



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

**Факультет прикладної математики
Кафедра системного програмування і спеціалізованих комп'ютерних
систем**

**Лабораторна робота № 1
з дисципліни “Бази даних і засоби управління”**

**Виконав
студент III курсу
групи КВ-84**

Антонюк АІ

Київ 2020

Варіант

Університет (Студенти , спеціальності , факультети)

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 «сущність-зв'язок»;
- назва нотації.

У звіті щодо пункту №2 завдання має бути:

- опис процесу перетворення (наприклад, “сущність 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 зображені обов'язково **повинні мати назву!**

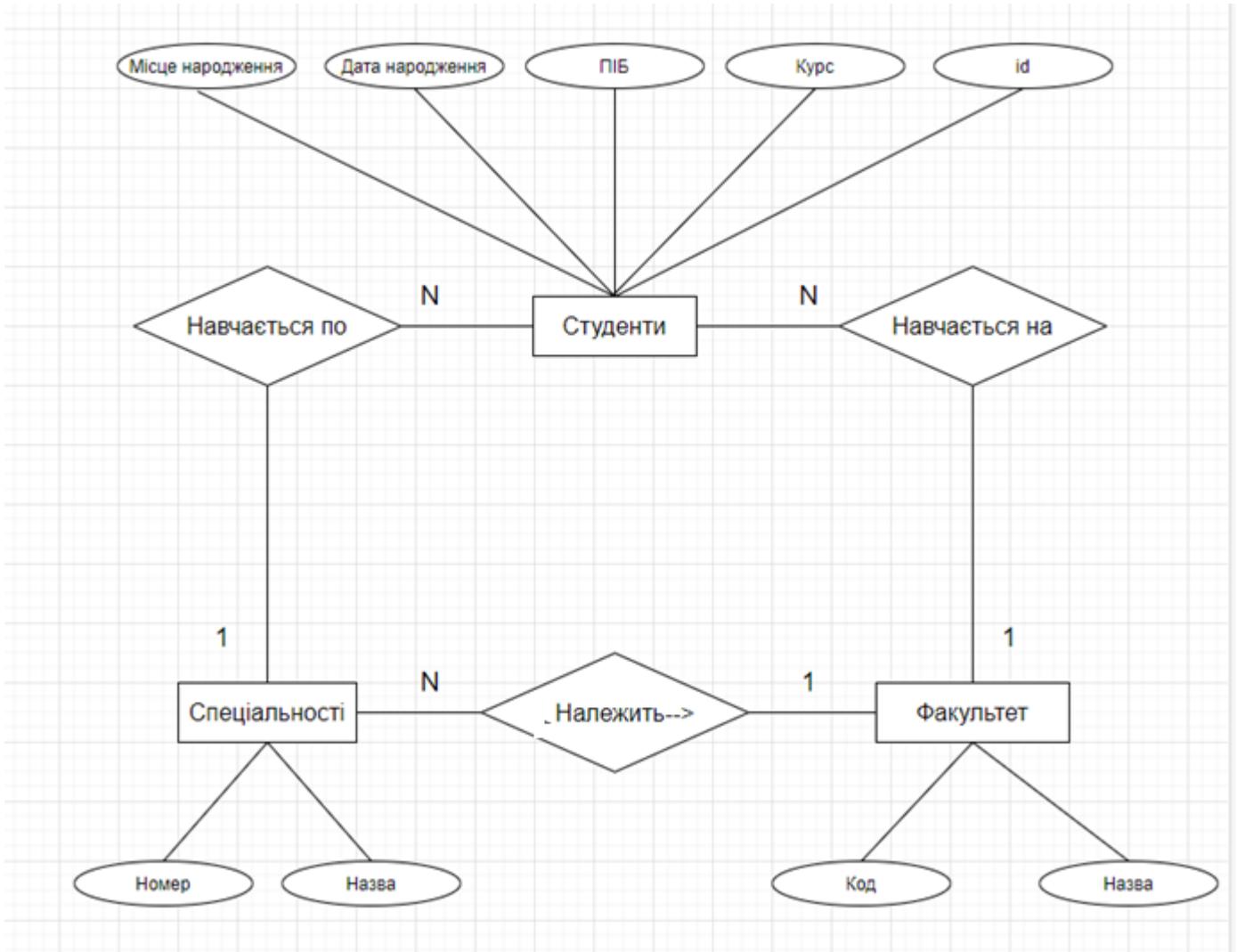
Звіт щодо пункту 1

Сутність «Студенти» - студенти, що навчаються в університеті.

Сутність «Спеціальності» - спеціальності на яких навчаються студенти.

Сутність «Факультет» - облік факультетів, які містить університет.

Графічний файл розробленої моделі «сутність-зв'язок»



Назва нотації: нотація Чена

Звіт щодо пункту 2

Сутності перетворено у таблиці відповідно до їх назв.

Схема бази даних у графічному вигляді



Відношення	Атрибут	Тип(розмір)
Відношення “Студенти”	<p>Id – унікальний id студента. Не допускає NULL</p> <p>ПІБ – прізвище, ім’я, по батькові студента. Не допускає NULL</p> <p>Дата народження – дата народження студента. Не допускає NULL</p> <p>Курс – номер курсу, на якому навчається студент. Не допускає NULL</p> <p>Місце народження – місто, де народився студент. Не допускає NULL</p>	<p>Текстовий</p> <p>Serial(Integer)</p> <p>Temporal type(date)</p> <p>Serial(Integer)</p> <p>Текстовий</p>
Відношення “Спеціальність”	<p>Номер – унікальний номер спеціальності. Не допускає NULL</p> <p>Назва – назва спеціальності, може інколи повторюватись. Не допускає NULL</p>	<p>Serial(Integer)</p> <p>Текстовий</p>
Відношення “Факультет”	<p>КОД – унікальний ID факультету. Не допускає NULL</p> <p>Назва – унікальна назва факультету. Не допускає NULL</p>	<p>Serial(Integer)</p> <p>Текстовий</p>

Звіт щодо пункту 3

Спеціальності – Студенти: кожен студент навчається по певній спеціальності, на кожній спеціальності може бути багато студентів.

Факультет – Студенти: кожен студент відноситься до певного факультету, кожен факультет має багато студентів .

Спеціальності – Факультети: кожна спеціальність належить певному факультету, одному факультету може належати декілька спеціальностей.

Функціональні залежності

Спеціальності

Номер -> назва: знаючи номер спеціальності, знаємо її назву.

Факультет

Код -> назва: знаючи код факультету, можемо взнати назву цього факультету, назва факультету унікальна.

Студенти

Id->ПІБ: знаючи id студента знаємо його ПІБ, ПІБ може повторюватись(дуже рідко).

Id->курс: знаючи id знаєм курс, на кожному курсі навчається багато студентів.

Id->дата народження: знаючи id знаєм дату народження, вона може повторюватись в різних студентів.

Id->місце народження: знаючи id знаєм місце народження студента, багато студентів народились в одному і тому ж місті.

Схема бази даних відповідає 1НФ, тому що передбачає лише одне значення атрибути у кожній комірці.

Схема відповідає 2НФ, тому що немає ключів, які складаються з двох і більше атрибутів.

Схема відповідає 3НФ, тому що немає транзитивних функціональних залежностей між ключовими та неключовими атрибутами.

Звіт щодо пункту 4

The screenshot shows the structure of the 'department' table. The table has five columns: stud_id (PK integer), pip (character varying(150)), course (integer), br_date (date), and place (character varying(50)). The data consists of three rows:

stud_id	pip	course	br_date	place
1	Sergienko_Anatoliy_Sergiyovu...	3	2000-12-12	Kyiv
2	Avgustenko Petr Sergiyovich	2	2002-10-10	Kherson
3	Proba_Vasil_Mukolayovich	4	1998-11-11	Cherkasy

The screenshot shows the structure of the 'specialization' table. The table has two columns: number (PK integer) and name (character varying(50)). The data consists of three rows:

number	name
73	Management
123	Computer_engineering
133	System_analysis

The screenshot shows the structure of the 'students' table. The table has five columns: stud_id (PK integer), pip (character varying(150)), course (integer), br_date (date), and place (character varying(50)). The data consists of three rows:

stud_id	pip	course	br_date	place
1	Sergienko_Anatoliy_Sergiyovu...	3	2000-12-12	Kyiv
2	Avgustenko Petr Sergiyovich	2	2002-10-10	Kherson
3	Proba_Vasil_Mukolayovich	4	1998-11-11	Cherkasy

