

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ
Кафедра системного програмування та спеціалізованих комп'ютерних
систем**

Лабораторна робота №1

з дисципліни «Бази даних та засоби управління»

**Тема: «Проектування бази даних та ознайомлення з базовими
операціями СУБД PostgreSQL»**

Виконав студент групи : КВ-23

ПІБ : Корольов Юрій Ігорович

Перевірив: _____

Київ 2024

Лабораторна робота № 1.

Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL

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 кожному з таблиць засобами pgAdmin 4.

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 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

Варіант 13

“Система управління персоналом компанії”

Пункт 1. Розробити модель «сутність-зв'язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».

Сутність “Працівники” – містить перелік всіх працівників компанії, має наступні атрибути:

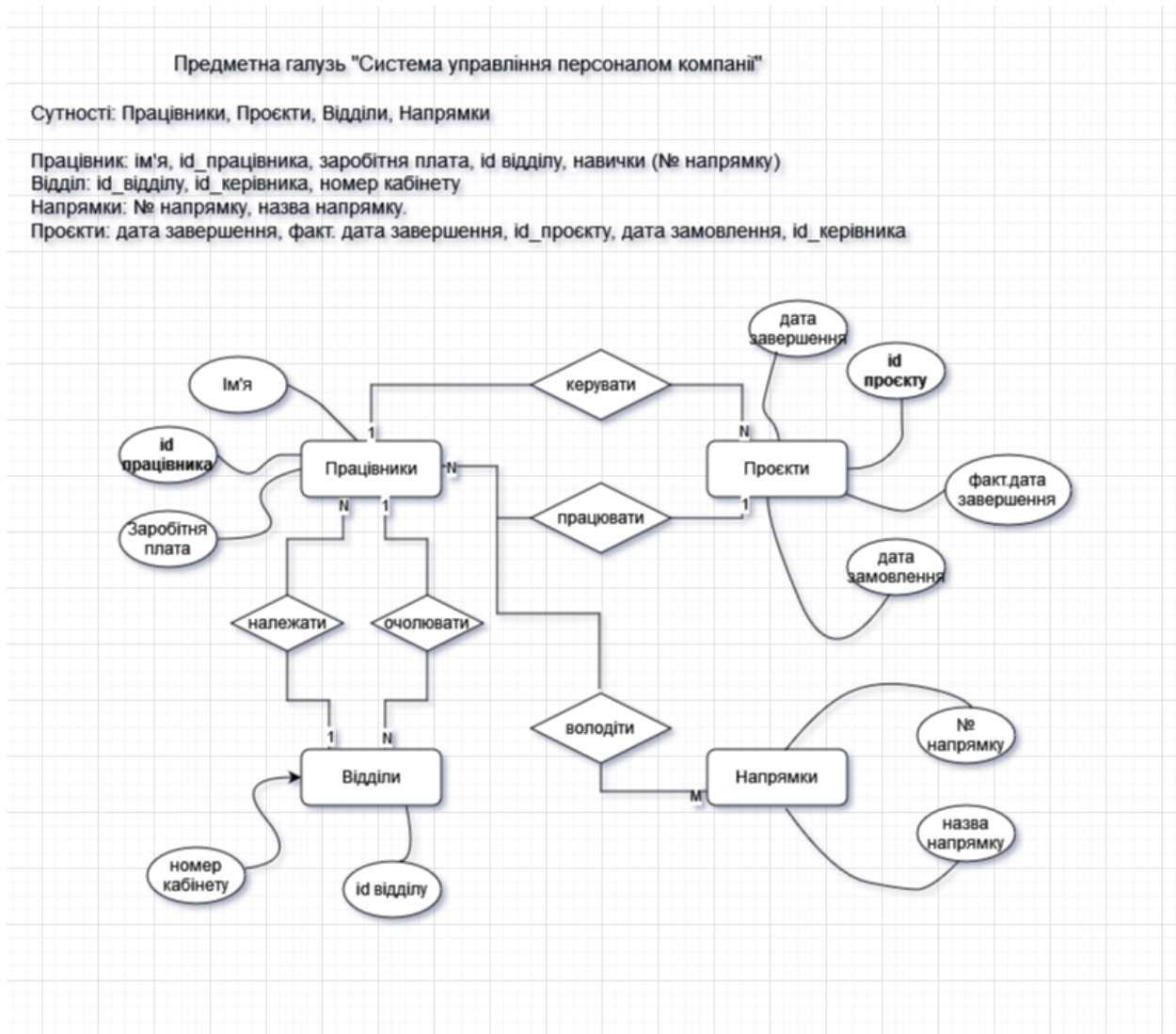
ім'я, навички (№напрямку), **id працівника**, заробітня плата, id відділу.

Сутність “Відділи” – облік всіх відділів, має наступні атрибути:

id відділу, id керівника (працівника), номер кабінету.

Сутність “Навички” – перелік напрямків, за якими може працювати робітник (у даній моделі робітник може працювати тільки за одним напрямком), містить наступні атрибути: **№ навички**, назва напрямку.

Сутність “Проекти” – перелік проектів якими займається компанія має наступні атрибути: id керівника проекту, **id проекту**, дата замовлення, дата завершення, фактична дата завершення.



Назва нотації: нотація Чена.

Зв'язок "керувати" між сутностями "Працівники" і "Проєкти".

- Зв'язок **обов'язковий** для сутності "Проєкти": кожний проєкт пов'язаний з певним керівником.
- Зв'язок **необов'язковий** для сутності "Працівники": працівник не обов'язково має бути керівником певного проєкту.

Отже, зв'язок типу **1 : N (required:optional)**.

Зв'язок "працювати" між сутностями "Працівники" і "Проєкти":

- Зв'язок **необов'язковий** для сутності “Працівники”: працівник може не працювати над жодним проєктом в певний проміжок часу.
- Зв'язок є **необов'язковим** для сутності “Проекти”: на виконання проєкту може не бути відведено ні одного працівника

Отже, обидва опціональні, зв'язок типу **1 : N (both are optional)**.

Зв'язок “*володіти*” між сутностями “Працівники” та “Напрямки”:

- Зв'язок є **необов'язковим** для сутності “Працівники”: працівник може не мати жодного напрямку.
- Зв'язок є **необов'язковим** для сутності “Напрямки”: напрямком може не володіти жоден із працівників.

Отже, обидва опціональні, зв'язок типу **M : N (both are optional)**.

Зв'язок “*очолювати*” між сутностями “Працівники” та “Відділи”:

- Зв'язок є **необов'язковим** для сутності “Працівники”: працівник не зобов'язаний бути керівником відділу
- Зв'язок є **обов'язковим** для сутності “Відділи”: відділ завжди має мати керівника.

Отже, маємо зв'язок типу **1 : N (required:optional)**.

Зв'язок “*належати*” між сутностями “Працівники” та “Відділи”:

- Зв'язок є **обов'язковим** для сутності “Працівники”: кожен працівник має належати до певного відділу.
- Зв'язок є **обов'язковим** для сутності “Відділи”: відділ має містити хоча б одного працівника.

Отже, маємо зв'язок типу **1 : N (both are required)**.

Пункт 2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.

Сутність “Працівники” було перетворено в таблицю “Employees”; Сутність “Напрямки” було перетворено на таблицю “Directions”; Сутність “Проекти”

було перетворено на таблицю “Projects”; Сутність “Відділи” було перетворено на таблицю “Departments”.

Зв’язок “Employees” -> *належати* -> “Departments” зумовив появу додаткового атрибуту department_id в таблиці “Employees”, який є зовнішнім ключем для цієї таблиці.

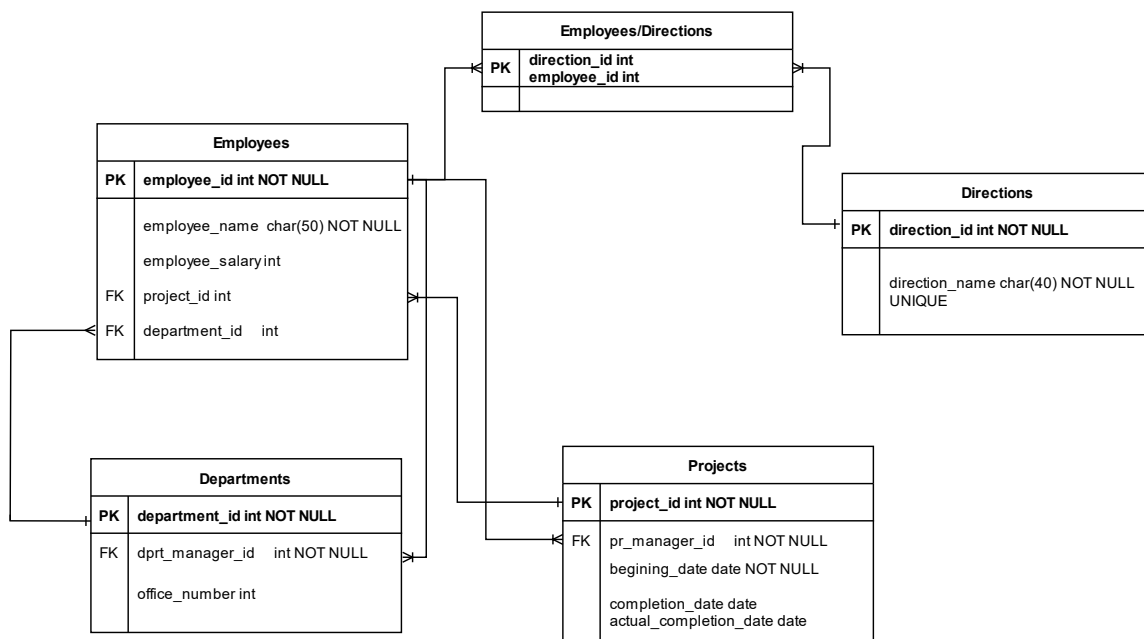
Зв’язок “Employees” -> *очолювати* -> “Departments” зумовив появу атрибуту dprt_manager_id, який є зовнішнім ключем в таблиці “Departments”.

Зв’язок “Employees” -> *працювати* -> “Projects” зумовив появу атрибуту project_id, який є зовнішнім ключем в таблиці “Employees”.

Зв’язок “Employees” -> *керувати* -> “Projects” зумовив появу атрибуту pr_manager_id, який є зовнішнім ключем в таблиці “Projects”.

Зв’язок “Employees” -> *володіти* -> “Directions” зумовив появу додаткової таблиці “Employees/Directions”, ключовий атрибут якої складається з двох атрибутів: direction_id, employee_id. Це було зроблено для реалізації зв’язку M:N (1 or both optional).

Перетворена ER діаграма у таблиці БД.



Пункт 3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).

Визначимо функціональні залежності в таблицях:

1. Employees

employee_id -> employee_name, employee_salary, project_id, department_id

employee_id -> employee_name

employee_id -> employee_salary

employee_id -> project_id

employee_id -> department_id

ID працівника визначає його ім'я, заробітню плату, ID проєкту над яким він працює та ID відділу.

2. Departments

department_id -> dprt_manager_id, office_number

department_id -> dprt_manager_id

department_id -> office_number

ID відділу визначає номер офісу та ідентифікатор його керівника. Оскільки номер офісу прив'язаний до одного конкретного відділу, номер офісу в даному випадку може виступати в якості первинного ключа та визначати ідентифікатор відділу та ідентифікатор керівника.

3. Projects

project_id -> pr_manager_id, beginning_date, completion_date, actual_completion_date

project_id -> beginning_date

project_id -> completion_date

project_id -> actual_completion_date

project_id -> pr_manager_id

Ідентифікатор проєкту визначає ідентифікатор керівника, дату початку, дату завершення та фактичну дату завершення.

4. Directions

direction_id -> direction_name

direction_name -> direction_id

В даному випадку маємо два потенційних ключа.

Перевіримо належність бази даних до 1, 2 та 3 нормальних форм.

Вимоги **1 нормальної форми 1NF**:

- Зберігати **атомарність** значення атрибуту.
- Кожен рядок має бути **унікальним**.

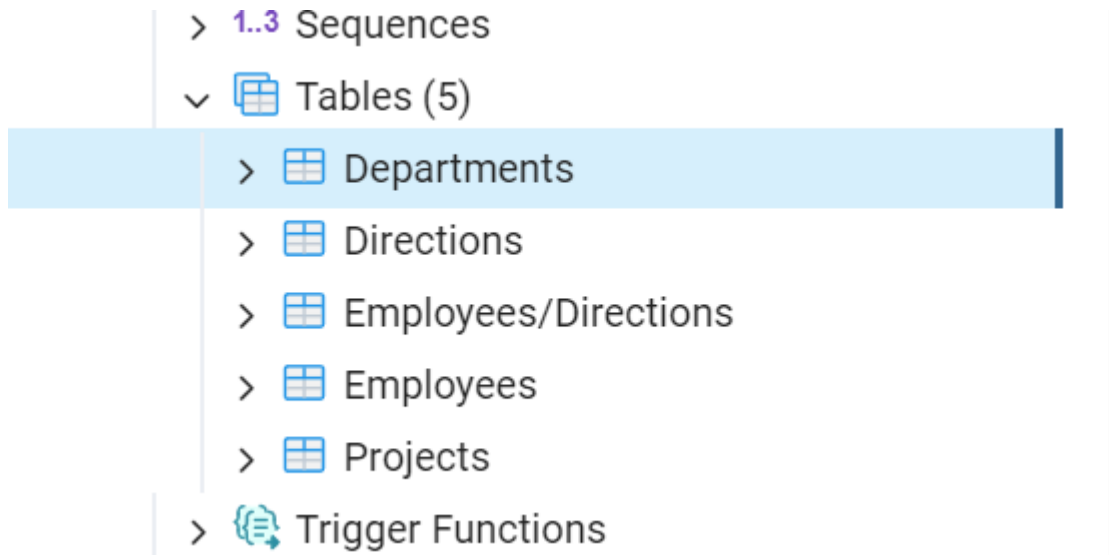
Вимоги **2 нормальної форми 2NF**:

- Таблиця повинна бути в 1 нормальній формі.
- Кожен не ключовий атрибут повинен **функціонально залежати від цілого ключа**, а не його частини.

Вимоги **3 нормальної форми 3NF**:

- Таблиця або відношення R повинна знаходитись в 2 нормальній формі
- Кожен непервинний атрибут з цього відношення або таблиці є не транзитивно залежний від кожного кандидатного ключа від R.
- 1NF: *Схема бази даних відповідає 1NF* тому що передбачає атомарність значень атрибутів у кожній комірці та не передбачає дублікатних рядків.
- 2NF: Схема бази даних відповідає 1NF та не містить складних первинних ключів. *Отже, схема відповідає 2NF.*
- 3NF: Схема бази даних відповідає 2NF. Також кожен непервинний атрибут в кожній із таблиць не є транзитивно залежним від жодного з кандидатних(потенційних) ключів. *Отже, схема бази даних відповідає 3NF.*

Пункт 4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.



1. Таблиця Projects

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈



	project_id [PK] integer	begining_date date	completion_date date	actual_completion_date date	pr_manager_id integer
1	2	2024-10-10	2024-10-13	2024-10-14	2
2	1	2024-10-14	[null]	[null]	1

Columns

+

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	project_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text"/>
	begining_date	date			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
	completion_da	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
	actual_complet	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
	pr_manager_id	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>

Primary Key Foreign Key Check Unique Exclude

				+
	Name	Columns	Referenced Table	
 	Projects_pr_manager_id_fkey	(pr_manager_id) -> (employee_id)	public.Employees	





2. Таблиця Directions

Data Output Messages Notifications



	direction_id [PK] integer	direction_name "char[]"
1	1	{"C++ development"}
2	2	{"Java development"}

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	direction_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
 	direction_name	"char[]"			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

3. Таблиця Employees

Data Output Messages Notifications



	employee_id [PK] integer	employee_name "char[]"	employee_salary integer	project_id integer	department_id integer
1	1	{"Boris"}	118	1	2
2	2	{"Kevin"}	1000	2	1

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	employee_id	integer					<input type="text"/>
	employee_name	"char[]"					<input type="text"/>
	employee_salary	integer					<input type="text"/>
	project_id	integer					<input type="text"/>
	department_id	integer					<input type="text"/>

Primary Key

Foreign Key

Check

Unique

Exclude

	Name	Columns	Referenced Table	
	Employees_department_id_fkey	(department_id) -> (department_id)	public.Departments	
	Employees_project_id_fkey	(project_id) -> (project_id)	public.Projects	

4. Таблиця Employees/Directions

Data Output

Messages

Notifications

<

Columns								+
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default	
		direction_id	integer v				<input type="text"/>	
		employee_id	integer v				<input type="text"/>	

				+
	Name	Columns	Referenced Table	
		Employees/Directions_direction_id_fk	(direction_id) -> (direction_id)	
		Employees/Directions_employee_id_fk	(employee_id) -> (employee_id)	

5. Таблиця Departments

Data Output

Messages

Notifications

	department_id [PK] integer	dprt_manager_id integer	office_number integer
1	1	1	29
2	2	1	17

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
		department_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		dprt_manager_id	integer			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
		office_number	integer			<input type="checkbox"/>	<input type="checkbox"/>	

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns	Referenced Table
		Departments_dprt_manager_id_fkey	(dprt_manager_id) -> (employee_id)	public.Employees

ПОСИЛАННЯ НА РЕПОЗИТОРІЙ GitHub:

https://github.com/YuryKorolyov/Data_bases