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

Инструменты хранения и анализа больших данных

Смоляков Руслан Игоревич БД-241м

**Практическая работа 2.1. Изучение методов хранения данных на основе NoSQL**

**Вариант 23**

Направление подготовки/специальность

38.04.05 - Бизнес-информатика

Бизнес-аналитика и большие данные

(очная форма обучения)

Руководитель дисциплины:

Босенко Т.М., доцент департамента

информатики, управления и технологий,

кандидат технических наук

Москва

2025

**Введение**

***Цель занятия:***

***Цель данного занятия — освоение работы с базой данных GraphDB в виртуальной машине с использованием Docker. Студенты научатся загружать RDF-данные в базу, выполнять запросы с использованием SPARQL и анализировать результаты, используя функциональность GraphDB.***

***Задачи:***

1. ***Настроить и запустить контейнер с GraphDB с помощью Docker.***
2. ***Загрузить RDF-данные о фильмах в базу данных.***
3. ***Ознакомиться с основами SPARQL-запросов.***
4. ***Выполнить различные SPARQL-запросы для получения информации из базы данных.***
5. ***Проанализировать и интерпретировать результаты выполнения запросов.***

***Необходимое ПО:***

***Операционная система: Ubuntu 22.***

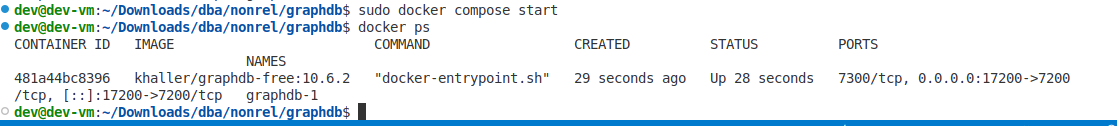
***СУБД: GraphDB.***

***Docker: для запуска контейнера с GraphDB.***

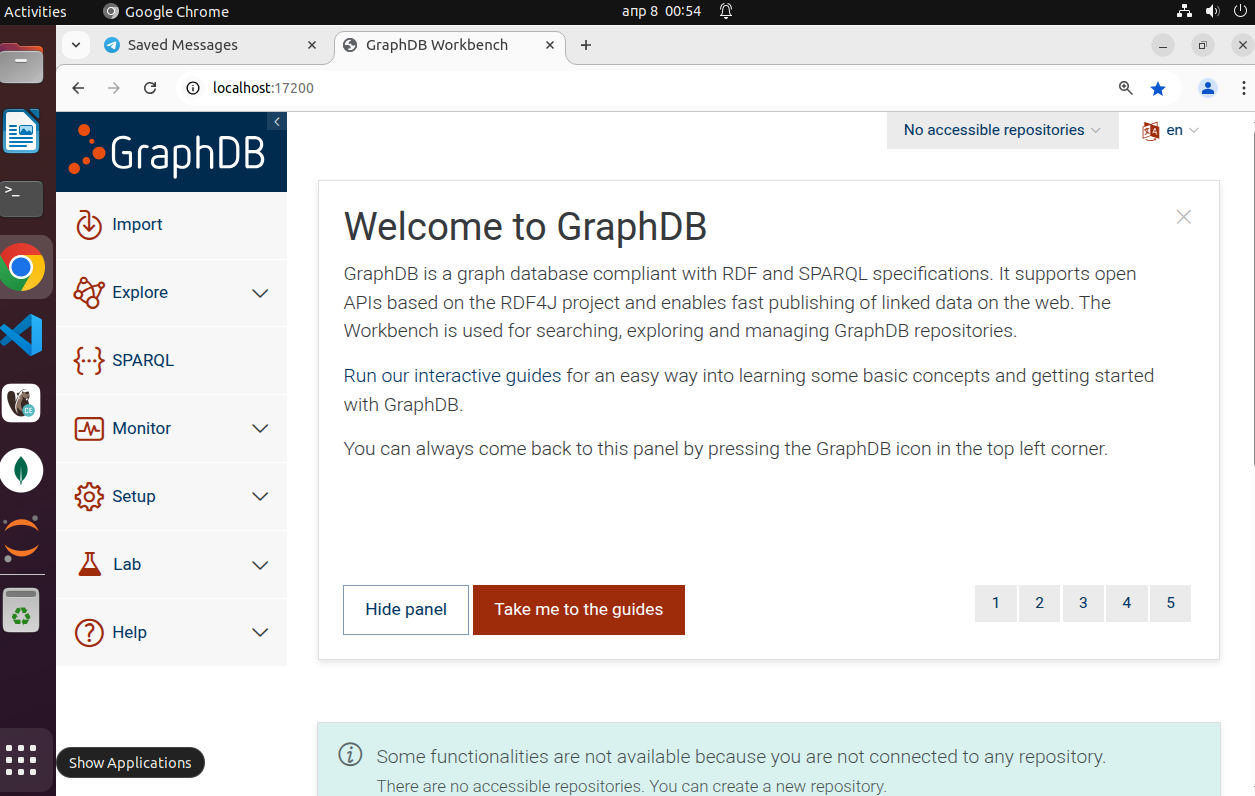
***Среда разработки: SPARQL редактор для выполнения запросов.***

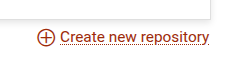
**Вариант 23**

1. Напишите запрос для поиска всех фильмов с участием "Morgan Freeman".
2. Найдите все фильмы, которые были сняты после 2010 года.
3. Выведите все фильмы, которые сняты в жанре фэнтези.
4. Найдите все фильмы, у которых рейтинг меньше 6,0 и количество комментариев меньше 100.
5. Напишите запрос для вывода всех фильмов, снятых в 2000-х годах.

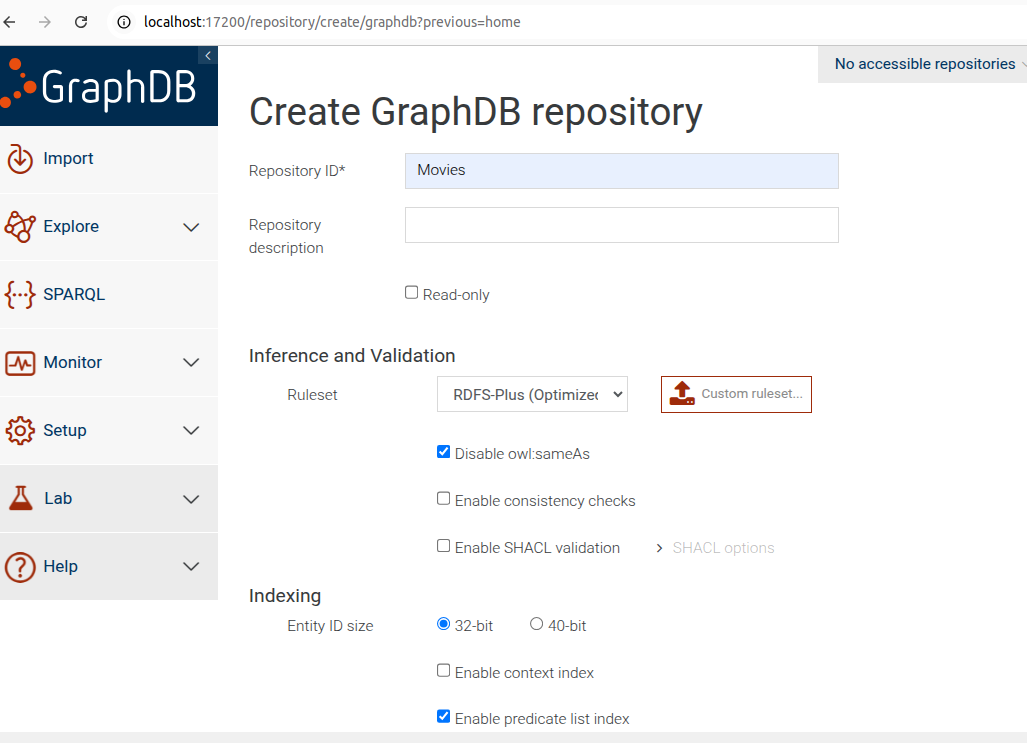
Запустим docker-compose:  


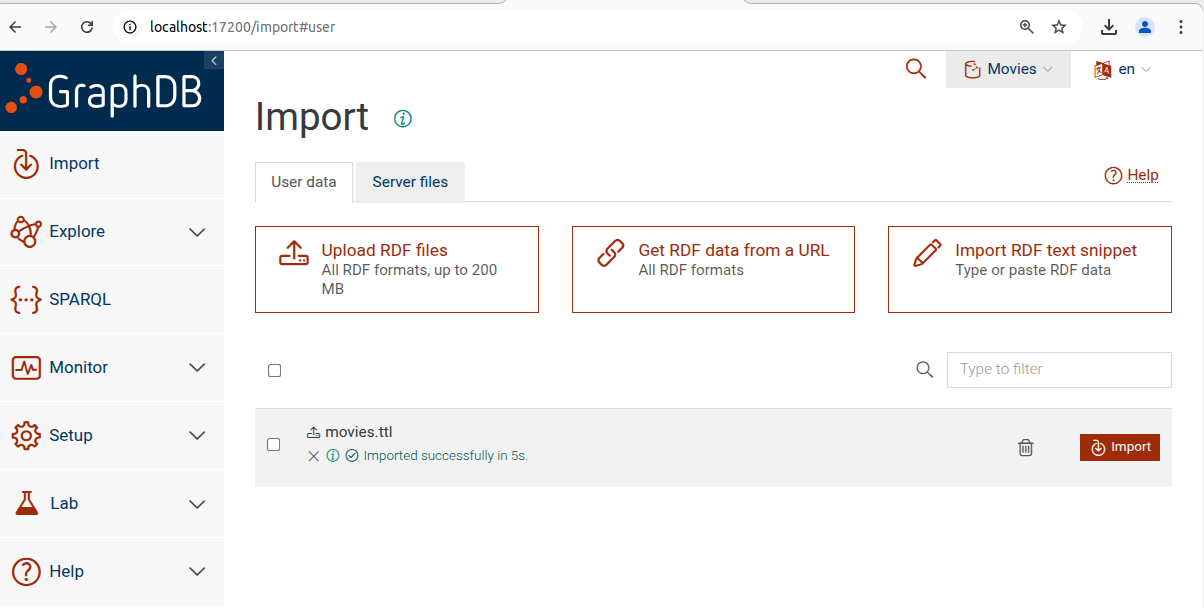
Проверим перейдя по адресу «http://localhost:17200».



Создаем новый репозиторий:  


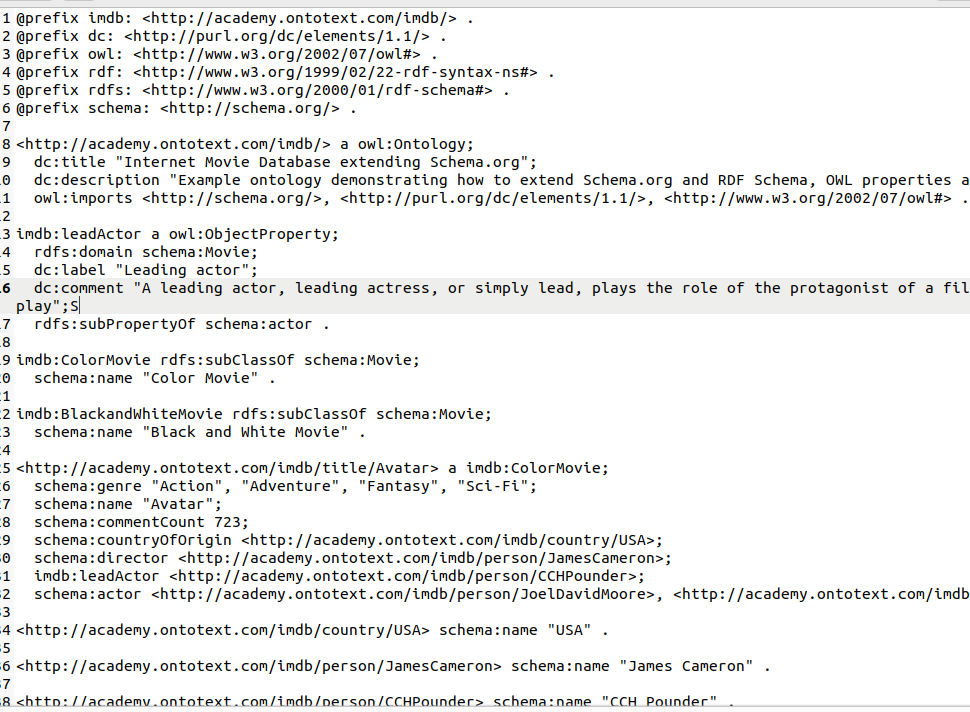
Задаем ID «Movies», остальные настройки не меняем:



Скачаем данные о фильмах (с GitHub, который представлен в задании - <https://github.com/BosenkoTM/nosql-workshop/blob/main/07-working-with-graphdb>). Далее импортируем данные:  


Перед написанием запросов изучим структуру данных

Переходим в SPARQL и начинаем писать запросы.

Фрагмент:  


Видим используемые префиксы, а также предикаты:

* **schema:name**: Название фильма.
* **schema:genre**: Жанр фильма.
* **schema:commentCount**: Количество комментариев.
* **schema:actor**: Актеры, участвующие в фильме.
* **schema:director**: Режиссер фильма.

1. Напишите запрос для поиска всех фильмов с участием "Morgan Freeman".



Добавил фильтр, чтобы объединить цветные и черно-белые фильмы.

Вывод (буду прикреплять то, что поместилось, но целые таблицы закину на GitHub:



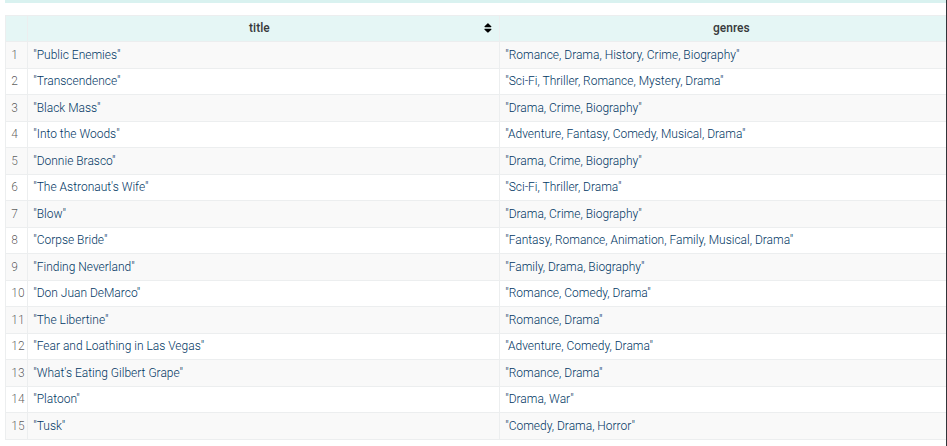
1. Найдите все фильмы, которые были сняты после 2010 года.

В данных нет года съемки фильма и рейтинга, поэтому запросы 2, 4 и 5 не получится выполнить, сам придумал на какие заменить:

Все цветные фильмы с участием Джонни Деппа в жанре драма:



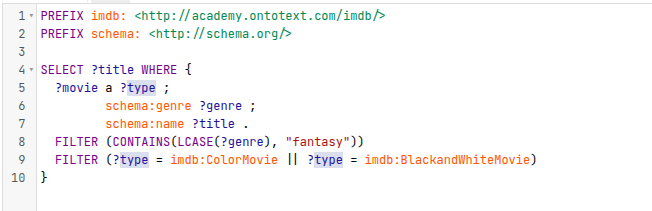
Вывод:

  
Использовал GROUP\_CONCAT, чтобы выводились все жанры, а не только «Драмма»

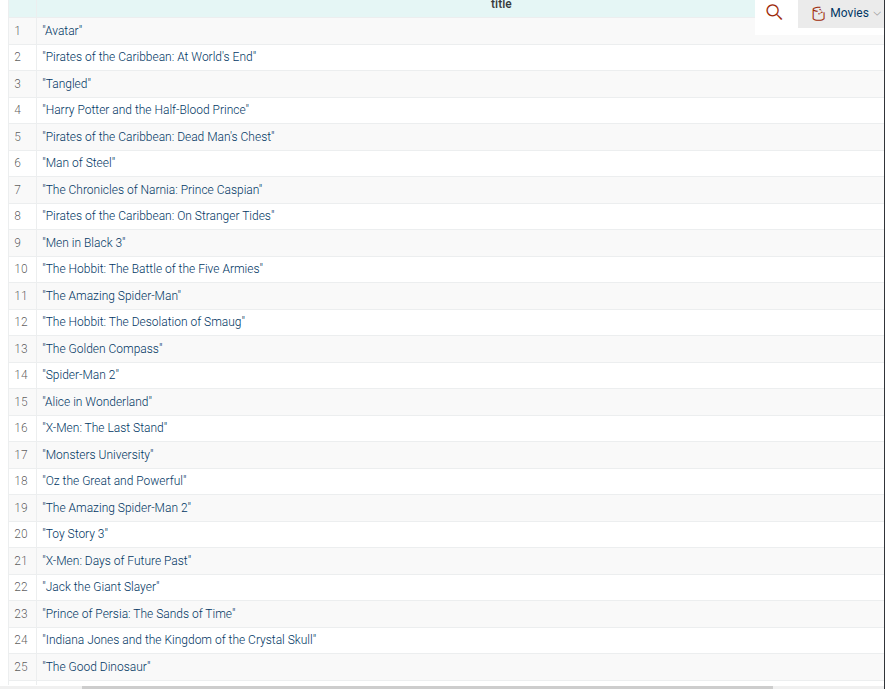
Мы ищем фильмы (**imdb:ColorMovie**), где актером является Джонни Депп.

Проверяем, есть ли среди жанров фильма слово "drama" (используем функцию **CONTAINS** для поиска подстроки).

1. Выведите все фильмы, которые сняты в жанре фэнтези.



Вывод:



1. Найдите все фильмы, у которых рейтинг меньше 6,0 и количество комментариев меньше 100.

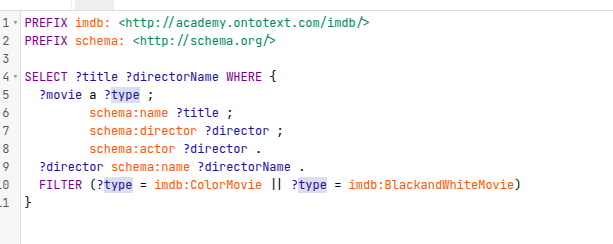
Заменил на: «Фильмы, снятые в США с количеством комментариев больше 500»

****

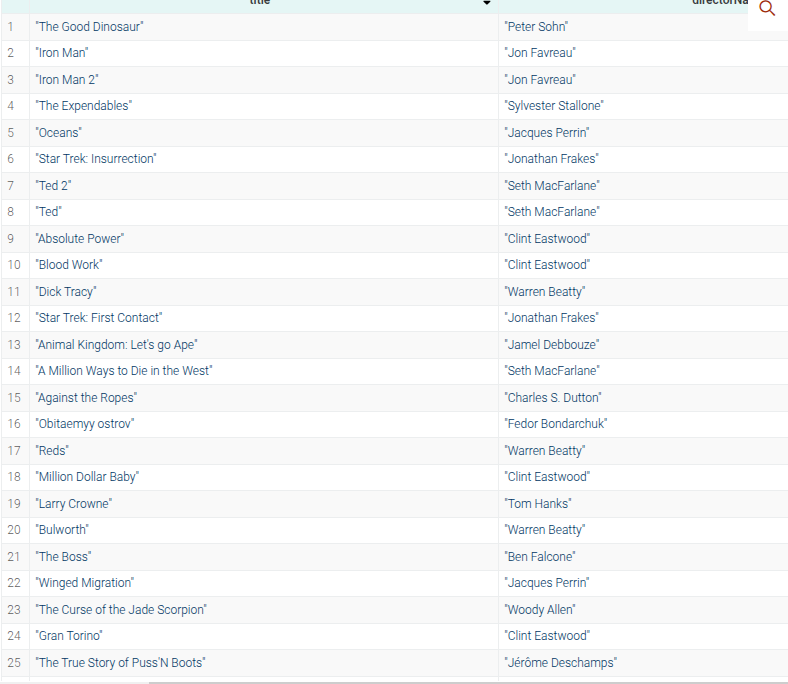
****

1. Напишите запрос для вывода всех фильмов, снятых в 2000-х годах.

Заменил на: Фильмы, где режиссер также является актером



Вывод:



Ссылка на гитхаб с результатами (папка results) и листингом каждого запроса (папка queries):

https://github.com/Ruslanishka/BDSAD/tree/main/2.1