

Лабораторна робота
з предмету Базы даних
з теми: “Проектування бази даних та
ознайомлення з базовими операціями СУБД
PostgreSQL”

Підготувала студентка
групи КП-03
Катрук Катерина
Варіант 5

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 кожен 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

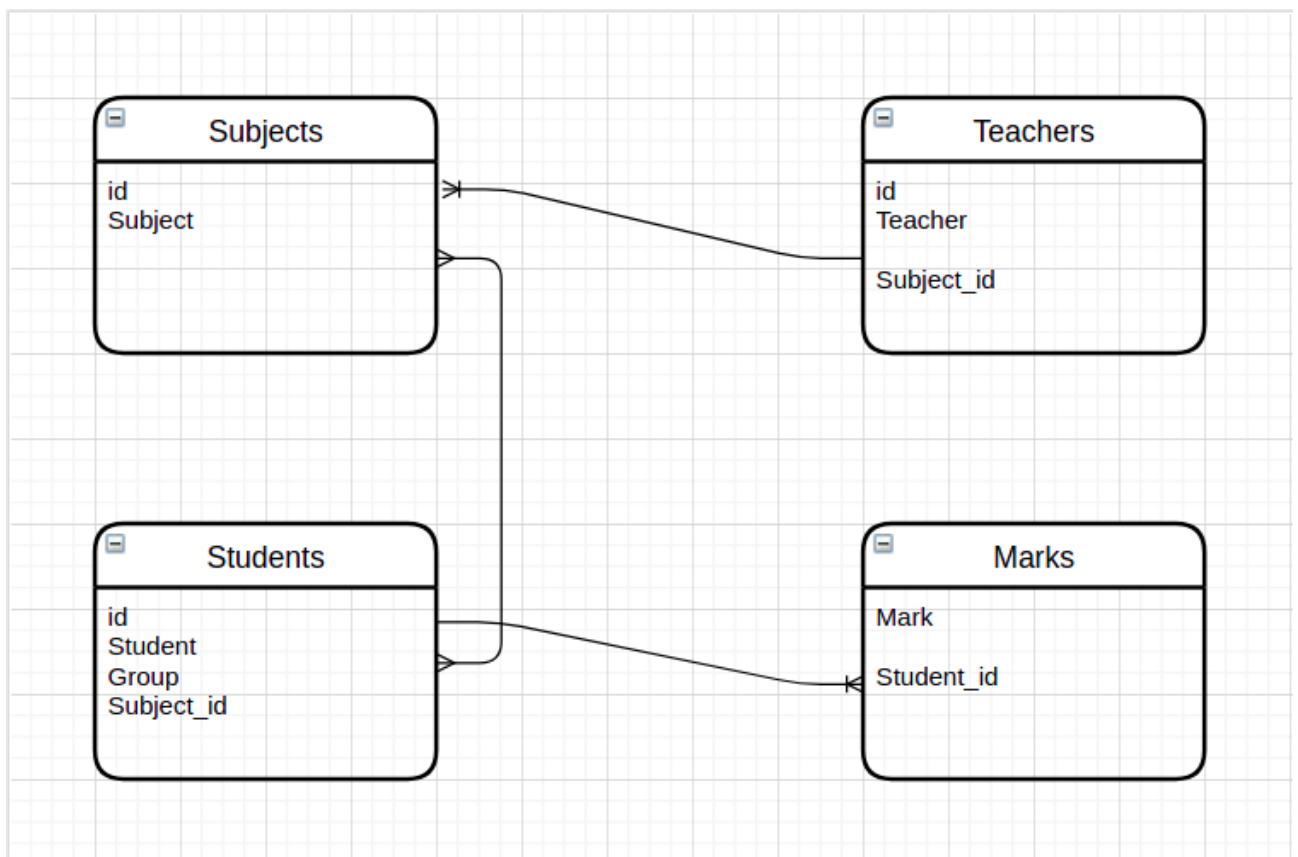
1. Створення ER-діаграм: Google Docs (Drawing) або <https://www.draw.io/> або <https://www.lucidchart.com>
2. Середовище для створення таблиць відлагодження SQL-запитів до бази даних – pgAdmin 4.
3. СУБД - PostgreSQL 10 або 11.

Дисципліни «Бази даних та засоби управління» та «Бази даних». Осінь 2021 року

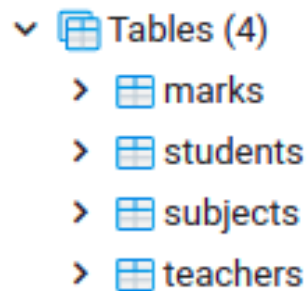
Вимоги до оформлення лабораторної роботи у електронному вигляді
Опис лабораторної роботи у репозиторії GitHub включає: назву лабораторної роботи, варіант студента, графічний файл розробленої моделі «сутність-зв'язок», структуру нормалізованої бази даних з назвами таблиць та зв'язками між ними, 2-3 копії екранних форм (screenshots) вмісту таблиць бази даних з pgAdmin4.

Виконання

У даній лабораторній роботі ми розробили модель «сутність-зв'язок» предметної галузі “школа”



Перетворили розроблену модель у схему бази даних PostgreSQL та нормалізували її до 3 нормальної форми



На прикладі SQL коду покажемо, які зв'язки між таблицями

Таблиця subjects:

```
CREATE TABLE IF NOT EXISTS public.subjects
(
    id integer NOT NULL,
    subject text COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT subjects_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE public.subjects
    OWNER to postgres;
```

Таблиця teachers:

```
CREATE TABLE IF NOT EXISTS public.teachers
(
    id integer NOT NULL,
    teacher text COLLATE pg_catalog."default" NOT NULL,
    subject_id integer NOT NULL,
    CONSTRAINT teachers_pkey PRIMARY KEY (id),
    CONSTRAINT fk_subjects FOREIGN KEY (subject_id)
        REFERENCES public.subjects (id) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
    )

TABLESPACE pg_default;

ALTER TABLE public.teachers
    OWNER to postgres;
```

Таблиця students:

```
CREATE TABLE IF NOT EXISTS public.students
(
    id integer NOT NULL,
    student text COLLATE pg_catalog."default" NOT NULL,
    "group" text COLLATE pg_catalog."default" NOT NULL,
    subject_id integer NOT NULL,
    CONSTRAINT students_pkey PRIMARY KEY (id),
    CONSTRAINT fk_subject FOREIGN KEY (subject_id)
        REFERENCES public.subjects (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.students
    OWNER to postgres;
```

Таблиця marks:

```
CREATE TABLE IF NOT EXISTS public.marks
(
    id integer NOT NULL,
    mark integer NOT NULL,
    student_id integer NOT NULL,
    CONSTRAINT marks_pkey PRIMARY KEY (id),
    CONSTRAINT fk_student FOREIGN KEY (student_id)
        REFERENCES public.students (id) MATCH SIMPLE
```

```
ON UPDATE NO ACTION  
ON DELETE NO ACTION  
NOT VALID
```

)

```
TABLESPACE pg_default;
```

```
ALTER TABLE public.marks  
OWNER to postgres;
```

Також ми ознайомились із інструментарієм PostgreSQL та pgAdmin 4 та внесли декілька рядків даних у кожен з таблиць засобами pgAdmin4

subjects

	Id [PK] integer	subject text
1	1	Math
2	2	Biology
3	3	English

teachers

	Id [PK] integer	teacher text	subject_id integer
1	1	Maryna Kolesnik	1
2	2	Mykhailo Berkovec	3
3	3	Donald Trump	2

students

	Id [PK] integer	student text	group text	subject_id integer
1	1	Ihor Fedyshyn	KM-04	3
2	2	Kris Halushko	TA-67	2
3	3	Mykyta Soloduha	KM-04	3

marks

	Id [PK] integer	mark integer	student_id integer
1	1	15	1
2	2	13	2
3	3	40	3

Висновки

Отже, ми здобули навички проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL. Навчилися нормалізовувати базу даних до 3 нормальної форми, користуватись pgAdmin та створювати зв'язки між таблицями бази даних.