**Лабораторна робота №4**

**З дисципліни:** Бази даних та інформаційні системи  
**Студента групи МІТ-31:** Ларіонова С.О.

**Тема:** База даних CRM-системи для обслуговування клієнтів.

**Створення користувацького типу даних:**

CREATE TYPE ticket\_status AS ENUM ('open', 'pending', 'resolved');

CREATE TYPE ticket\_priority AS ENUM ('low', 'medium', 'high', 'urgent');



ALTER TABLE support\_tickets

ALTER COLUMN status TYPE ticket\_status using status::ticket\_status,

ALTER COLUMN priority TYPE ticket\_priority using priority::ticket\_priority;



**Створення користувацької функції:**

CREATE OR REPLACE FUNCTION product\_percentages\_in\_orders()

RETURNS TABLE(product\_id INT, percentage NUMERIC) AS $$

DECLARE

total\_orders NUMERIC;

BEGIN

SELECT SUM(amount) INTO total\_orders FROM orders;

RETURN QUERY

SELECT o.product\_id, ROUND(SUM(amount)/total\_orders\*100 , 2) percentage\_a

FROM orders o

GROUP BY o.product\_id

ORDER BY percentage\_a DESC;

END;

$$ LANGUAGE plpgsql;

-- Перевірка

SELECT p.name product, percentage

FROM product\_percentages\_in\_orders()

JOIN products p ON p.id = product\_id;



**Створення тригера для логування змін:**

CREATE TABLE customer\_changes (

change\_id SERIAL PRIMARY KEY,

customer\_id INT,

operation CHAR(1),

changed\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP);

CREATE OR REPLACE FUNCTION customer\_changes() RETURNS TRIGGER AS $$

BEGIN

INSERT INTO customer\_changes (customer\_id, operation)

VALUES (NEW.id, substring(TG\_OP from 1 for 1));

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER track\_customer\_changes

AFTER INSERT OR UPDATE OR DELETE ON customers

FOR EACH ROW

EXECUTE FUNCTION customer\_changes();

-- Перевірка

INSERT INTO customers(name, last\_name, email, phone, address) VALUES

('Emily', 'Davis', 'emily.davis@example.com', '456-789-0123', '321 Pine St, Thatown, USA'),

('Michael', 'Brown', 'michael.brown@example.com', '567-890-1234', '654 Maple St, Yourtown, USA');

UPDATE customers

SET email = 'new.email@example.com', phone = '987-654-3210'

WHERE id = 105;

DELETE FROM customers WHERE id = 104;

SELECT \* FROM customer\_changes;



**Оновлення ER-діаграми:**

* Додано нову сутність customer\_changes
* Додано зв’язок між customer\_changes та customers

**Висновки:** В цій лабораторній роботі було реалізовано користувацькі типи даних, користувацьку функцію та тригер для відстеження змін у таблиці. Використання цих засобів SQL допомагає пришвидшити роботу з базами даних та дозволяє краще контролювати які саме дані зберігаються та як вони змінюються для запобігання помилок та підтримки стабільної роботи системи.

**Додатки:**

Оновлена ER-діаграма у файлі er\_diagram.pdf.