**Домашнє завдання до теми «Apache Kafka»**

**Завдання 1: Створення топіків**

Для початку було реалізовано створення трьох топіків у Kafka за допомогою бібліотеки **confluent\_kafka**. У скрипті *task1\_create\_topics.py* створено топіки для зберігання даних сенсорів та сповіщень про перевищення порогових значень. Назви топіків включають унікальний ідентифікатор, щоб уникнути конфліктів. Програма спочатку перевіряє наявність існуючих топіків у Kafka та створює нові лише за потреби. Після успішного виконання на екрані виводиться перелік створених топіків, а також повний список усіх доступних топіків у Kafka, підтверджуючи коректну роботу скрипта.

A screenshot of a computer program

Description automatically generated

**Завдання 2: Генерація та відправка даних сенсорів**

Наступним етапом було реалізовано симуляцію роботи сенсорів у скрипті *task2\_generate\_data.py*. Сенсори генерують випадкові значення температури у діапазоні від 25°C до 45°C та вологості від 15% до 85%. Кожен запуск скрипта імітує окремий сенсор із унікальним ідентифікатором sensor\_id, який залишається незмінним для одного запуску. Дані відправляються до топіку **building\_sensors** у Kafka. Для перевірки коректної роботи програми було запущено два екземпляри скрипта паралельно, що дозволило згенерувати та надіслати дані з кількох сенсорів одночасно. У терміналі виводяться як згенеровані значення температури й вологості, так і підтвердження про успішне відправлення даних до Kafka.

A screenshot of a computer program

Description automatically generated

**Завдання 3: Обробка та фільтрація даних**

На цьому етапі у скрипті *task3\_process\_data.py* було реалізовано обробку отриманих даних. Програма підписується на топік **building\_sensors** і отримує дані у реальному часі. Далі дані перевіряються на відповідність заданим умовам: якщо температура перевищує 40°C, формується сповіщення, яке відправляється у топік **temperature\_alerts**. Якщо вологість виходить за межі 20%-80%, сповіщення відправляється у топік humidity\_alerts. У терміналі відображається як “сирі” дані, отримані з топіка **building\_sensors**, так і оброблені дані, що відповідають умовам. Також показується підтвердження про успішне відправлення відфільтрованих сповіщень у відповідні топіки.

A screen shot of a computer program

Description automatically generated

**Завдання 4: Читання та вивід сповіщень**

Останнім кроком було читання сповіщень з топіків **temperature\_alerts** та **humidity\_alerts**, яке реалізовано у скрипті *task4\_display\_alerts.py*. Програма підписується на обидва топіки й отримує сповіщення у реальному часі. У терміналі відображаються отримані сповіщення із зазначенням типу перевищення (температура або вологість) та ідентифікатора сенсора, який надіслав дані. Це дозволяє відслідковувати критичні значення у показниках датчиків і підтверджує, що система обробки даних працює коректно.

A screenshot of a computer

Description automatically generated

**Висновок**

В рамках завдання було реалізовано повний цикл роботи з Apache Kafka: створення топіків, генерацію та відправлення даних сенсорів, обробку отриманих даних і їх фільтрацію, а також виведення сповіщень про перевищення порогових значень. Програма успішно демонструє обробку потокових даних у реальному часі за допомогою Kafka.