

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.

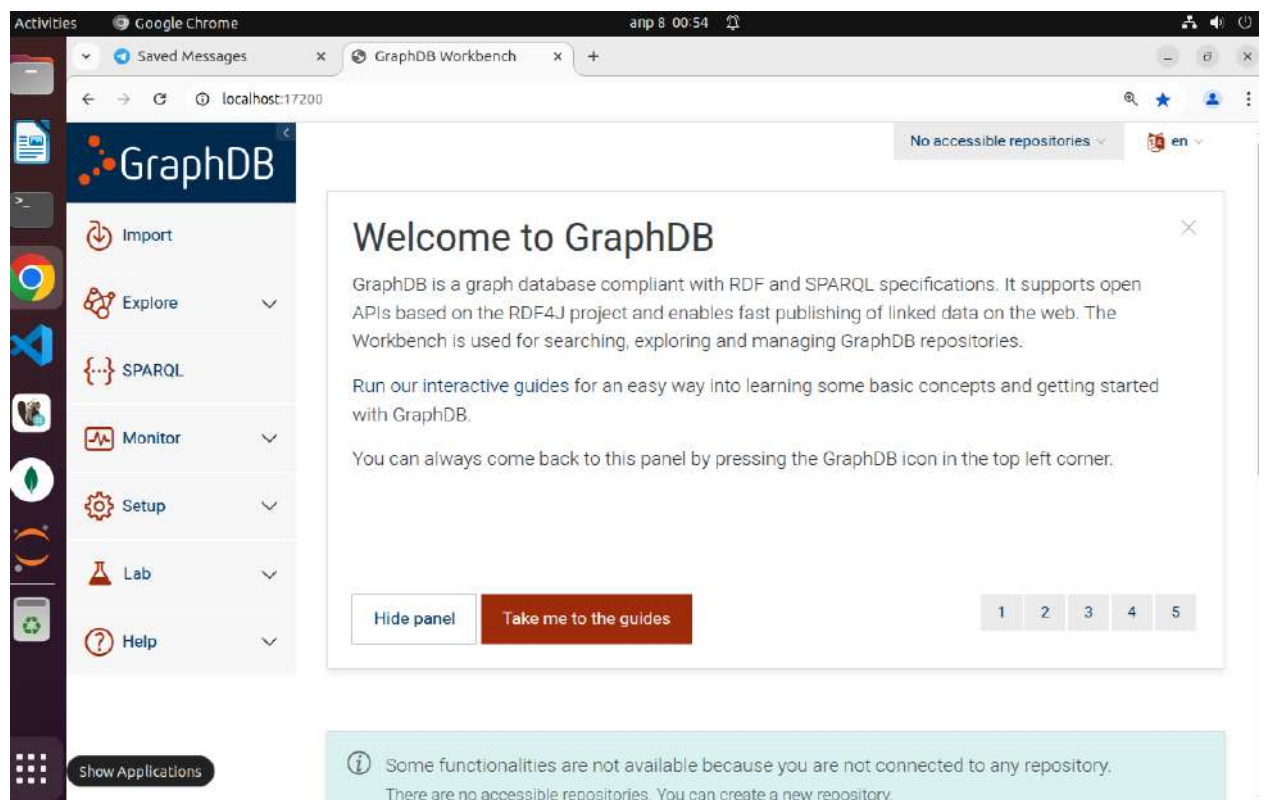
Запустим docker-compose:

```
dev@dev-vm:~/Downloads/dba/nonrel/graphdb$ sudo docker compose start
dev@dev-vm:~/Downloads/dba/nonrel/graphdb$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
481a44bc8396	khaller/graphdb-free:10.6.2	"docker-entrypoint.sh"	29 seconds ago	Up 28 seconds	7300/tcp, 0.0.0.0:17200->7200
/tcp, [::]:17200->7200/tcp	graphdb-1				

```
dev@dev-vm:~/Downloads/dba/nonrel/graphdb$
```

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



Создаем новый репозиторий:

 Create new repository

Задаем ID «Movies», остальные настройки не меняем:

localhost:17200/repository/create/graphdb?previous=home

GraphDB

No accessible repositories

Create GraphDB repository

Repository ID*

Repository description

☐ Read-only

Inference and Validation

Ruleset RDFS-Plus (Optimizer) Custom ruleset...

☒ Disable owl:sameAs

☐ Enable consistency checks

☐ Enable SHACL validation > SHACL options

Indexing

Entity ID size ☒ 32-bit ☐ 40-bit

☐ Enable context index

☒ Enable predicate list index

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:

localhost:17200/import#user

GraphDB

Import

User data **Server files** Help

☐ **Upload RDF files**
All RDF formats, up to 200 MB

☐ **Get RDF data from a URL**
All RDF formats

☐ **Import RDF text snippet**
Type or paste RDF data

☐ **movies.ttl**
Imported successfully in 5s. Import

Перед написанием запросов изучим структуру данных

Переходим в SPARQL и начинаем писать запросы.

Фрагмент:

```
1 @prefix imdb: <http://academy.ontotext.com/imdb/> .
2 @prefix dc: <http://purl.org/dc/elements/1.1/> .
3 @prefix owl: <http://www.w3.org/2002/07/owl#> .
4 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
5 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
6 @prefix schema: <http://schema.org/> .
7
8 <http://academy.ontotext.com/imdb/> a owl:Ontology;
9   dc:title "Internet Movie Database extending Schema.org";
10  dc:description "Example ontology demonstrating how to extend Schema.org and RDF Schema, OWL properties a
11  owl:imports <http://schema.org/>, <http://purl.org/dc/elements/1.1/>, <http://www.w3.org/2002/07/owl#> .
12
13  imdb:leadActor a owl:ObjectProperty;
14    rdfs:domain schema:Movie;
15    dc:label "Leading actor";
16    dc:comment "A leading actor, leading actress, or simply lead, plays the role of the protagonist of a fil
17    play";
18    rdfs:subPropertyOf schema:actor .
19
20  imdb:ColorMovie rdfs:subClassOf schema:Movie;
21    schema:name "Color Movie" .
22
23  imdb:BlackandWhiteMovie rdfs:subClassOf schema:Movie;
24    schema:name "Black and White Movie" .
25
26  <http://academy.ontotext.com/imdb/title/Avatar> a imdb:ColorMovie;
27    schema:genre "Action", "Adventure", "Fantasy", "Sci-Fi";
28    schema:name "Avatar";
29    schema:commentCount 723;
30    schema:countryOfOrigin <http://academy.ontotext.com/imdb/country/USA>;
31    schema:director <http://academy.ontotext.com/imdb/person/JamesCameron>;
32    imdb:leadActor <http://academy.ontotext.com/imdb/person/CCHPounder>;
33    schema:actor <http://academy.ontotext.com/imdb/person/JoelDavidMoore>, <http://academy.ontotext.com/imdb
34
35  <http://academy.ontotext.com/imdb/country/USA> schema:name "USA" .
36
37  <http://academy.ontotext.com/imdb/person/JamesCameron> schema:name "James Cameron" .
38
39  <http://academy.ontotext.com/imdb/person/CCHPounder> schema:name "CCH Pounder" .
```

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".

```
1 PREFIX imdb: <http://academy.ontotext.com/imdb/>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT ?title WHERE {
5   ?movie a ?type ;
6           schema:actor <http://academy.ontotext.com/imdb/person/MorganFreeman> ;
7           schema:name ?title .
8   FILTER (?type = imdb:ColorMovie || ?type = imdb:BlackandWhiteMovie)
9 }
```

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

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

1	"The Dark Knight"
2	"Evan Almighty"
3	"Batman Begins"
4	"Oblivion"
5	"Transcendence"
6	"Ben-Hur"
7	"Now You See Me 2"
8	"Bruce Almighty"
9	"Deep Impact"
10	"Wanted"
11	"Now You See Me"
12	"Olympus Has Fallen"
13	"Hard Rain"
14	"The Sum of All Fears"
15	"Ted 2"
16	"Dreamcatcher"
17	"RED"
18	"The Lego Movie"
19	"Invictus"
20	"Chain Reaction"
21	"Robin Hood: Prince of Thieves"
22	"Outbreak"
23	"The Big Bounce"
24	"The Bucket List"
25	"Unleashed"

2. 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:

```
1 PREFIX imdb: <http://academy.ontotext.com/imdb/>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT ?title (GROUP_CONCAT(?genre; SEPARATOR=", ") AS ?genres) WHERE {
5   ?movie a imdb:ColorMovie ;
6           schema:actor <http://academy.ontotext.com/imdb/person/JohnnyDepp> ;
7           schema:name ?title ;
8           schema:genre ?genre .
9   FILTER EXISTS {
10     ?movie schema:genre ?dramaGenre .
11     FILTER (CONTAINS(LCASE(?dramaGenre), "drama"))
12   }
13 }
14 GROUP BY ?movie ?title
```

Вывод:

	title	genres
1	"Public Enemies"	"Romance, Drama, History, Crime, Biography"
2	"Transcendence"	"Sci-Fi, Thriller, Romance, Mystery, Drama"
3	"Black Mass"	"Drama, Crime, Biography"
4	"Into the Woods"	"Adventure, Fantasy, Comedy, Musical, Drama"
5	"Donnie Brasco"	"Drama, Crime, Biography"
6	"The Astronaut's Wife"	"Sci-Fi, Thriller, Drama"
7	"Blow"	"Drama, Crime, Biography"
8	"Corpse Bride"	"Fantasy, Romance, Animation, Family, Musical, Drama"
9	"Finding Neverland"	"Family, Drama, Biography"
10	"Don Juan DeMarco"	"Romance, Comedy, Drama"
11	"The Libertine"	"Romance, Drama"
12	"Fear and Loathing in Las Vegas"	"Adventure, Comedy, Drama"
13	"What's Eating Gilbert Grape"	"Romance, Drama"
14	"Platoon"	"Drama, War"
15	"Tusk"	"Comedy, Drama, Horror"

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).

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

```
1 PREFIX imdb: <http://academy.ontotext.com/imdb/>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT ?title WHERE {
5   ?movie a ?type ;
6         schema:genre ?genre ;
7         schema:name ?title .
8   FILTER (CONTAINS(LCASE(?genre), "fantasy"))
9   FILTER (?type = imdb:ColorMovie || ?type = imdb:BlackandWhiteMovie)
10 }
```

Вывод:

	title	Q	Movies
1	"Avatar"		
2	"Pirates of the Caribbean: At World's End"		
3	"Tangled"		
4	"Harry Potter and the Half-Blood Prince"		
5	"Pirates of the Caribbean: Dead Man's Chest"		
6	"Man of Steel"		
7	"The Chronicles of Narnia: Prince Caspian"		
8	"Pirates of the Caribbean: On Stranger Tides"		
9	"Men in Black 3"		
10	"The Hobbit: The Battle of the Five Armies"		
11	"The Amazing Spider-Man"		
12	"The Hobbit: The Desolation of Smaug"		
13	"The Golden Compass"		
14	"Spider-Man 2"		
15	"Alice in Wonderland"		
16	"X-Men: The Last Stand"		
17	"Monsters University"		
18	"Oz the Great and Powerful"		
19	"The Amazing Spider-Man 2"		
20	"Toy Story 3"		
21	"X-Men: Days of Future Past"		
22	"Jack the Giant Slayer"		
23	"Prince of Persia: The Sands of Time"		
24	"Indiana Jones and the Kingdom of the Crystal Skull"		
25	"The Good Dinosaur"		

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

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

```
1 PREFIX imdb: <http://academy.ontotext.com/imdb/>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT ?title ?commentCount WHERE {
5   ?movie a ?type ;
6       schema:name ?title ;
7       schema:countryOfOrigin <http://academy.ontotext.com/imdb/country/USA> ;
8       schema:commentCount ?commentCount .
9   FILTER (?commentCount > 500)
10  FILTER (?type = imdb:ColorMovie || ?type = imdb:BlackandWhiteMovie)
11 }
```

1	"The Hunger Games: Catching Fire"	"502"^^xsd:integer
2	"The Hobbit: The Desolation of Smaug"	"509"^^xsd:integer
3	"Captain America: The First Avenger"	"510"^^xsd:integer
4	"The Conjuring"	"511"^^xsd:integer
5	"Captain America: Civil War"	"516"^^xsd:integer
6	"Thor"	"516"^^xsd:integer
7	"Ant-Man"	"517"^^xsd:integer
8	"The Girl with the Dragon Tattoo"	"517"^^xsd:integer
9	"Star Trek"	"518"^^xsd:integer
10	"Dawn of the Planet of the Apes"	"521"^^xsd:integer
11	"Source Code"	"522"^^xsd:integer
12	"Oz the Great and Powerful"	"525"^^xsd:integer
13	"V for Vendetta"	"525"^^xsd:integer
14	"Dark Shadows"	"526"^^xsd:integer
15	"Rise of the Planet of the Apes"	"529"^^xsd:integer
16	"It Follows"	"533"^^xsd:integer
17	"Nightcrawler"	"534"^^xsd:integer
18	"Inside Llewyn Davis"	"535"^^xsd:integer
19	"Whiplash"	"535"^^xsd:integer
20	"Inside Out"	"536"^^xsd:integer
21	"The Grand Budapest Hotel"	"536"^^xsd:integer
22	"Lincoln"	"538"^^xsd:integer
23	"American Hustle"	"538"^^xsd:integer
24	"X-Men: Days of Future Past"	"539"^^xsd:integer
25	"Oblivion"	"539"^^xsd:integer

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

Заменял на: Фильмы, где режиссер также является актером

```
1 PREFIX imdb: <http://academy.ontotext.com/imdb/>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT ?title ?directorName WHERE {
5   ?movie a ?type ;
6         schema:name ?title ;
7         schema:director ?director ;
8         schema:actor ?director .
9   ?director schema:name ?directorName .
10  FILTER (?type = imdb:ColorMovie || ?type = imdb:BlackandWhiteMovie)
11 }
```

Вывод:

	title	directorName
1	"The Good Dinosaur"	"Peter Sohn"
2	"Iron Man"	"Jon Favreau"
3	"Iron Man 2"	"Jon Favreau"
4	"The Expendables"	"Sylvester Stallone"
5	"Oceans"	"Jacques Perrin"
6	"Star Trek: Insurrection"	"Jonathan Frakes"
7	"Ted 2"	"Seth MacFarlane"
8	"Ted"	"Seth MacFarlane"
9	"Absolute Power"	"Clint Eastwood"
10	"Blood Work"	"Clint Eastwood"
11	"Dick Tracy"	"Warren Beatty"
12	"Star Trek: First Contact"	"Jonathan Frakes"
13	"Animal Kingdom: Let's go Ape"	"Jamel Debbouze"
14	"A Million Ways to Die in the West"	"Seth MacFarlane"
15	"Against the Ropes"	"Charles S. Dutton"
16	"Obitaemyy ostrov"	"Fedor Bondarchuk"
17	"Reds"	"Warren Beatty"
18	"Million Dollar Baby"	"Clint Eastwood"
19	"Larry Crowne"	"Tom Hanks"
20	"Bulworth"	"Warren Beatty"
21	"The Boss"	"Ben Falcone"
22	"Winged Migration"	"Jacques Perrin"
23	"The Curse of the Jade Scorpion"	"Woody Allen"
24	"Gran Torino"	"Clint Eastwood"
25	"The True Story of Puss'N Boots"	"J��r��me Deschamps"

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

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