

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Лабораторна робота №1**

з дисципліни

**«Бази даних і засоби управління»**

Тема: «**Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL та PgAdmin 4**»

Виконав: студент 3 курсу

Група: КВ-03

Дунець А.М

Перевірив: Петрашенко А.В.

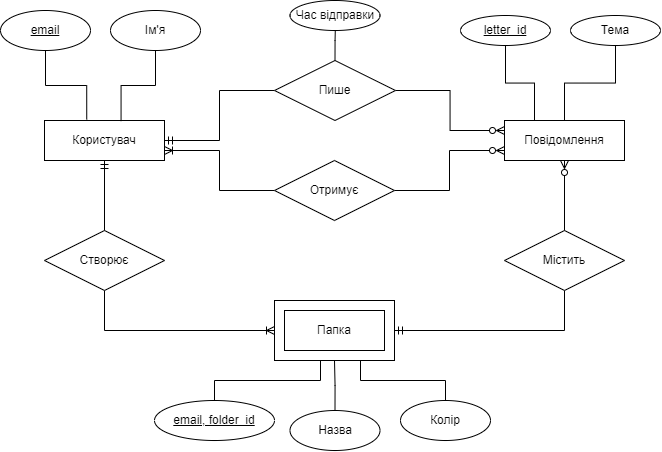
Київ – 2022

*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 кожну з таблиць засобами pgAdmin 4.

**Етап 1** - Розробка моделі “сутність-звʼязок” для предметної галузі “Електронна пошта”(нотація Crow’s foot)



*Опис моделі:*

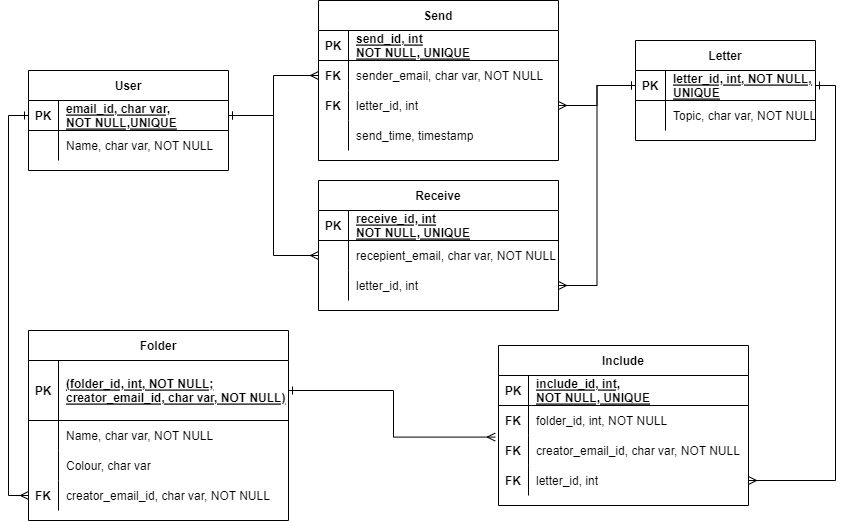
Електронна пошта має три сутності:Користувач, Повідомлення, Папка. Сутність Папка не може існувати без сутності користувач, тому вона слабка.

Сутність **Користувач** містить такі атрибути: email, Ім’я. Вона має з сутністю **Повідомлення** два зв’язки. Перший з них це зв’язок **Пише**, тип якого це 1:N(N >= 0), цей зв'язок має свій власний атрибут: час надсилання.

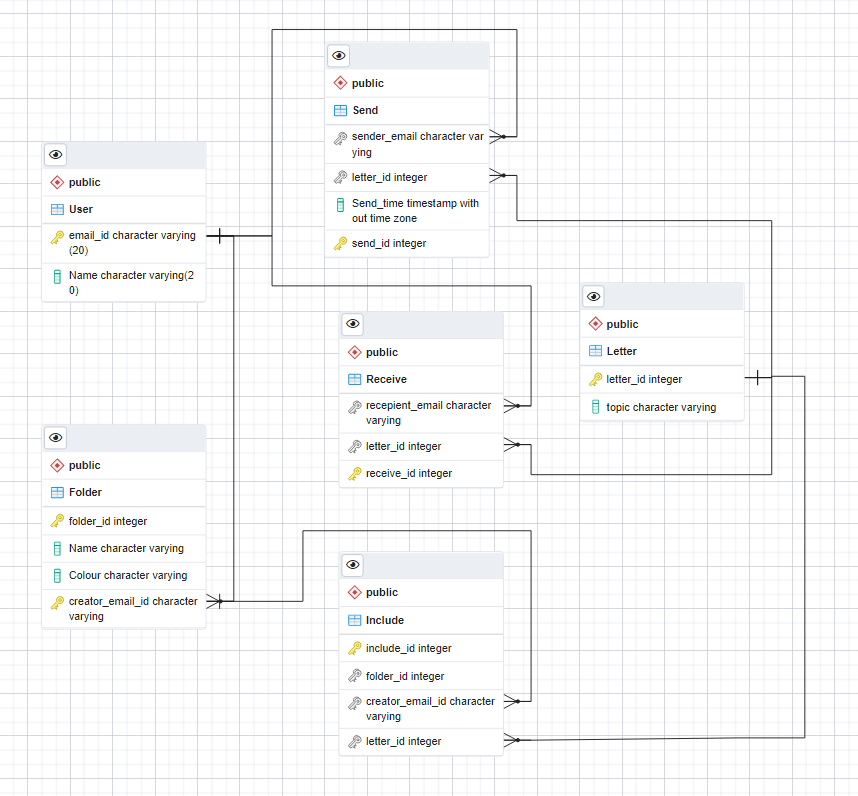
Другий зв’язок **Отримує** має тип M:N, де M > 0, N >= 0. Також **Користувач** має зв’язок **Створює** типу 1:N(N > 0) з сутністю **Папка**

Сутність **Папка** має такі 3 атрибути: складений ключовий атрибут (email, folder\_id), Назва, Колір. **Папка** має зв’язок **Містить** типу 1:N(N >= 0) з сутністю **Повідомлення**.

Сутність **Повідомлення** має 3 зв’язки, які вже були перераховані та містить такі атрибути: letter\_id, Тема.

**Етап 2** - Перетворення розробленої моделі у схему бази даних (таблиці) PostgreSQL

*Графічне перетворення*

**

*Перетворення у pgAdmin 4*

*Опис перетворення:*

Сутність **Користувач** перетворили в таблицю **User**, а звʼязок із сутністю **Папка**(перетворено в таблицю **Folder**) спровокував появу ключа creator\_email\_id в цій сутності. Зв'язки **Пише** та **Отримує** сутності **Користувач** з сутністю **Повідомлення**(перетворено в таблицю **Letter**) спричинили появу відповідних таблиць **Send** та **Receive**. Зв’язок **Містить** між сутністю **Папка** та **Повідомлення** також потребував додаткової таблиці **Include**.

**Етап 3** - Нормалізація схеми бази даних до третьої нормальної форми

*Функціональні залежності:*

Folder((**folder\_id, creator\_email\_id**), Name, colour)

(**folder\_id, creator\_email\_id**)→ name, colour

(**folder\_id, creator\_email\_id**)→ name

(**folder\_id, creator\_email\_id**)→ colour

Include(**include\_id**, folder\_id, creation\_email\_id, letter\_id)

**include\_id** → folder\_id, creation\_email\_id, letter\_id

**include\_id** → folder\_id

**include\_id** → creation\_email\_id

**include\_id** → letter\_id

Letter(**letter\_id**, topic)

**letter\_id** → topic

Receive(**receive\_id**, recipient\_email, letter\_id)

**receive\_id** → recipient\_email, letter\_id

**receive\_id** → recipient\_email

**receive\_id** → letter\_id

Send(**send\_id**, sender\_email, letter\_id, send\_time)

**send\_id** → sender\_email, letter\_id, send\_time

**send\_id** → sender\_email

**send\_id** → letter\_id

**send\_id** → send\_name

User(**email\_id**, name)

**email\_id** → name

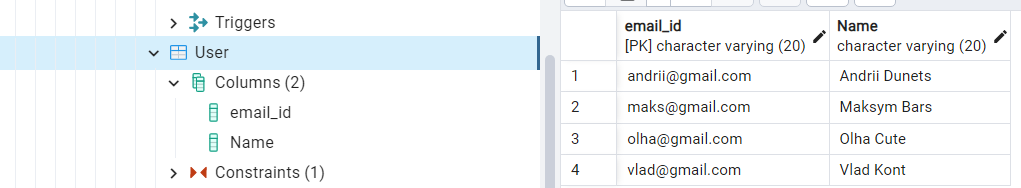
Схема бази даних відповідає 1НФ, тому що кожна таблиця має основний ключ, всі рядки унікальні(уникнено повторень груп), всі атрибути атомарні(мають по одному значенню, а не множину значень).

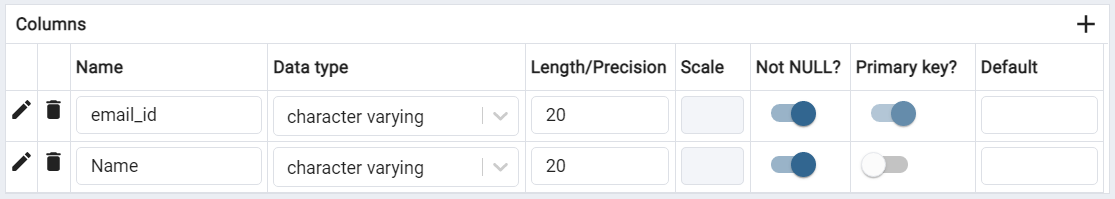
Схема бази даних відповідає 2НФ, тому що схема виконується 1НФ та всі її неключові атрибути функціонально залежні від одного ключа в цілому

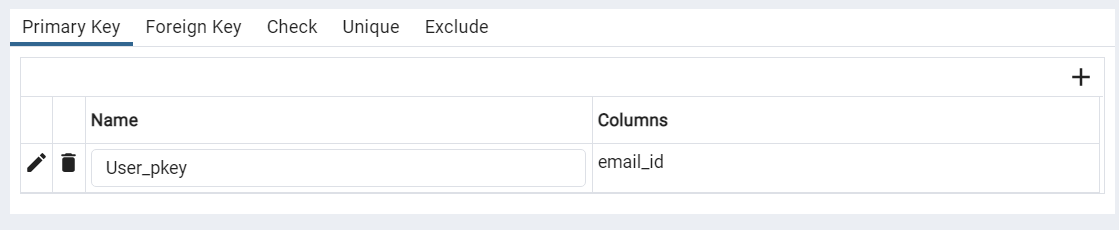
Схема бази даних відповідає 3НФ, тому що виконується 2НФ та відсутні транзитивні функціональні залежності не ключових атрибутів від ключових.

**Етап 4** - наведення скріншотів роботи з PostgreSQL та pgAdmin 4

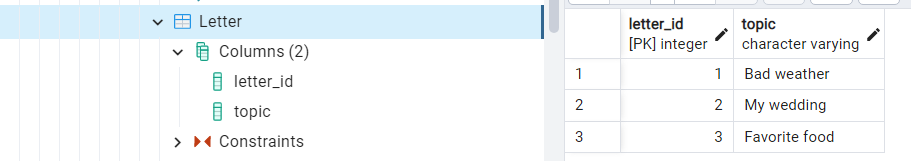
*Таблиця* ***User***

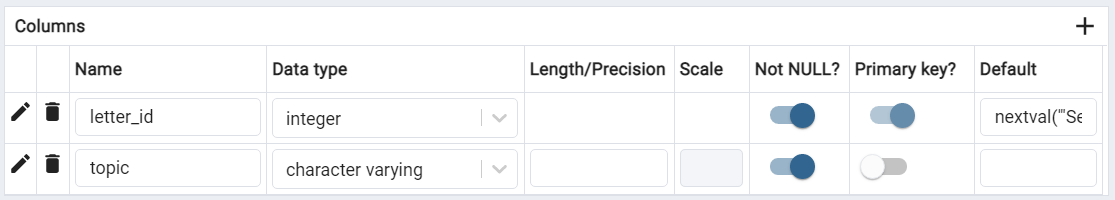






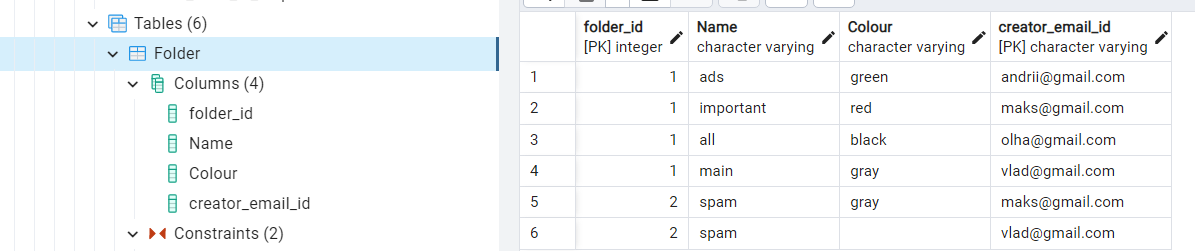
*Таблиця* ***Letter***

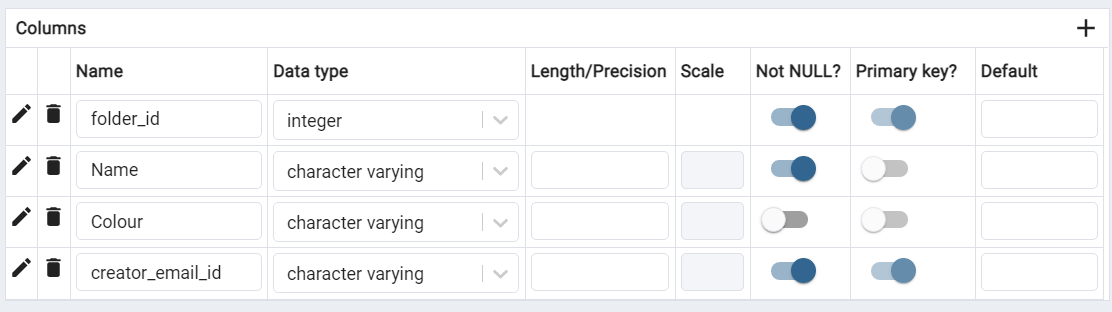
**

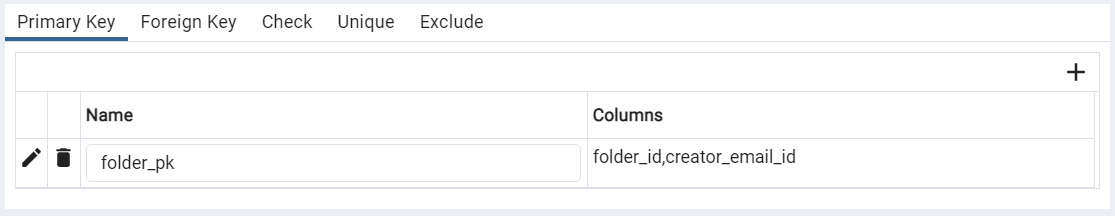


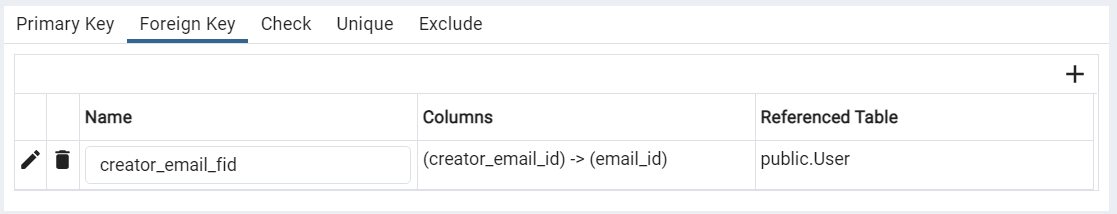


*Таблиця* ***Folder***

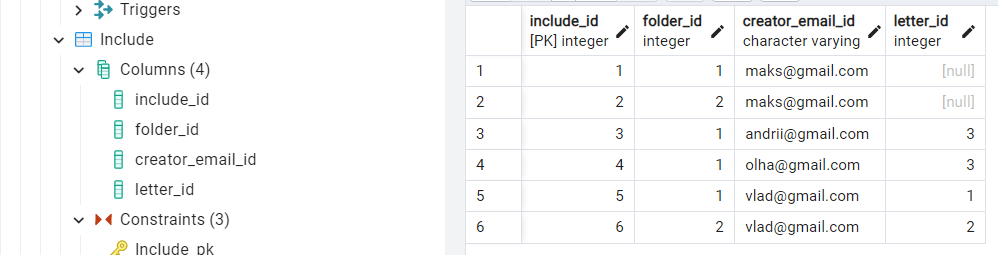


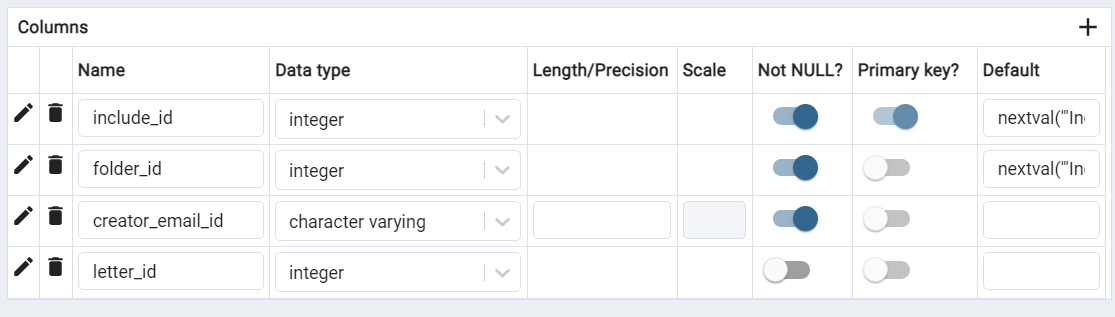


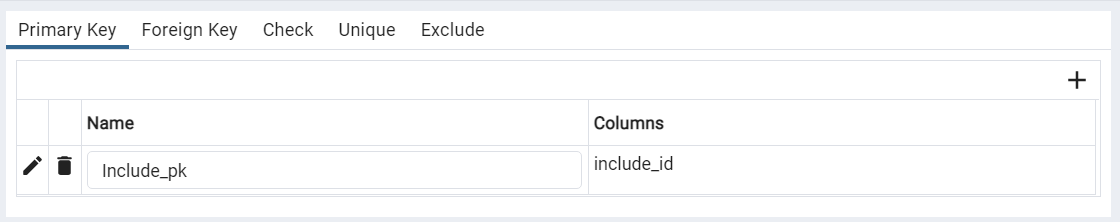




*Таблиця* ***Include***

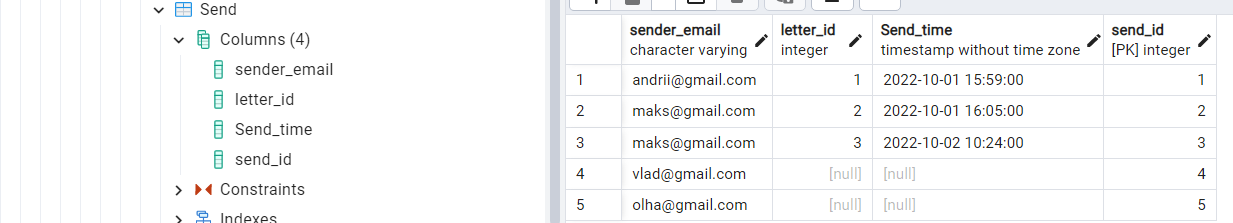


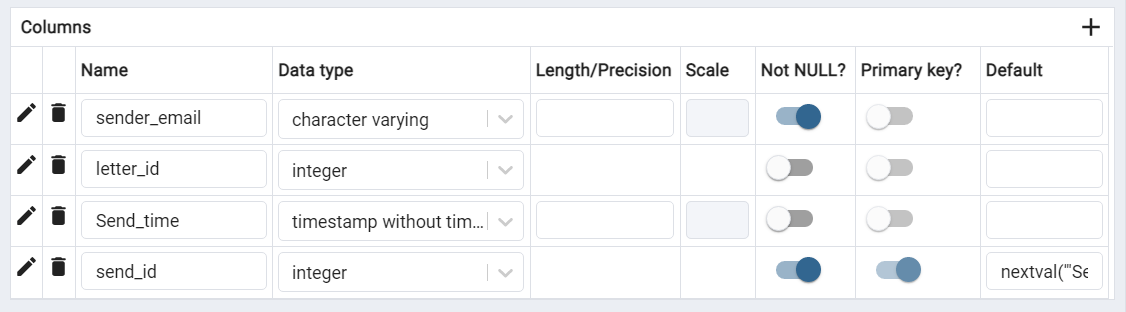




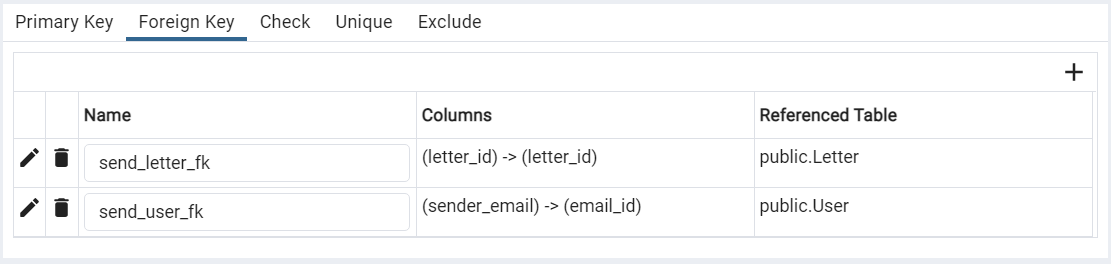


*Таблиця* ***Send***

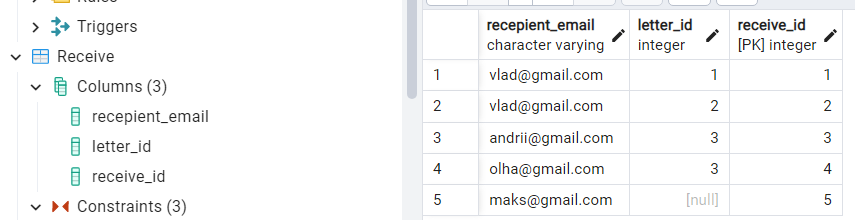


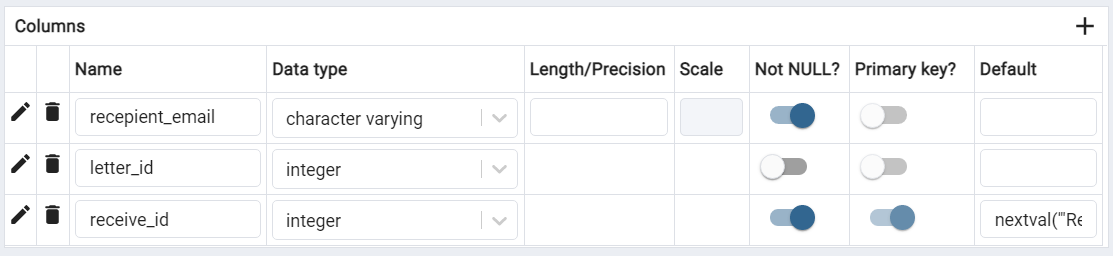


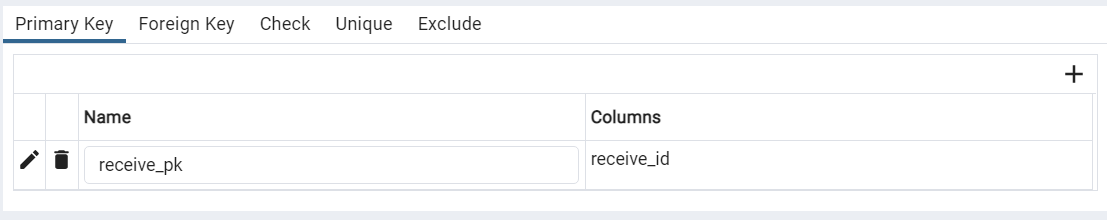




*Таблиця* ***Receive***

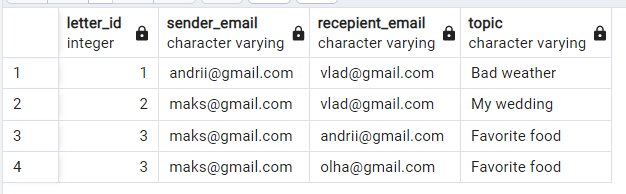


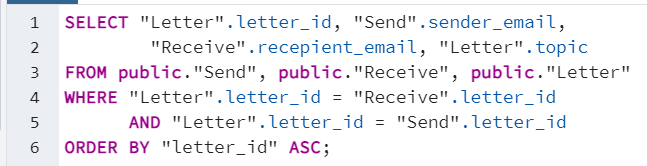






*Приклад вибору інформації з кількох таблиць*

******



Посилання на гіт: [dynetskov/dataBases (github.com)](https://github.com/dynetskov/dataBases)