МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ

РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

Кафедра «Учетные системы и технологии бизнес-менеджмента»

**ЛАБОРАТОРНАЯ РАБОТА**

###### по дисциплине «Средства и технологии анализа и разработки информационных систем»

Выполнил: Принял:

студент группы ГИ-31 ст. преподаватель

Алексеенко А.С. Козлов В,Г.

# Гомель 2025

# ЛАБОРАТОРНАЯ РАБОТА №10

## РАЗРАБОТКА СИСТЕМЫ ОБРАБОТКИ СООБЩЕНИЙ С ИСПОЛЬЗОВАНИЕМ APACHE KAFKA

**Цель работы:** Разработка системы асинхронной обработки сообщений с использованием Apache Kafka, включая создание продюсеров, консьюмеров и настройку их заимодействия.

1) При помощи конфигурационного файла docker-compose.yml были установлены ZooKeeper и Kafka:

version: '3'

services:

  zookeeper:

    image: confluentinc/cp-zookeeper:7.4.0

    environment:

      ZOOKEEPER\_CLIENT\_PORT: 2181

      ZOOKEEPER\_TICK\_TIME: 2000

    ports:

      - "2181:2181"

  kafka:

    image: confluentinc/cp-kafka:7.4.0

    depends\_on:

      - zookeeper

    ports:

      - "9092:9092"

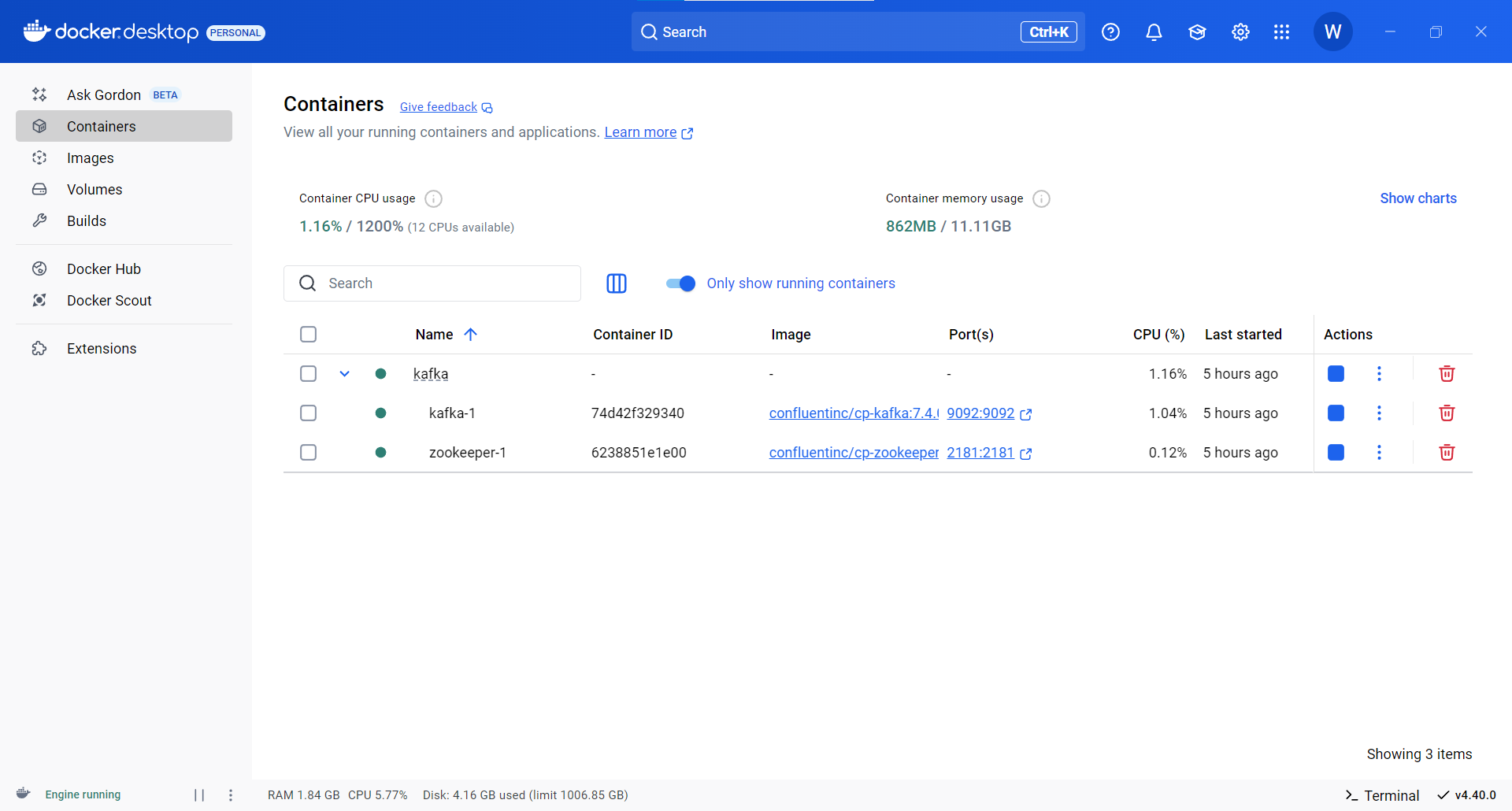
    environment:

      KAFKA\_BROKER\_ID: 1

      KAFKA\_ZOOKEEPER\_CONNECT: zookeeper:2181

      KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT://localhost:9092

      KAFKA\_OFFSETS\_TOPIC\_REPLICATION\_FACTOR: 1



2) Создан топик user\_actions с 3 партициями:

kafka-topics.bat --create --topic user\_actions --partitions 3 --replication-factor 1 --bootstrap-server localhost:9092

3) Созданы файлы producer.py и consumer.py (в папке)

4) Создана база данных с таблицей:

CREATE DATABASE kafka\_events;

CREATE TABLE user\_actions (

id SERIAL PRIMARY KEY,

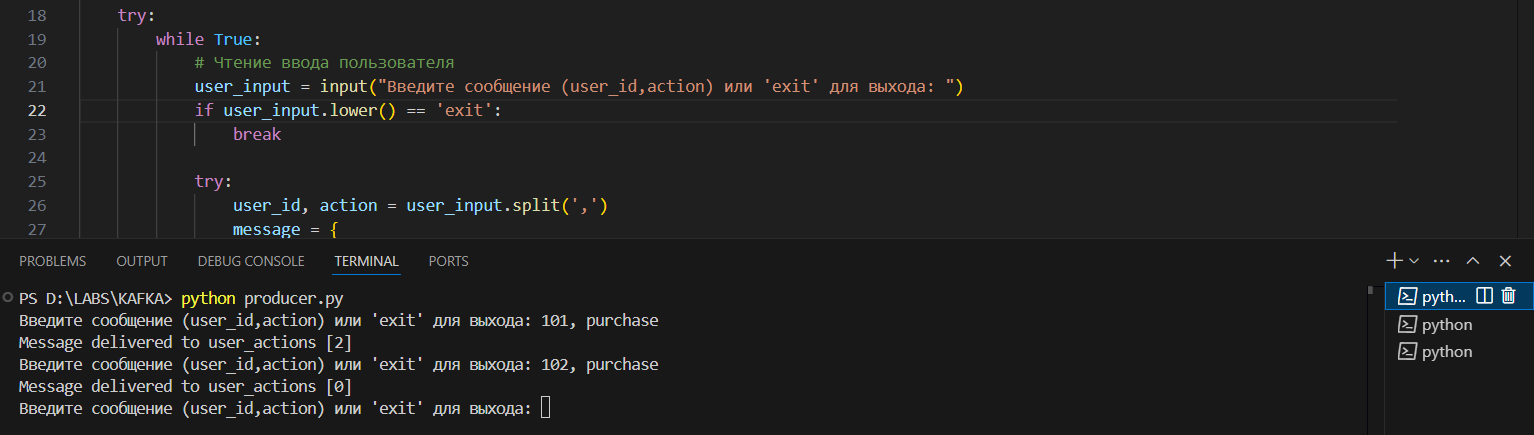
user\_id INTEGER NOT NULL,

action VARCHAR(50) NOT NULL,

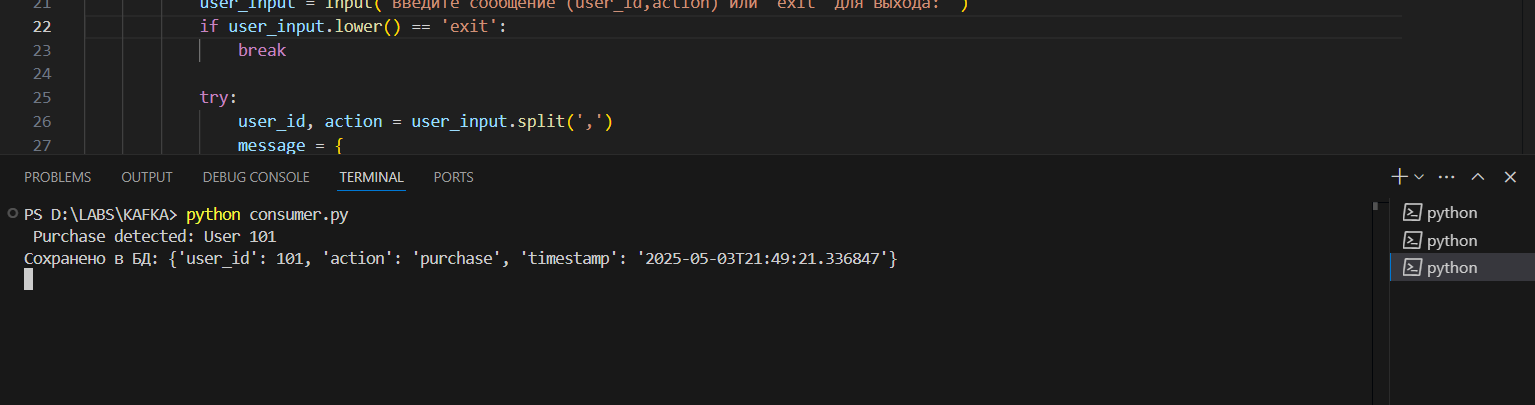
timestamp TIMESTAMP NOT NULL,

processed\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP);

5) Запуск producer.py и двух consumer.py. Отправлены 2 сообщения: юзеру с id 101 и юзеру с id 102



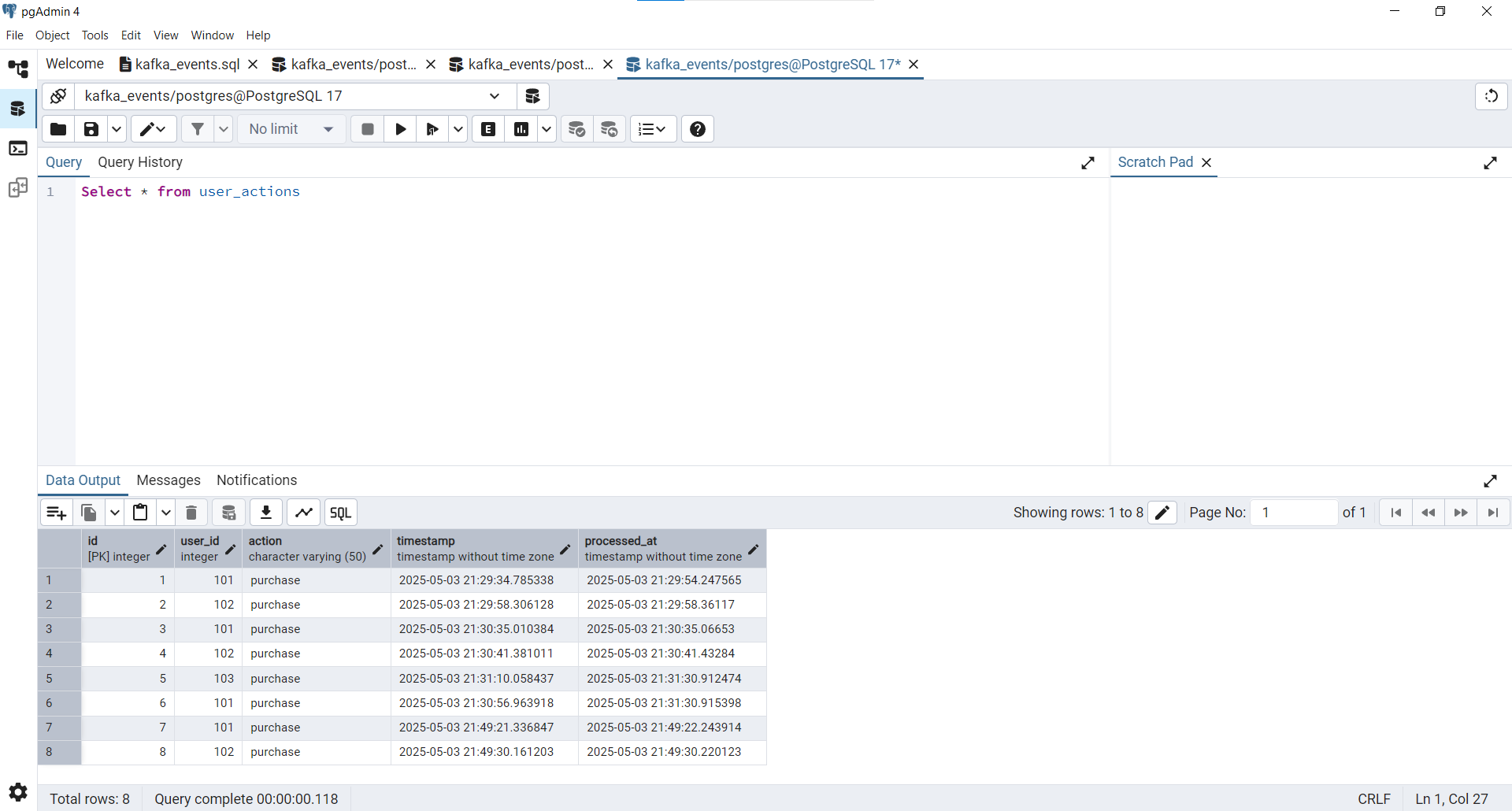
Юзер с id 101 получил сообщение:



Юзер с id 102 получил сообщение:



6) Сохранение данных в БД:



## Контрольные вопросы:

1. ZooKeeper хранит метаданные Kafka (брокеры, топики, партиции) и выбирает контроллер, но современные версии Kafka (KRaft) могут работать без ZooKeeper.

2. Сообщения в партиции строго упорядочены. Продюсер должен отправлять их последовательно, а консьюмер — обрабатывать одной нитью на партицию.

3. Earliest — чтение с самого начала партиции (все данные), Latest — только новые сообщения после подключения консьюмера.