**Міністерство освіти і науки України**

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

**ЛАБОРАТОРНА РОБОТА №1**

***із дисципліни***

***“******Бази даних та засоби управління”***

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

**Виконав**

**студент III курсу**

**ФПМ групи КВ-03**

**Недашківський Д. О.**

**Перевірив(ла)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

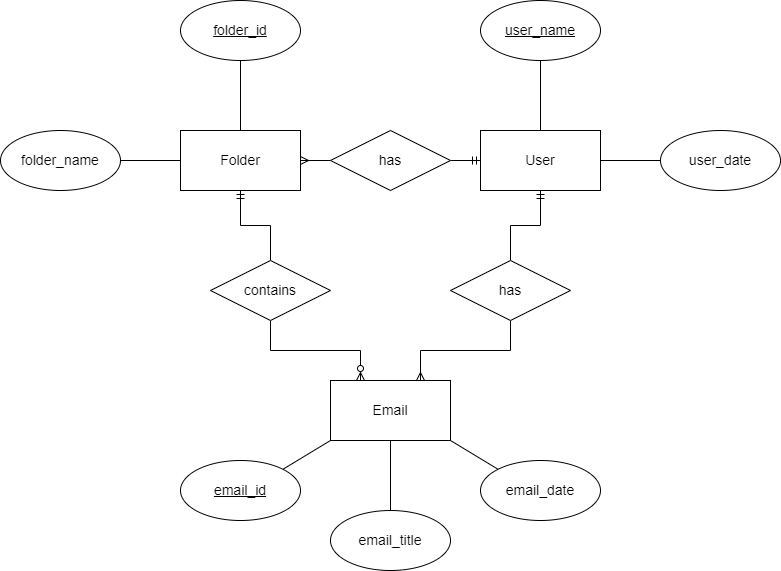
**Київ – 2022**

**Завдання**

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.

**Хід роботи**

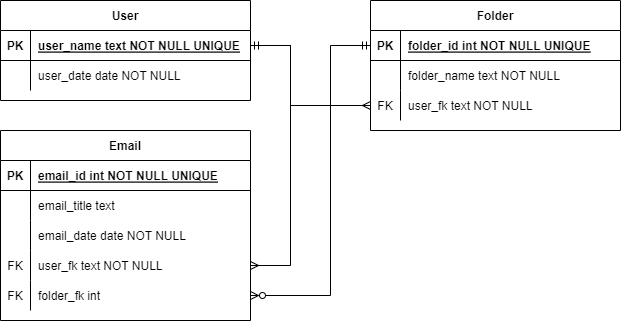
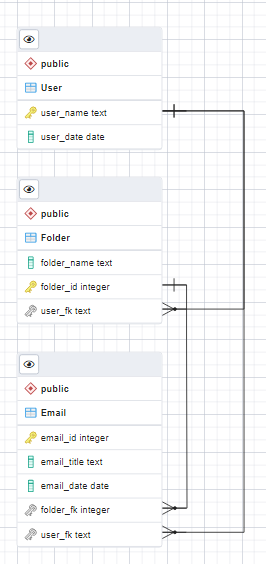
Діаграма ER-моделі умовної електронної пошти:



Наявні сутності:

* Сутність User – описує користувачів електронної пошти. Кожен користувач має унікальне ім’я та дату створення облікового запису. Один користувач може мати багато папок та листів.
* Сутність Folder – описує папки, які мають користувачі та містять листи. Кожна папка має ім’я та унікальний ID-номер. Одна папка може містити багато листів.
* Сутність Email – описує електронні листи, які мають користувачі, та які можуть лежати в папках. Кожен лист має заголовок, дату створення та унікальний ID-номер. Один лист знаходиться в одного користувача, і може знаходитися в папці.

ER-модель, перетворена в схему бази даних PostgreSQL для pgAdmin 4:



Опис процесу перетворення:

Сутність User було перетворено в таблицю User.  
Сутність Folder було перетворено в таблицю Folder. Зв’язок has (1:N) із сутністю User зумовив появу зовнішнього ключа user\_fk.

Сутність Email було перетворено в таблицю Email. Зв’язки has (1:N) та contains (1:N) із сутностями User і Folder відповідно зумовили появи зовнішніх ключів user\_fk та folder\_fk відповідно.

Нормалізація схеми бази даних:

Схема відповідає 1НФ, оскільки всі атрибути є атомарними та кожен запис є унікальним.

Функціональні залежності схеми:

User (user\_name, user\_date)

user\_name → user\_date (визначення первинного ключа)

Folder (folder\_id, folder\_name, user\_fk)

folder\_id → folder\_name, user\_fk (визначення первинного ключа)

folder\_id → folder\_name (F5)  
folder\_id → user\_fk (F5)

Email (email\_id, email\_title, email\_date, user\_fk, folder\_fk)

email\_id → email\_title, email\_date, user\_fk, folder\_fk (визначення первинного ключа)

email\_id → email\_title (F5)

email\_id → email\_date (F5)

email\_id → user\_fk (F5)

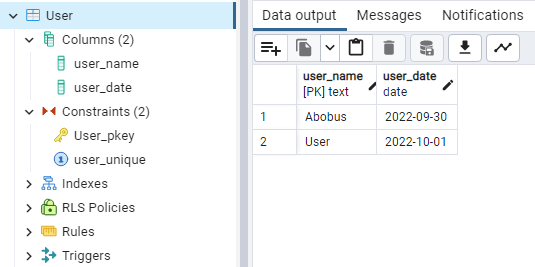
email\_id → folder\_fk (F5)

Схема відповідає 2НФ, оскільки схема відповідає 1НФ та кожен неключовий атрибут залежить від усього первинного ключа.

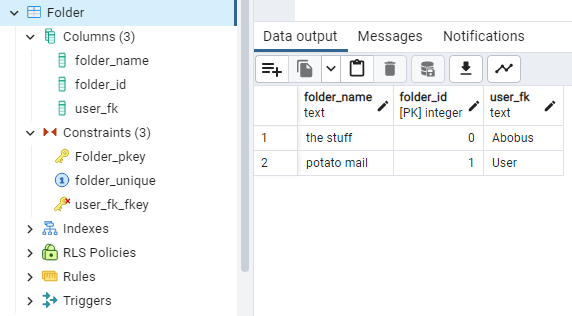
Схема відповідає 3НФ, оскільки в схемі немає залежностей між неключовими атрибутами.

Введення даних в кожну з таблиць:

User



Folder



Email

