** btree

(subjects\_id **ASC** NULLS **LAST**)

TABLESPACE pg\_default;