



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 KB-84
Антонюк А І

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 «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!**

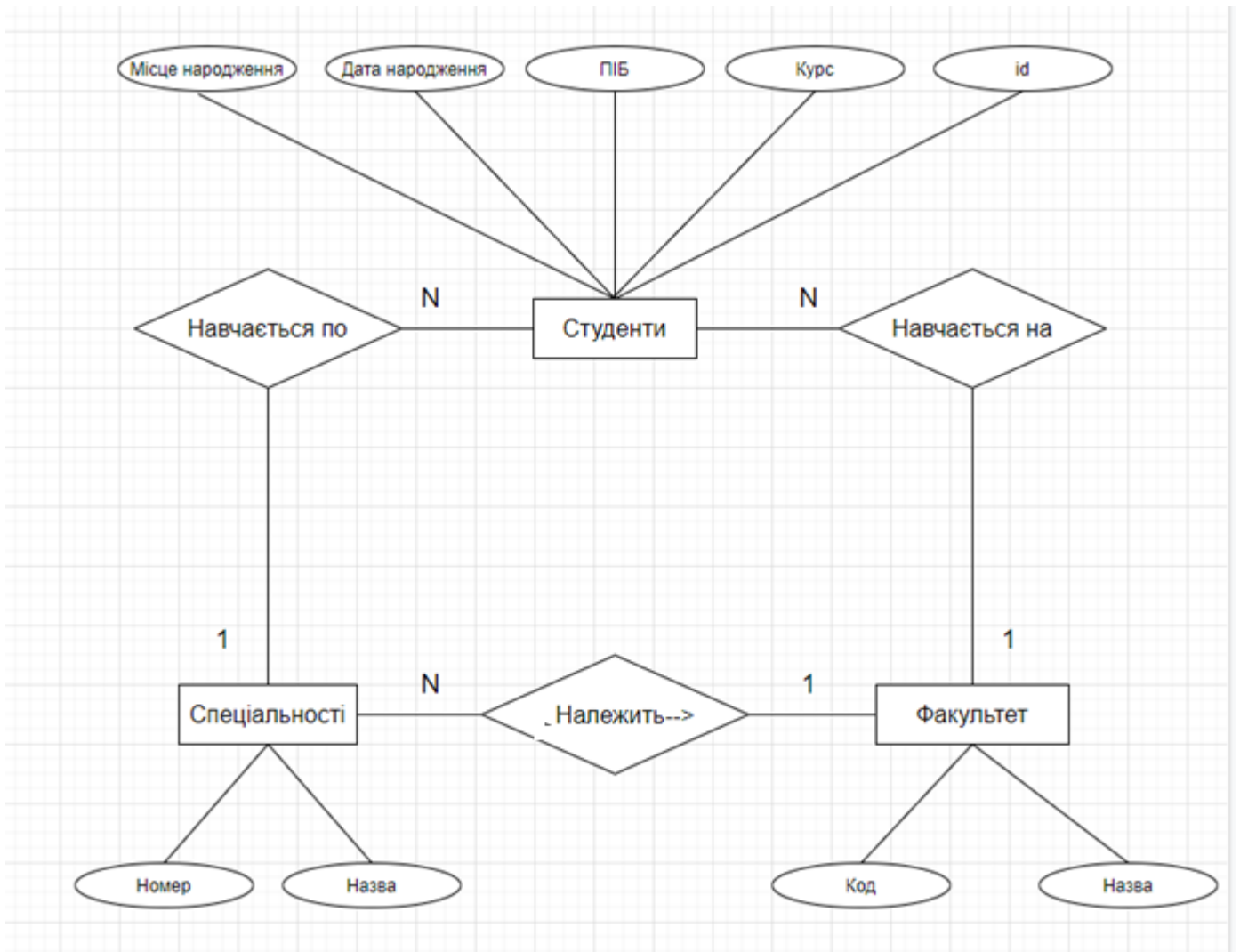
Звіт щодо пункту 1

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»

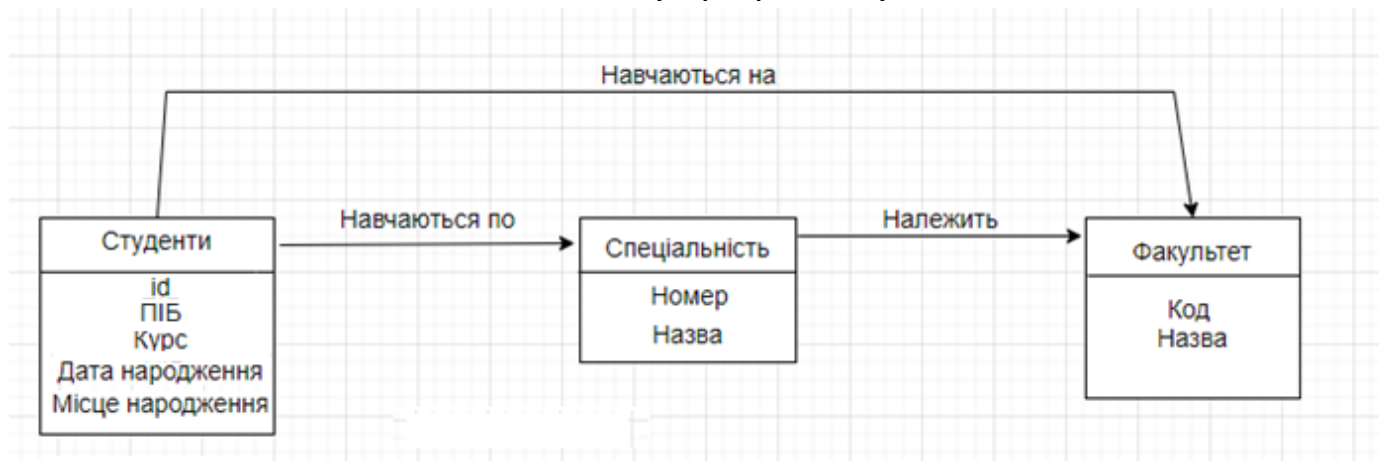


Назва нотації: нотація Чена

Звіт щодо пункту 2

Сутності перетворено у таблиці відповідно до їх назв.

Схема бази даних у графічному вигляді



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

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 атрибута 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

▼ department

Columns

▼ Constraints (1)

department_pkey

Indexes

RLS Policies

Rules

Triggers

specialization

students

Data Output

Explain

Messages

Notifications

	stud_id [PK] integer	pip character varying (150)	course integer	br_date date	place character varying (50)
1		1 Sergienko_Anatoliy_Sergiyovu...		3 2000-12-12	Kyiv
2		2 Avgustenko Petr Sergiyovich		2 2002-10-10	Kherson
3		3 Proba_Vasil_Mukolayovich		4 1998-11-11	Cherkasy

> department	
▼ specialization	
> Columns	
▼ Constraints (2)	
FK specialization	
specialization_pkey	
> Indexes	
> RLS Policies	
> Rules	
> Triggers	
> students	

Data Output	Explain	Messages	Notifications
number [PK] integer	name character varying (50)		
1	73 Management		
2	123 Computer_engineering		
3	133 System_analysis		

Tables (3)

department

specialization

students

Columns (5)

Constraints (3)

FK students

FK students2

students_pkey

Indexes

RLS Policies

Rules

Triggers

Data Output

Explain

Messages

Notifications

stud_id [PK] integer	pip character varying (150)	course integer	br_date date	place character varying (50)
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

