

# **Лабораторна робота №3**

## **з дисципліни «Бази даних»**

**Виконали:**

Хміль Денис ІМ-44

Осипенко Гліб ІМ-44

Київ 2025

# Звіт

## Короткий звіт про остаточну схему бази даних

Цілі — Написати запити SELECT для отримання даних (включаючи фільтрацію за допомогою WHERE та вибір певних стовпців). — Практикувати використання операторів INSERT для додавання нових рядків до таблиць. — Практикувати використання оператора UPDATE для зміни існуючих рядків (використовуючи SET та WHERE). — Практикувати використання операторів DELETE для безпечноного видалення рядків (за допомогою WHERE). — Вивчити основні операції маніпулювання даними (DML) у PostgreSQL та спостерігайте за їхнім впливом.

## Скріншоти

Написати запити SELECT для отримання даних (включаючи фільтрацію за допомогою WHERE та вибір певних стовпців).

The screenshot shows the pgAdmin interface. In the top navigation bar, 'Query' is selected. Below it, a code editor contains the following SQL query:

```
1 select first_name
2 from public.customer
3 where first_name = 'Alice';
```

Below the code editor is a 'Data Output' tab, which is currently active. It displays the result of the query:

	first_name
character varying (100)	Alice

Практикувати використання операторів INSERT для додавання нових рядків до таблиць.

The screenshot shows the pgAdmin interface. In the top navigation bar, 'Query' is selected. Below it, a code editor contains the following SQL query:

```
1 INSERT INTO customer
2 (customer_id, first_name, last_name, email, phone)
3 values
4 (gen_random_uuid(), 'David', 'Brown', 'david.brown@example.com', '+11223344556');
5
6 select * from customer
```

Below the code editor is a 'Data Output' tab, which is currently active. It displays the result of the query, showing four rows of data:

customer_id	first_name	last_name	email	phone
[PK] uuid	character varying (100)	character varying (100)	character varying (255)	character varying (50)
bf13170e-85ab-415c-82f5-cbd5d830...	Alice	Johnson	alice.johnson@example.c...	+1234567890
ac8ac471-009e-40ad-89ef-6ada5190...	Bob	Smith	bob.smith@example.com	+0987654321
ab6e3a61-e234-483e-9e46-b96d3759...	Carol	Miller	carol.miller@example.com	+1122334455
0daa667f-ab7f-4484-be69-7d44fb1a7...	David	Brown	david.brown@example.com	+11223344556

Практикувати використання оператора UPDATE для зміни існуючих рядків (використовуючи SET та WHERE)

Query    Query History

```
1 UPDATE customer
2 set email = 'meneZminly@example.com'
3 where customer_id = '0daa667f-ab7f-4484-be69-7d44fb1a7909';
4
5 select * from customer;
```

Data Output    Messages    Notifications

SQL

	customer_id [PK] uuid	first_name character varying (100)	last_name character varying (100)	email character varying (255)	phone character varying (50)
1	bf13170e-85ab-415c-82f5-cbd5d830deee	Alice	Johnson	alice.johnson@example.c...	+1234567890
2	ac8ac471-009e-40ad-89ef-6ada51900bd7	Bob	Smith	bob.smith@example.com	+0987654321
3	ab6e3a61-e234-483e-9e46-b96d3759f7a6	Carol	Miller	carol.miller@example.com	+1122334455
4	0daa667f-ab7f-4484-be69-7d44fb1a7909	David	Brown	meneZminly@example.c...	+11223344556

Практикувати використання оператори DELETE для безпечноого видалення рядків (за допомогою WHERE)

Query    Query History

```
1 delete from customer
2 where customer_id = '0daa667f-ab7f-4484-be69-7d44fb1a7909';
3
4 select * from customer;
```

Data Output    Messages    Notifications

SQL

	customer_id [PK] uuid	first_name character varying (100)	last_name character varying (100)	email character varying (255)	phone character varying (50)
1	bf13170e-85ab-415c-82f5-cbd5d830deee	Alice	Johnson	alice.johnson@example.c...	+1234567890
2	ac8ac471-009e-40ad-89ef-6ada51900bd7	Bob	Smith	bob.smith@example.com	+0987654321
3	ab6e3a61-e234-483e-9e46-b96d3759f7a6	Carol	Miller	carol.miller@example.com	+1122334455

## Висновок

Під час виконання лабораторної роботи №3 я на практиці закріпив навички роботи з DML (Data Manipulation Language) у PostgreSQL.

Я навчився виконувати базові OLTP-операції:

- **SELECT:** для вибірки та фільтрації даних за допомогою WHERE.
- **INSERT:** для додавання нових рядків у таблиці.
- **UPDATE:** для модифікації існуючих даних.
- **DELETE:** для видалення конкретних рядків.

Найважливішим висновком є розуміння критичної ролі оператора WHERE при виконанні запитів UPDATE та DELETE, оскільки його відсутність призводить до зміни або повної втрати всіх даних у

таблиці. Робота також дозволила перевірити, що схема бази даних, створена в попередніх лабораторних, коректно працює з даними.