Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

З лабораторної роботи № 3 з дисципліни

«Технології розробки вбудованих IoT системи»

„АДАПТЕРИ У PYTHON. СТВОРЕННЯ ТА ІНТЕГРАЦІЯ ВЛАСНОГО АДАПТЕРА”

**Перевірив**

(прізвище, ім'я, по батькові)

Київ 2023

1. Посилання на git repo: <https://github.com/faaant/iot>
2. Реалізація класу адаптера та методу `save\_data` адаптера:

import json

import logging

from typing import List

import pydantic\_core

import requests

from app.entities.processed\_agent\_data import ProcessedAgentData

from app.interfaces.store\_gateway import StoreGateway

class StoreApiAdapter(StoreGateway):

    def \_\_init\_\_(self, api\_base\_url):

        self.api\_base\_url = api\_base\_url

    def save\_data(self, processed\_agent\_data\_batch: List[ProcessedAgentData]):

        """

        Save the processed road data to the Store API.

        Parameters:

            processed\_agent\_data\_batch (dict): Processed road data to be saved.

        Returns:

            bool: True if the data is successfully saved, False otherwise.

        """

        try:

            # Prepare the data as a list of dictionaries

            data = []

            for item in processed\_agent\_data\_batch:

                # Convert datetime to ISO 8601 format string

                timestamp\_isoformat = item.agent\_data.timestamp.isoformat()

                # Create a dictionary with timestamp as string

                data\_item = item.model\_dump()

                data\_item['agent\_data']['timestamp'] = timestamp\_isoformat

                data.append(data\_item)

            # Make the POST request

            response = requests.post(f"{self.api\_base\_url}/processed\_agent\_data/", json=data)

            # Check if the request was successful (status code 200)

            if response.status\_code == 200 or response.ok :

                return True

            else:

                return False

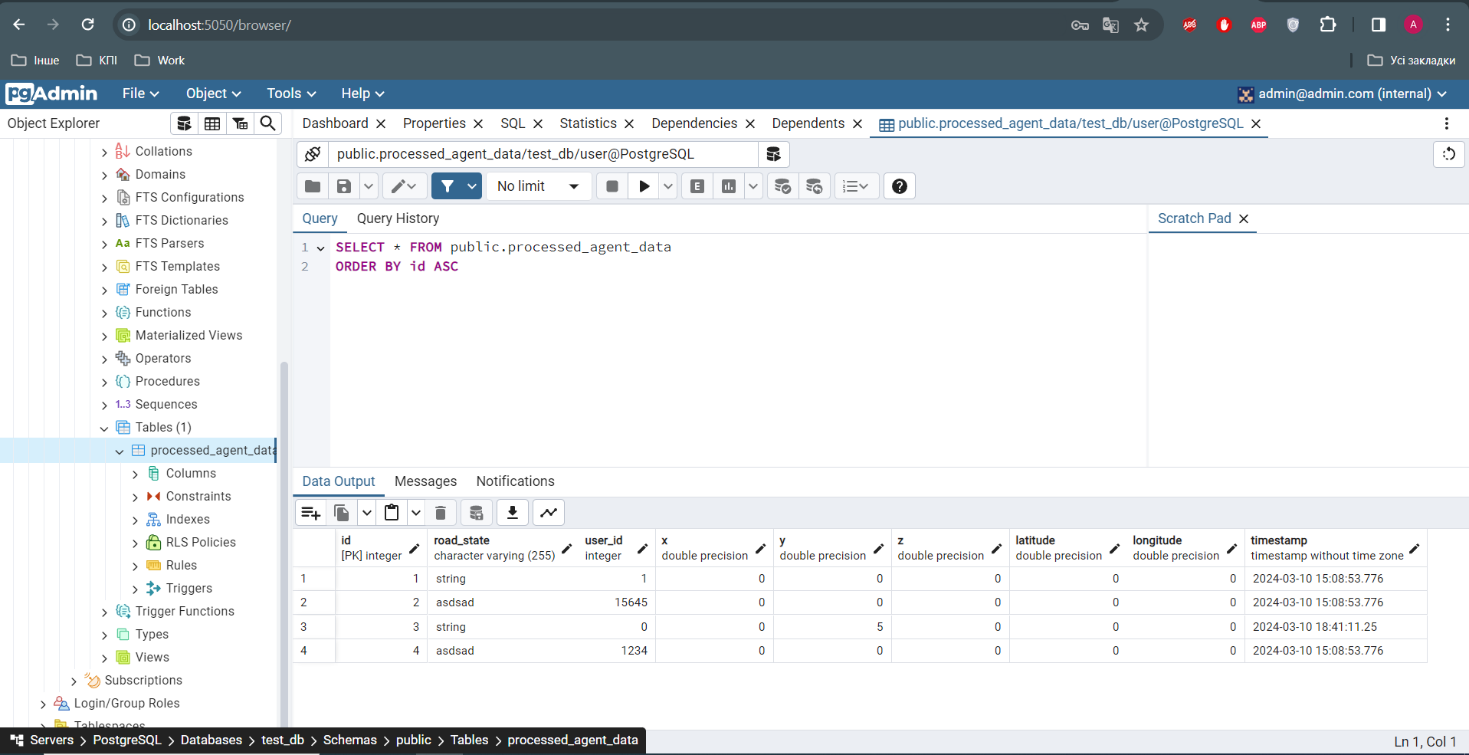
        except Exception as e:

            logging.info(f"Error occured {e}")

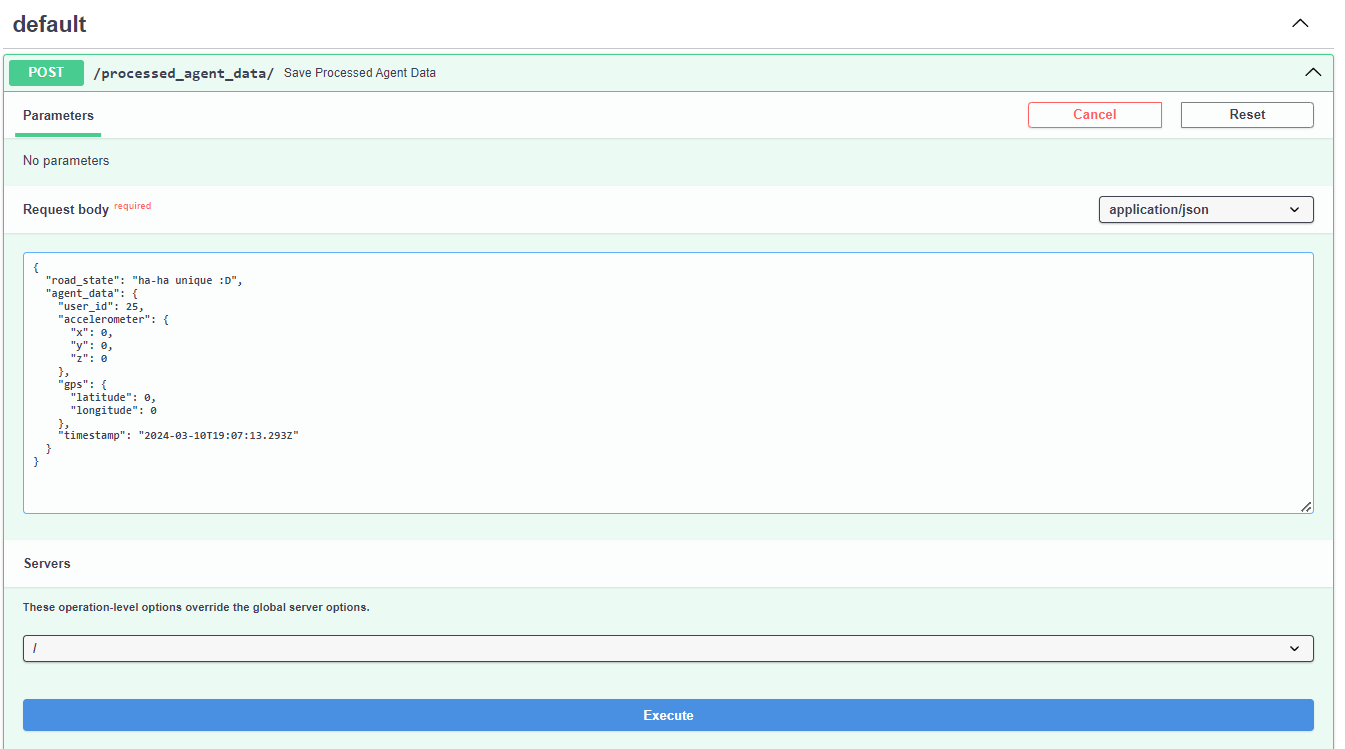
            return False

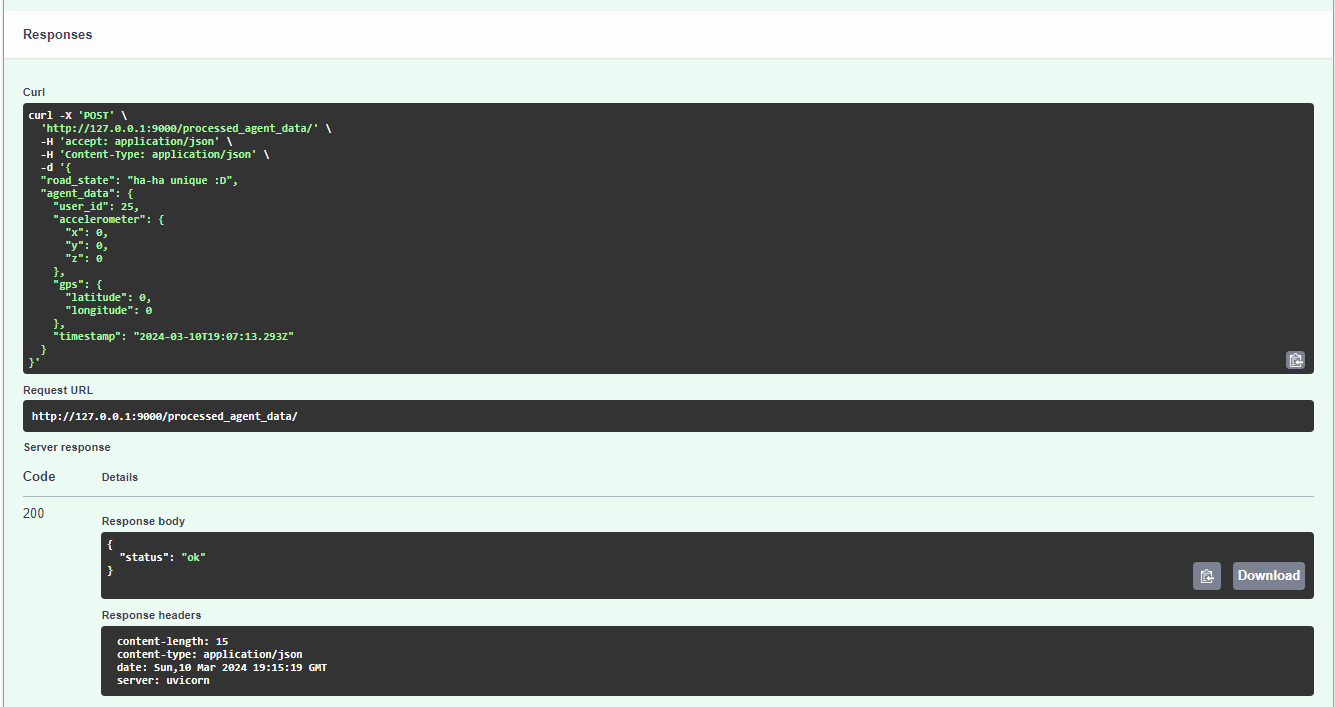
1. Демонстрація результатів роботи в Swagger/MQTT та PgAdmin

pgAdmin початковий стан:

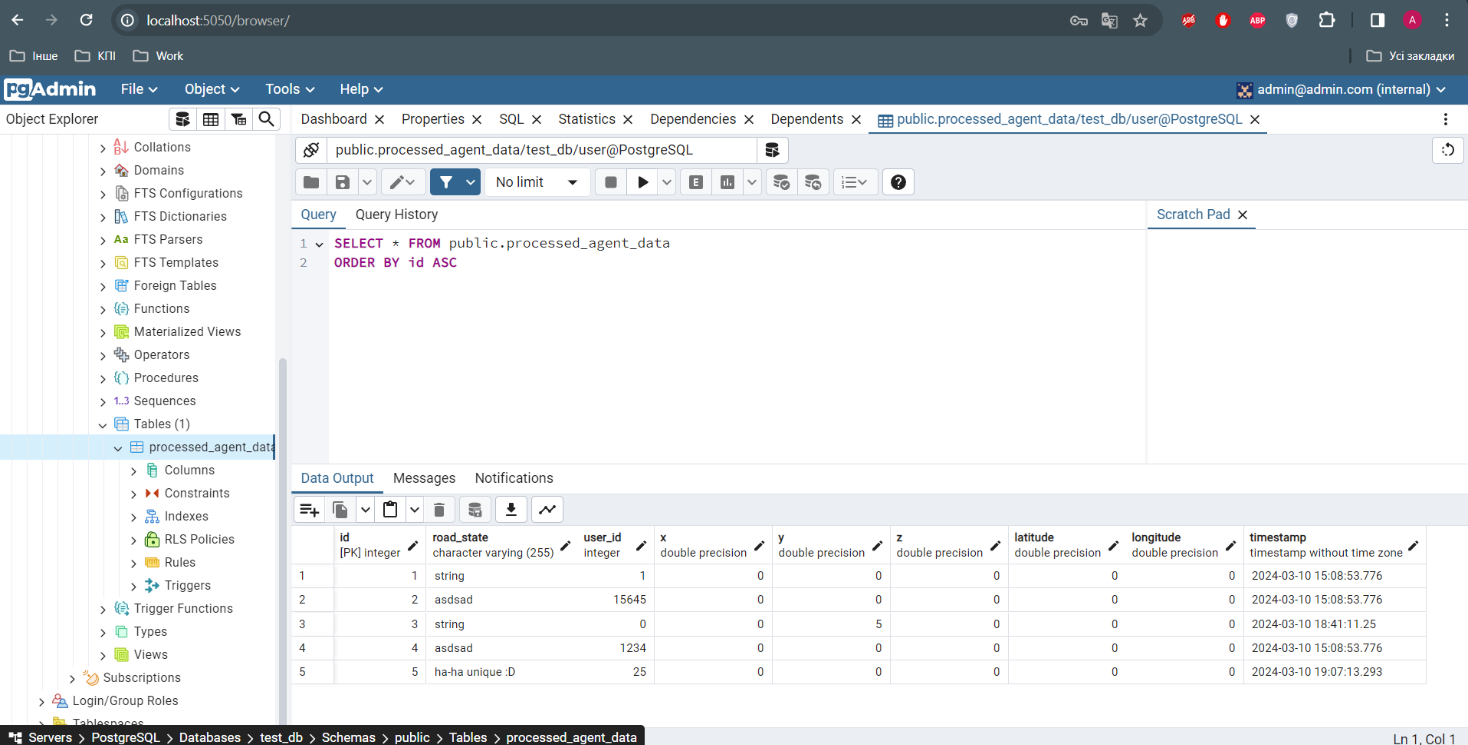


Swagger(hub):

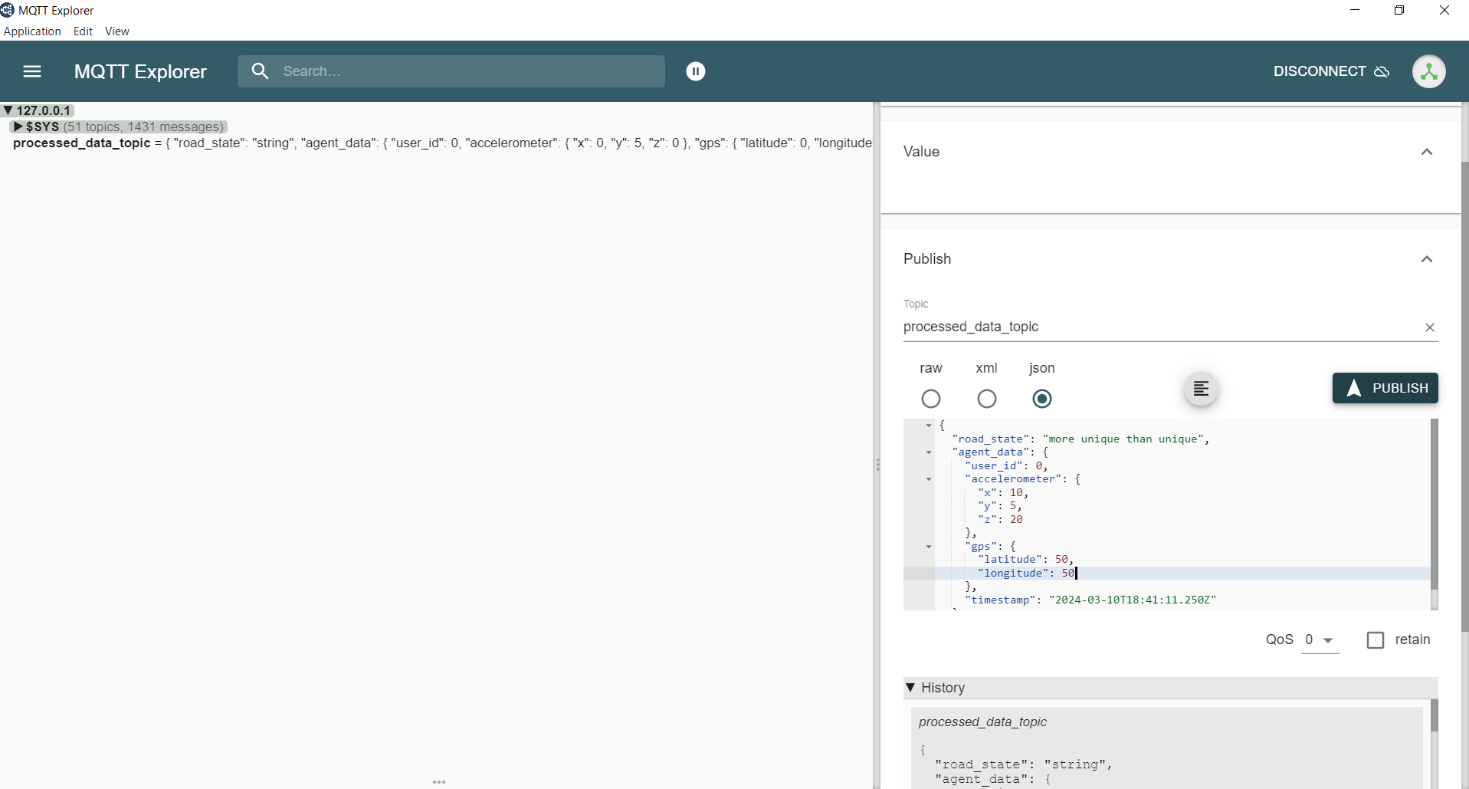


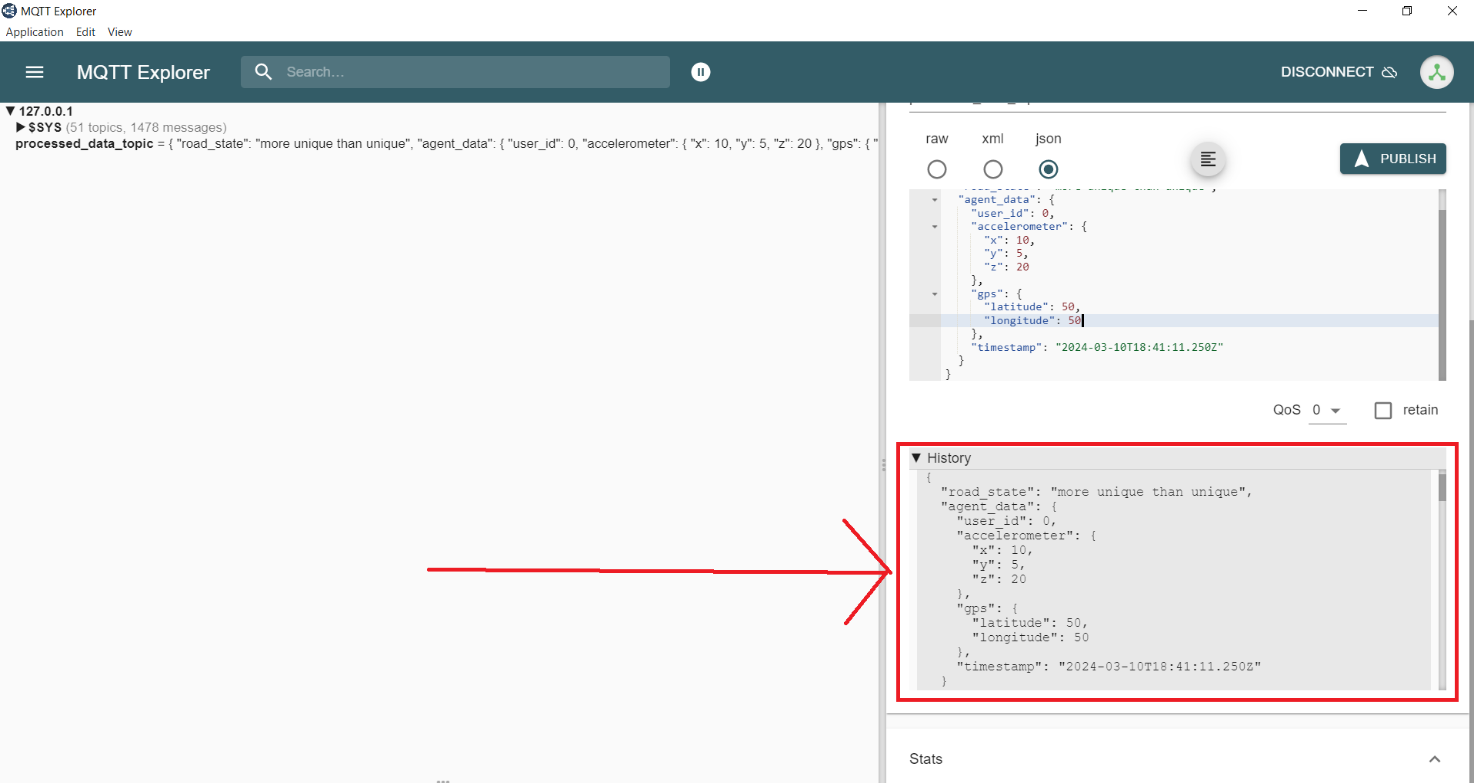


pgAdmin:

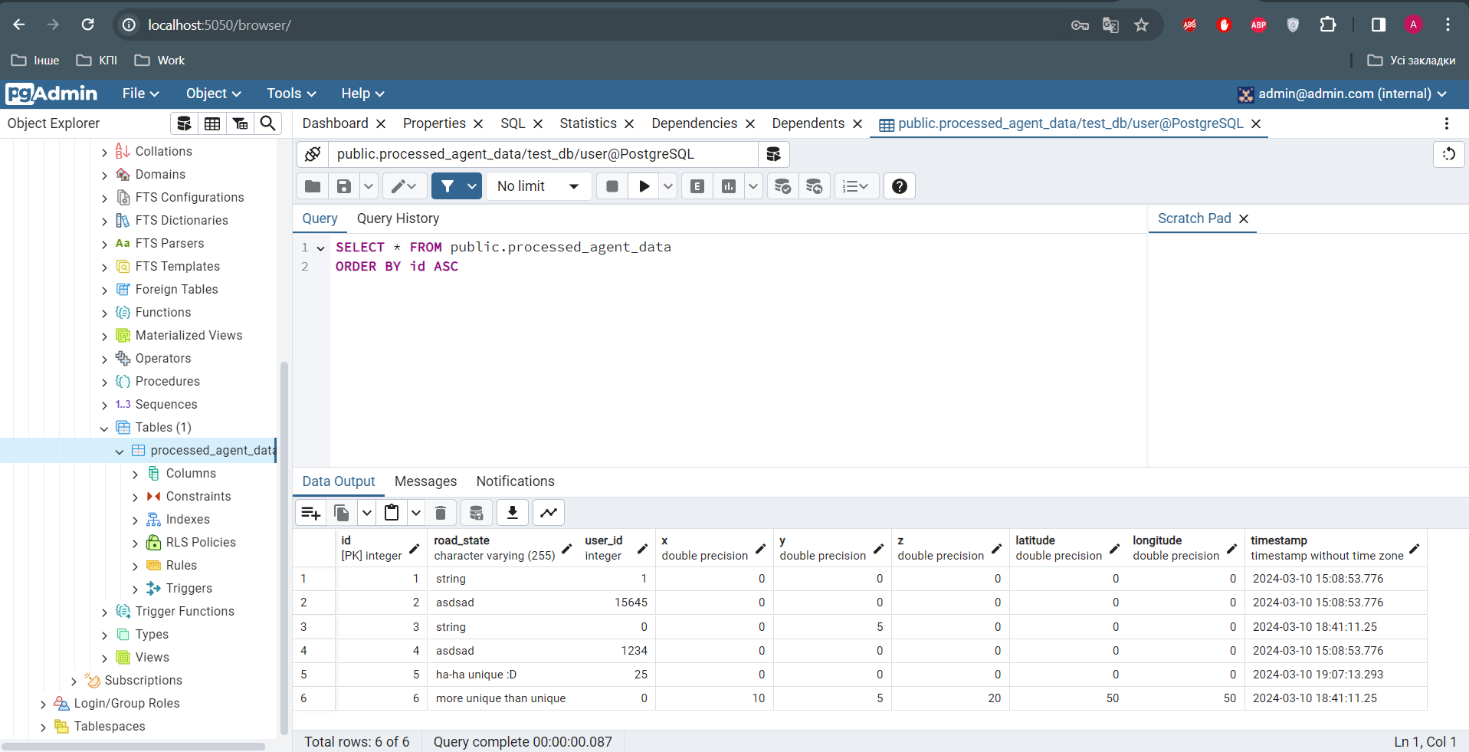


MQTT:



Бачимо оновлену історію  


pgAdmin:



**Висновок:** під час виконання даної лабораторної роботи ми ознайомились з поняттям адаптерів у Python та їх застосуванням. З допомогою мови Python реалізували класс StoreApiAdapter, який наслідує StorageGateway(наданий в скелетоні проекту), та імплементує його метод `save\_data` для збереження обробленої інформації.