Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Распределенные системы

**Лабораторная работа 4.1**

**Сравнение подходов хранения больших данных**

Выполнил(а): Ванярина Ю. А., группа: АДЭУ-211

Преподаватель: Босенко Т.М.

Москва

2024

**Цель работы**: сравнить производительность и эффективность различных подходов к хранению и обработке больших данных на примере реляционной базы данных PostgreSQL и документо- ориентированной базы данных MongoDB.

## **Оборудование и программное обеспечение**

* Компьютер с операционной системой Ubuntu.
* PostgreSQL.
* MongoDB.
* Python 3.x.
* Библиотеки: psycopg2, pymongo, pandas, matplotlib.

## **Теоретическая часть**

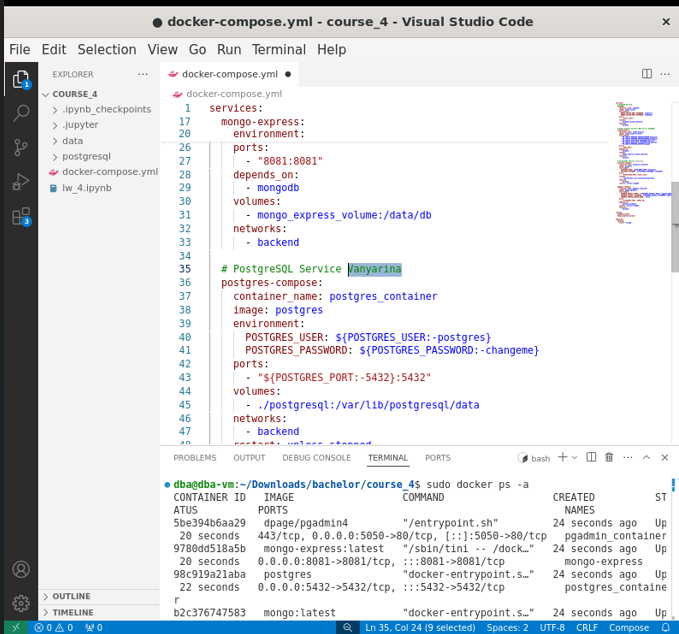
В современном мире объемы данных растут экспоненциально, что приводит к необходимости использования эффективных методов их хранения и обработки. Существует два основных подхода к хранению больших данных:

1. Реляционные базы данных (например, PostgreSQL)
2. NoSQL базы данных (например, MongoDB)

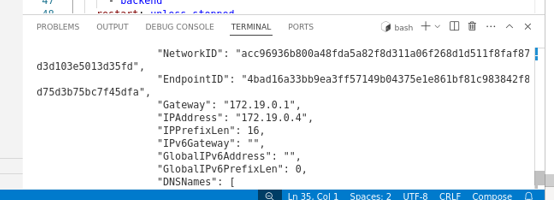
Каждый из этих подходов имеет свои преимущества и недостатки, которые мы рассмотрим в ходе выполнения лабораторной работы.

Ход работы.

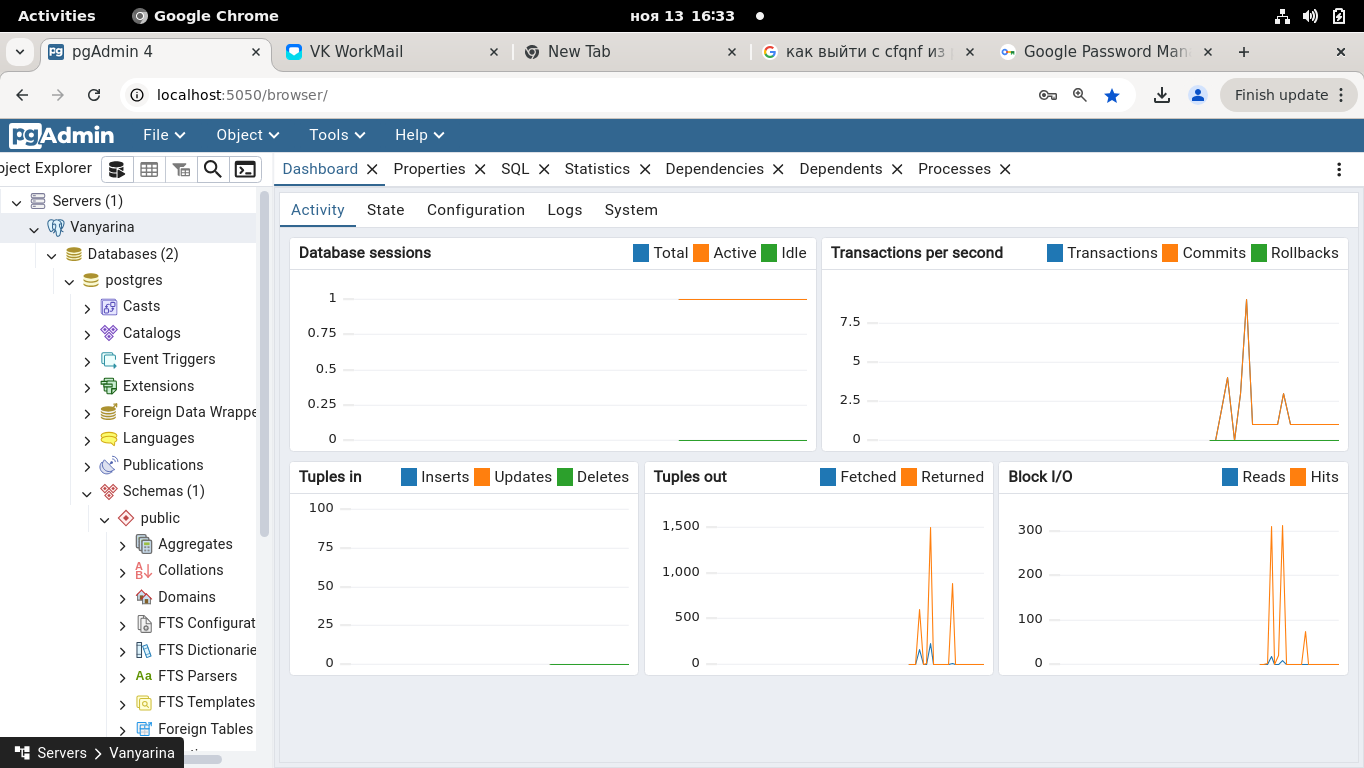
1. Подключились к виртуальной машине через удаленный доступ с помощью xrdp, предварительно узнав ip adress виртуальной машины и добавив порт 3389 к подключению
2. Следующим шагом были запущены контейнеры с помощью команды sudo docker compose up -d

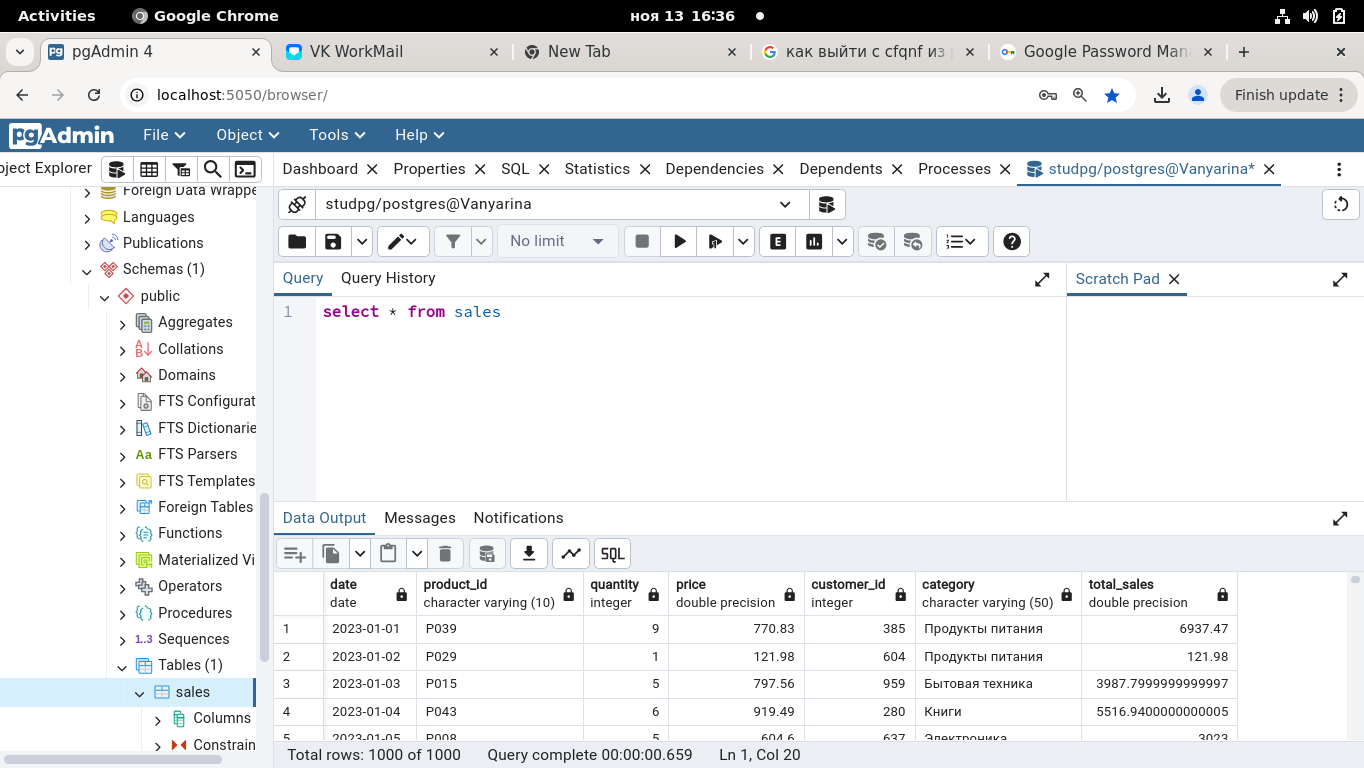


1. для подключения узнаем ip adress нужного нам контейнера с помощью команды sudo docker inspect [id\_container]



1. было выполнено подключение к pgadmin и добавлен сервер Vanyarina

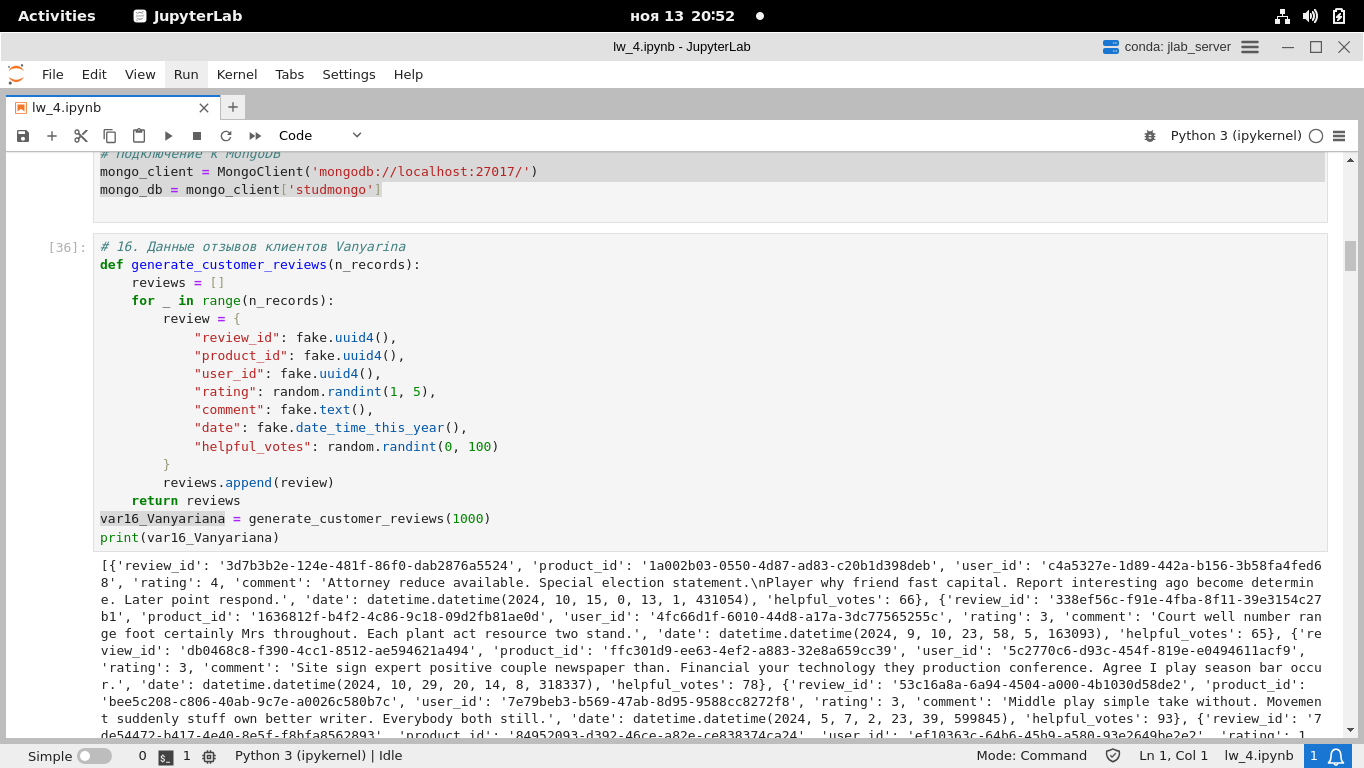


1. далее была загружена таблица sales 

Индивидуальное задание вариант 16

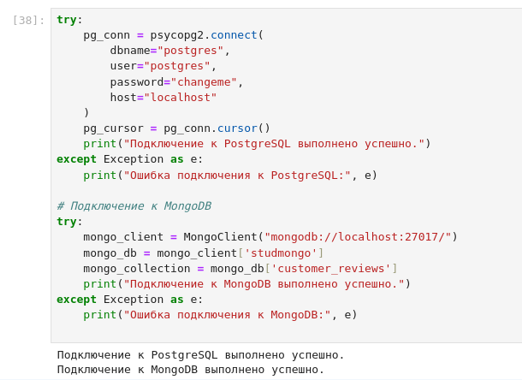
Вариант 16. Исследовать возможности и производительность при работе с JSON-данными на примере системы сбора и анализа отзывов клиентов.

1. В качестве тестовых данных были сгенерированы данные об отзывах клиентов

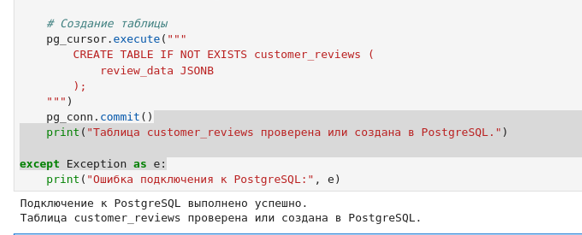


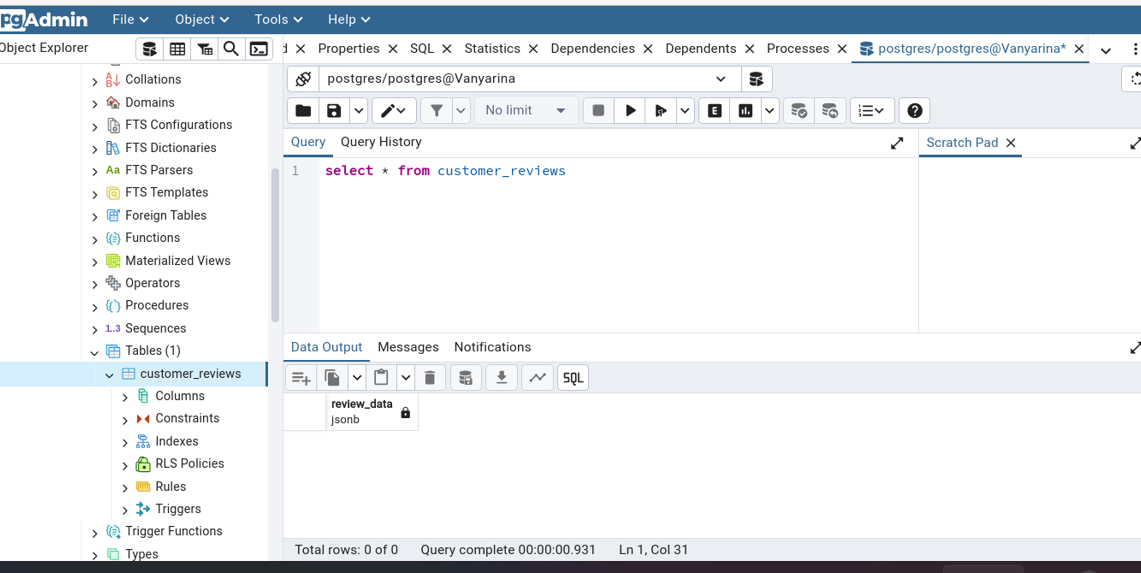
1. Далее с данными было выполнено несколько операций,где записывалось время выполнения операции в postgresql и mongodb

Подключение к postgresql и mongodb выполнено успешно

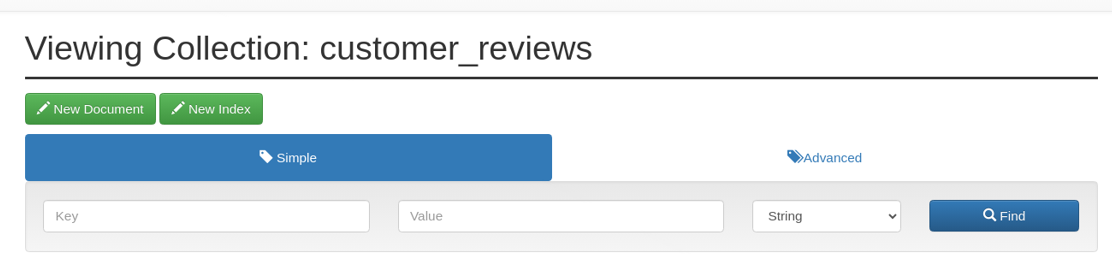


1. далее была создана таблица customer\_reviews (с указанием типа данных json)





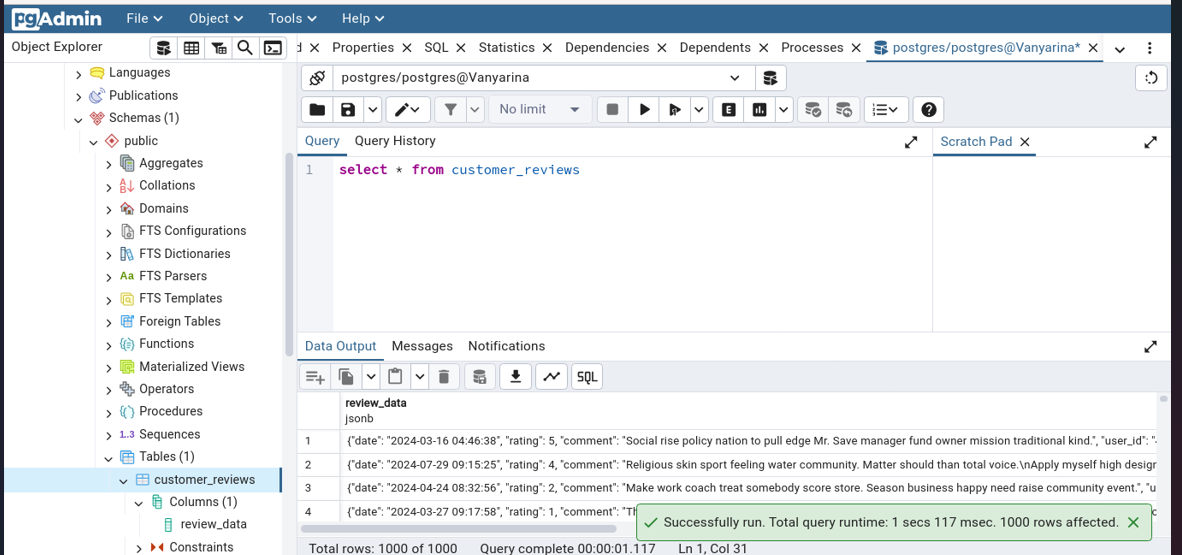
также в монго создали коллекцию customer\_reviews



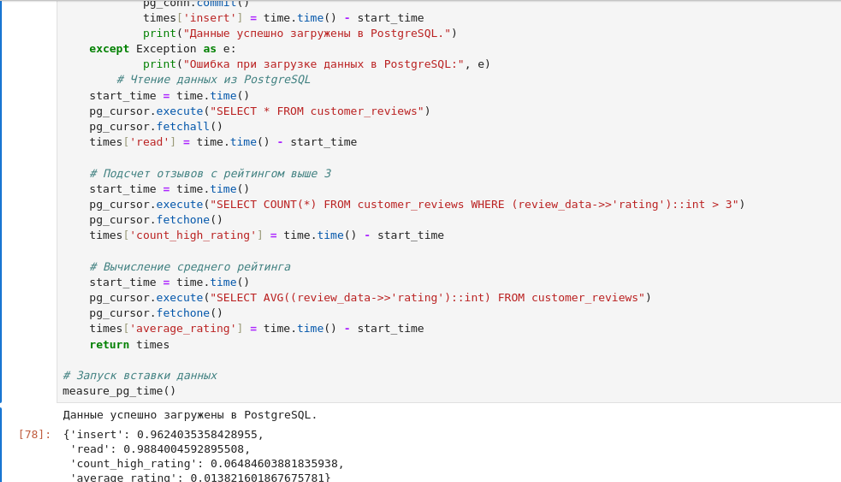
1. Далее были загружены данные в таблицу в postgresql

также использована функция для сохранения времени выполнения операции (вставка данных)





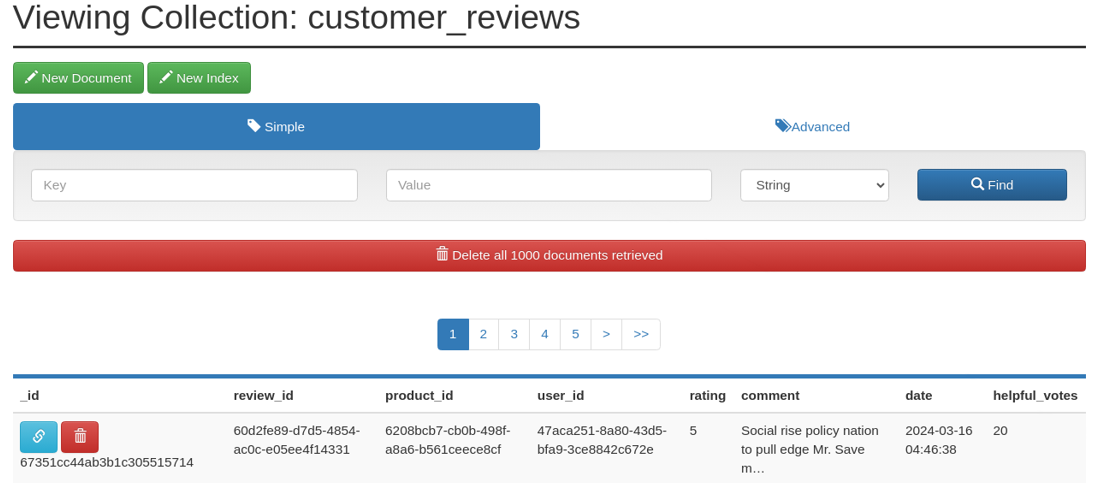
Также было записано время выполнения других операций,а именно: чтение данных,подсчёт рейтинга выше 3 и среднего рейтинга



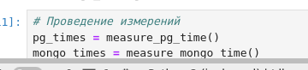
Проводим те же измерения в mongodb

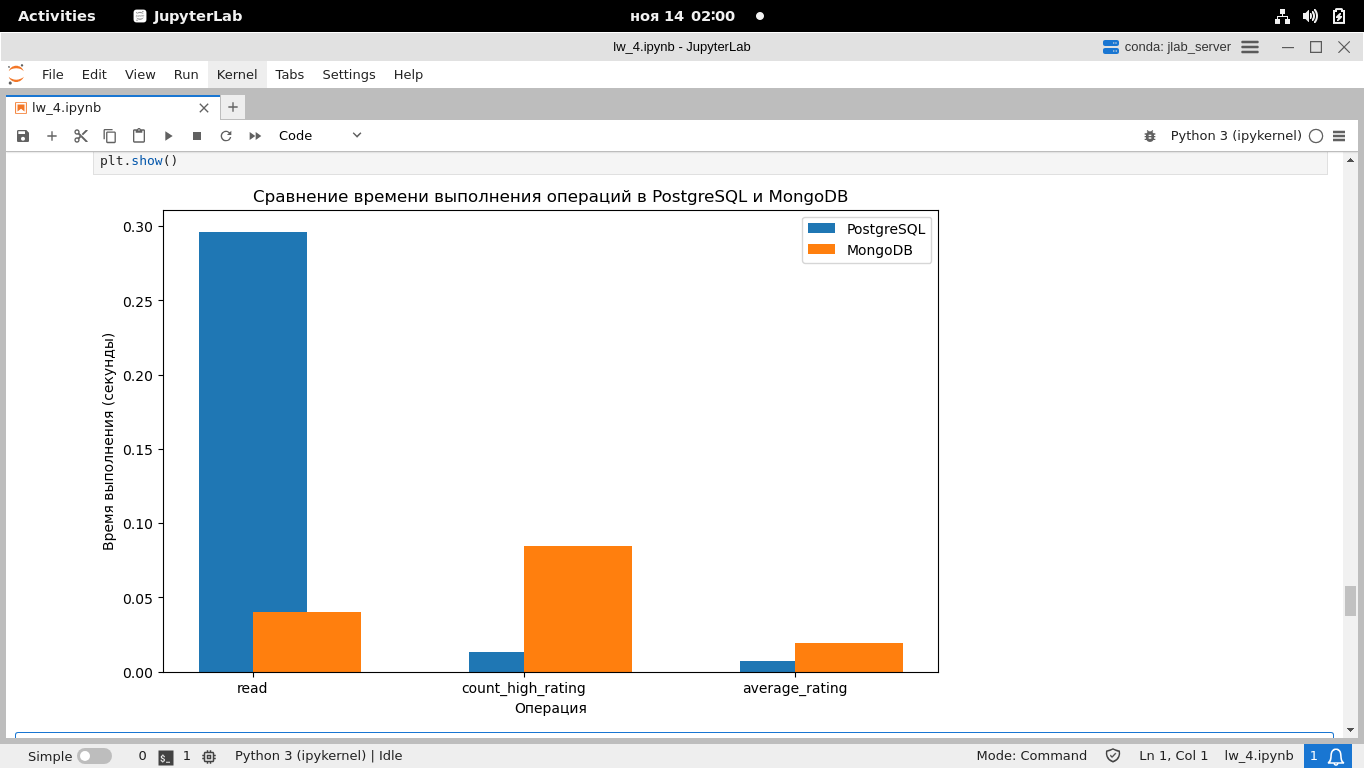






1. Проводим измерения с полученными временными данными и строим график для наглядности





Выводы по работе:

1. Удобнее,проще и привычнее было работать с postresql, на графике видно,что чтение данных было дольше в postresql
2. Подсчёт данных с рейтингом выше трёх postresql подсчитал за 00.64 с, что быстрее mongodb с результатом 0.1
3. Средний рейтинг postresql так же подсчитал быстрее за 0.013 с,а mongodb за 0.09 с