

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

Мареев Георгий Александрович БД-241м

Практическая работа 2.1 Часть 3. Рекомендательные системы. GraphDB

Вариант 13

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

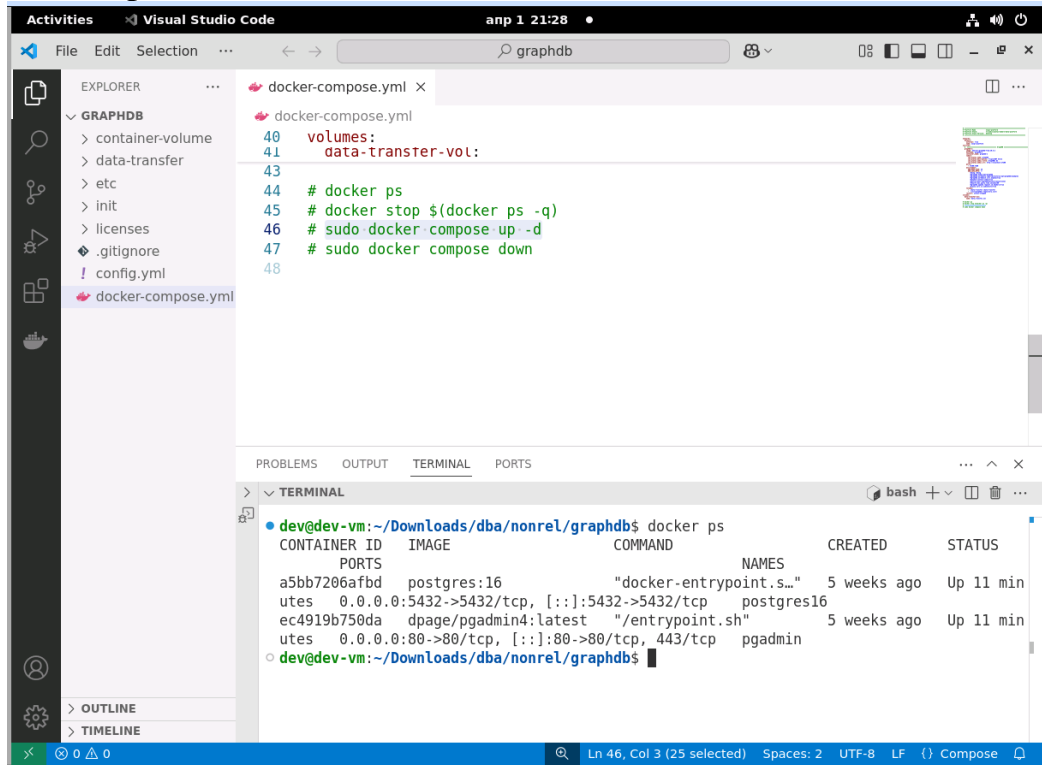
Москва
2025

Введение

Цель: освоение работы с базой данных GraphDB в виртуальной машине с использованием Docker, изучение RDF-данных, запросов с использованием SPARQL, анализ результатов, используя функциональность GraphDB.

Основная часть

Docker ps



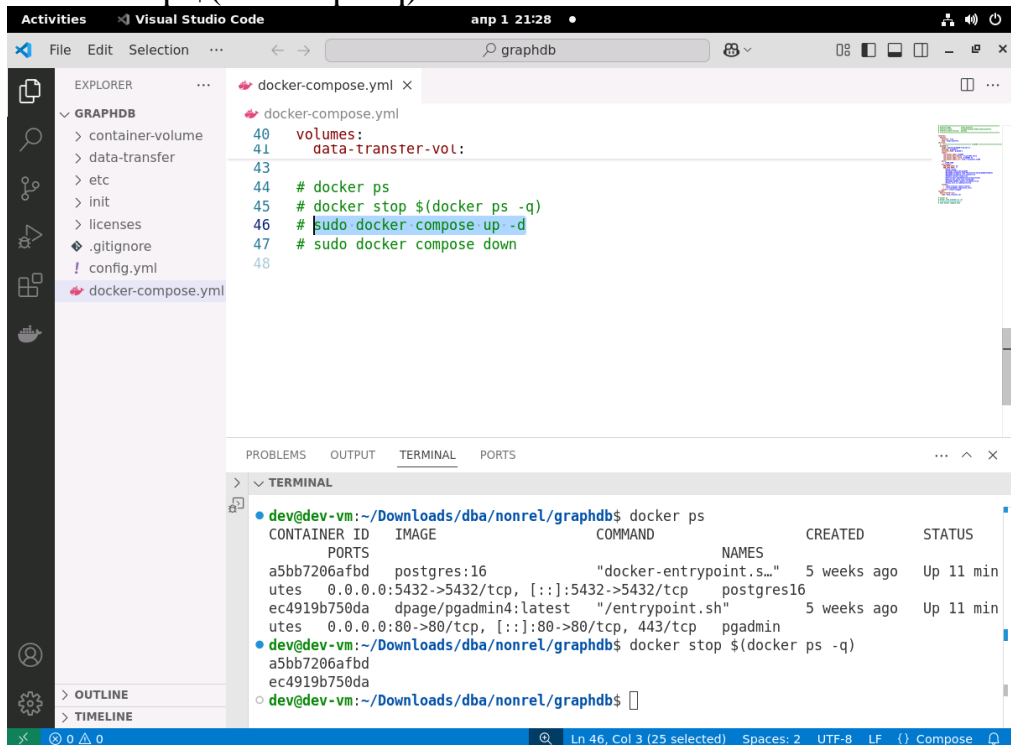
The screenshot shows the Visual Studio Code interface with a Docker Compose file open in the editor. The file is named `docker-compose.yml` and contains the following content:

```
40 volumes:
41   data-transfer-vol:
42
43
44 # docker ps
45 # docker stop $(docker ps -q)
46 # sudo docker compose up -d
47 # sudo docker compose down
48
```

The terminal window at the bottom shows the output of the `docker ps` command, displaying a table of running containers:

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS
a5bb7206afbd	postgres:16	"docker-entrypoint.s..."	postgres16	5 weeks ago	Up 11 min
ec4919b750da	dpape/pgadmin4:latest	"/entrypoint.sh"	pgadmin	5 weeks ago	Up 11 min

docker stop \$(docker ps -q)



The screenshot shows the Visual Studio Code interface with a Docker Compose file open in the editor. The file is named `docker-compose.yml` and contains the following content:

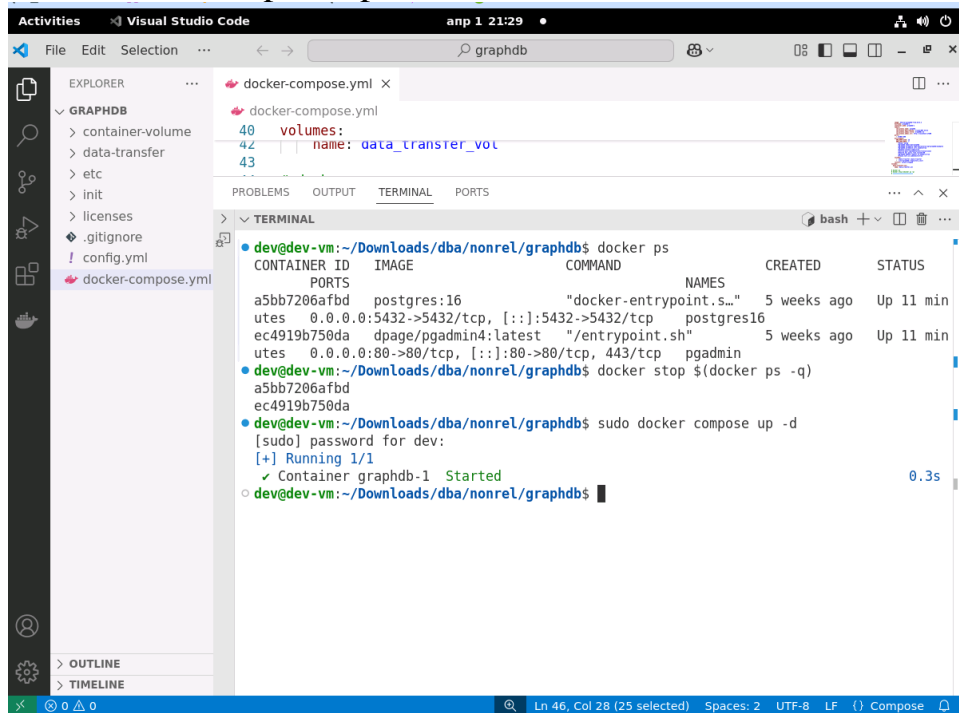
```
40 volumes:
41   data-transfer-vol:
42
43
44 # docker ps
45 # docker stop $(docker ps -q)
46 # sudo docker compose up -d
47 # sudo docker compose down
48
```

The terminal window at the bottom shows the output of the `docker ps` command, displaying a table of running containers:

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS
a5bb7206afbd	postgres:16	"docker-entrypoint.s..."	postgres16	5 weeks ago	Up 11 min
ec4919b750da	dpape/pgadmin4:latest	"/entrypoint.sh"	pgadmin	5 weeks ago	Up 11 min

Below the table, the terminal shows the execution of the `docker stop $(docker ps -q)` command, which successfully stops the containers.

sudo docker compose up -d



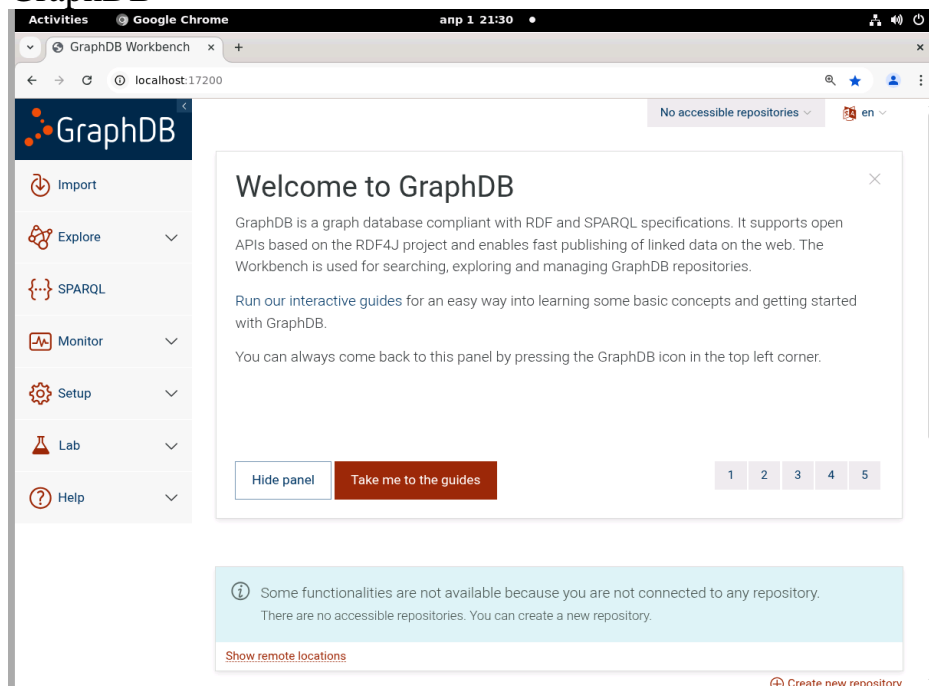
The screenshot shows the Visual Studio Code interface with the 'docker-compose.yml' file open in the editor. The file contains a 'volumes' section with a name 'data_transfer_vol'. The terminal window shows the following commands and output:

```
dev@dev-vm:~/Downloads/dba/nonrel/graphdb$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
a5bb7206afbd   postgres:16   "docker-entrypoint.s..."  5 weeks ago   Up 11 min
utes   0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp   postgres16
ec4919b750da   dpape/pgadmin4:latest   "/entrypoint.sh"        5 weeks ago   Up 11 min
utes   0.0.0.0:80->80/tcp, [::]:80->80/tcp, 443/tcp   pgadmin

dev@dev-vm:~/Downloads/dba/nonrel/graphdb$ docker stop $(docker ps -q)
a5bb7206afbd
ec4919b750da

dev@dev-vm:~/Downloads/dba/nonrel/graphdb$ sudo docker compose up -d
[sudo] password for dev:
[+] Running 1/1
  ✓ Container graphdb-1 Started                                0.3s
dev@dev-vm:~/Downloads/dba/nonrel/graphdb$
```

GraphDB



Создание БД

The image shows two screenshots of the GraphDB web interface in a Google Chrome browser window.

Top Screenshot: Import Page

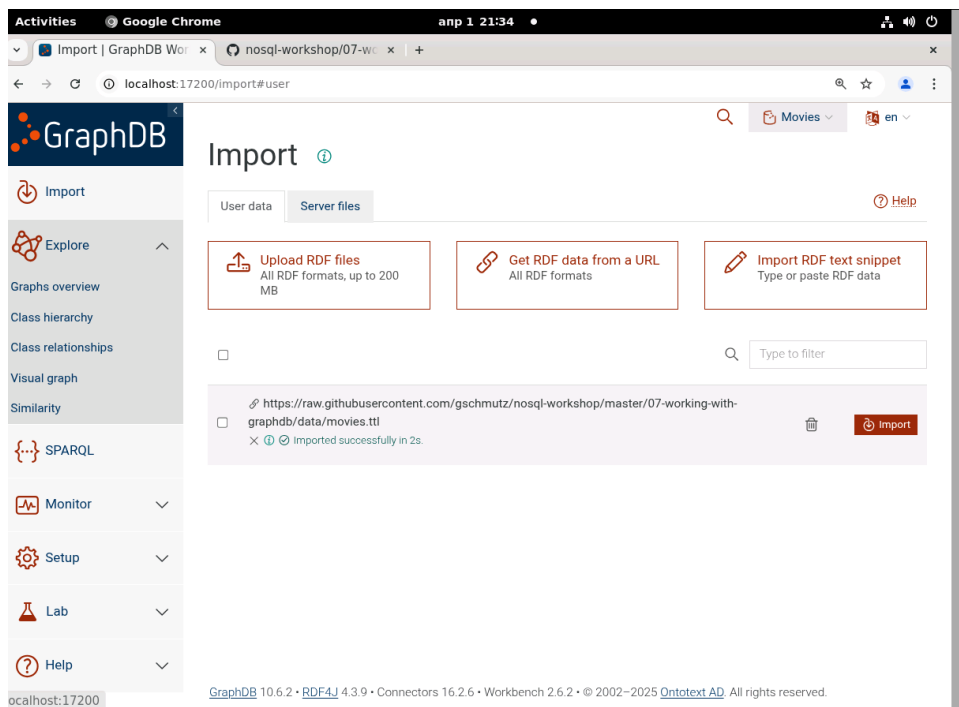
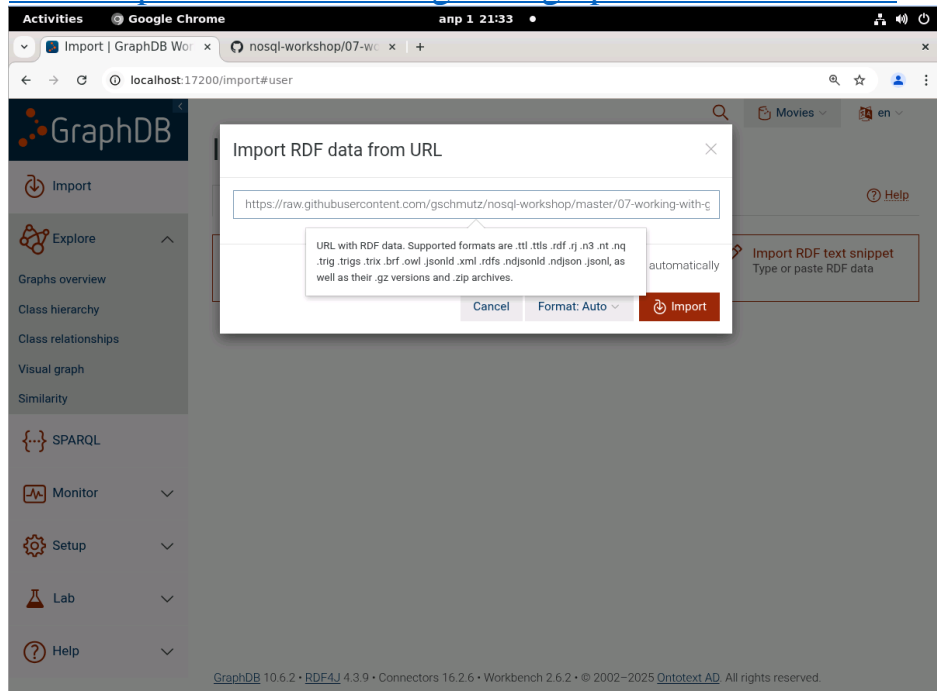
- URL:** `localhost:17200/import`
- Page Title:** Import
- Message:** "Some functionalities are not available because you are not connected to any repository. There are no accessible writable repositories. You can create a new repository." Below this is a link "Show remote locations".
- Buttons:** "Create new repository" (with a plus icon).
- Footer:** GraphDB 10.6.2 • RDF4J 4.3.9 • Connectors 16.2.6 • Workbench 2.6.2 • © 2002–2025 Ontotext AD. All rights reserved.

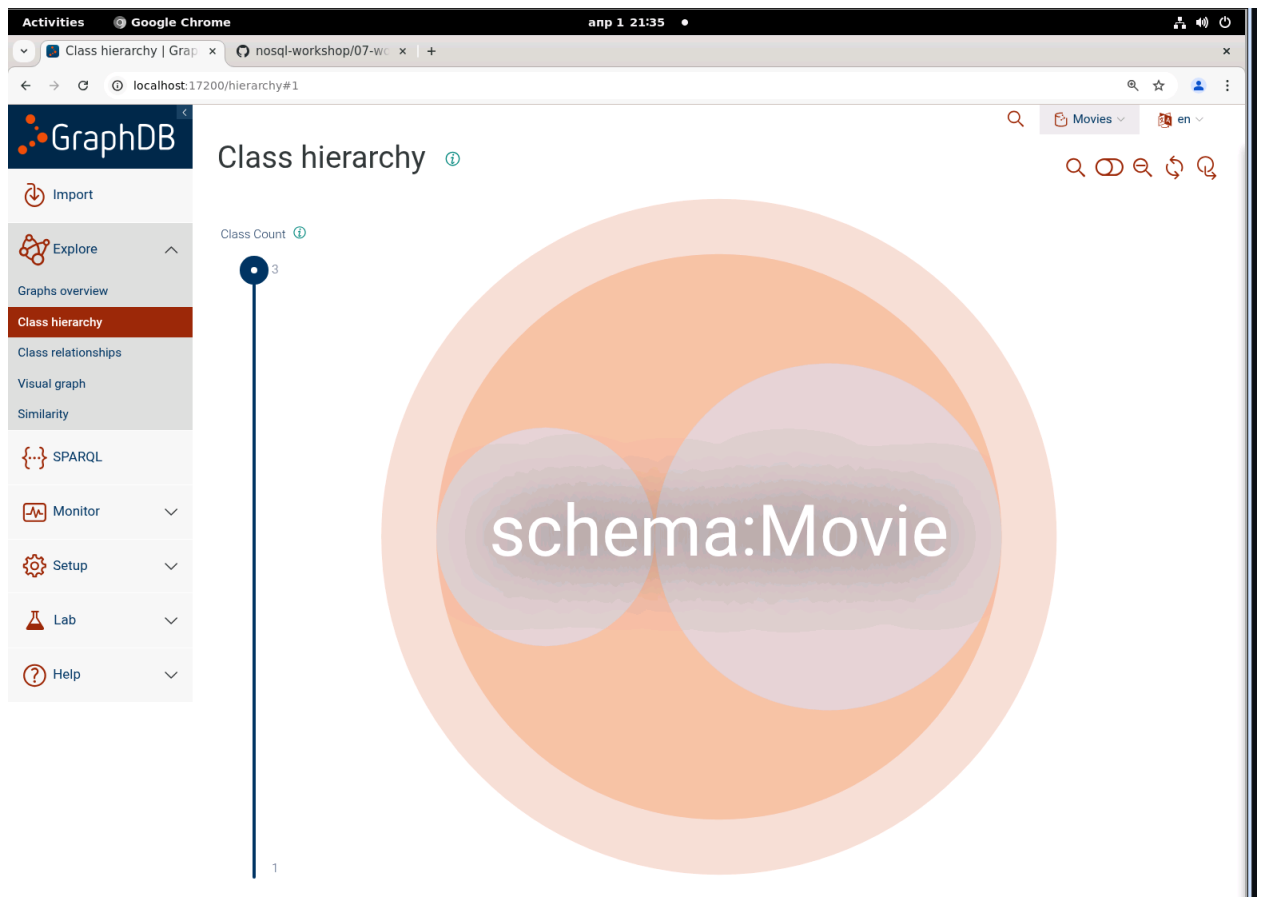
Bottom Screenshot: Create GraphDB repository Page

- URL:** `localhost:17200/repository/create/graphdb?previous=home`
- Page Title:** Create GraphDB repository
- Form Fields:**
 - Repository ID*:** Movies
 - Repository description:** Praktika
 - Read-only:** ☐
- Inference and Validation:**
 - Ruleset:** RDFS-Plus (Optimizer) (dropdown menu)
 - Buttons:** Custom ruleset...
 - Options:**
 - ☒ Disable owl:sameAs
 - ☐ Enable consistency checks
 - ☐ Enable SHACL validation (with a link to SHACL options)
- Indexing:**
 - Entity ID size:** 32-bit (selected), 40-bit
 - ☐ Enable context index
 - ☒ Enable predicate list index
 - ☐ Enable full-text search (FTS) index

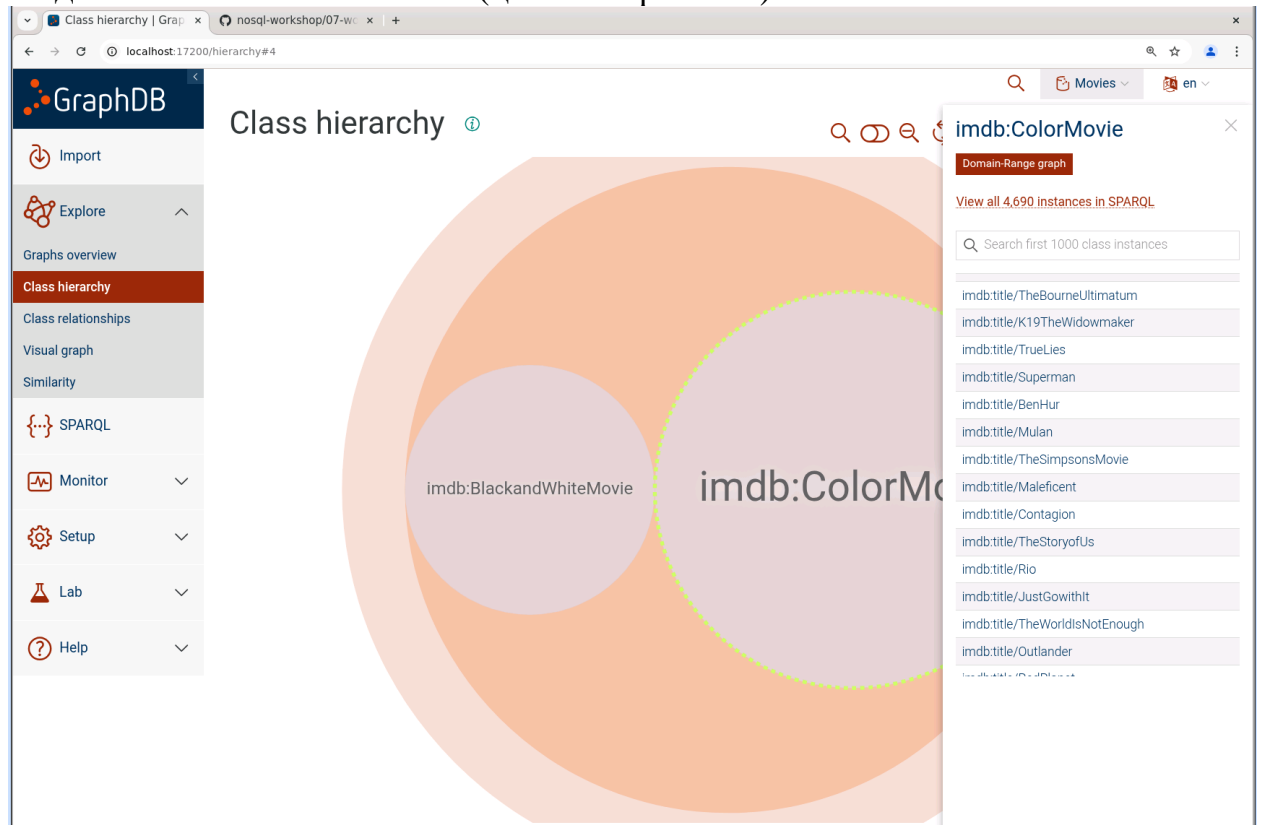
Импорт данных

Источник: <https://raw.githubusercontent.com/gschmutz/nosql-workshop/master/07-working-with-graphdb/data/movies.ttl>

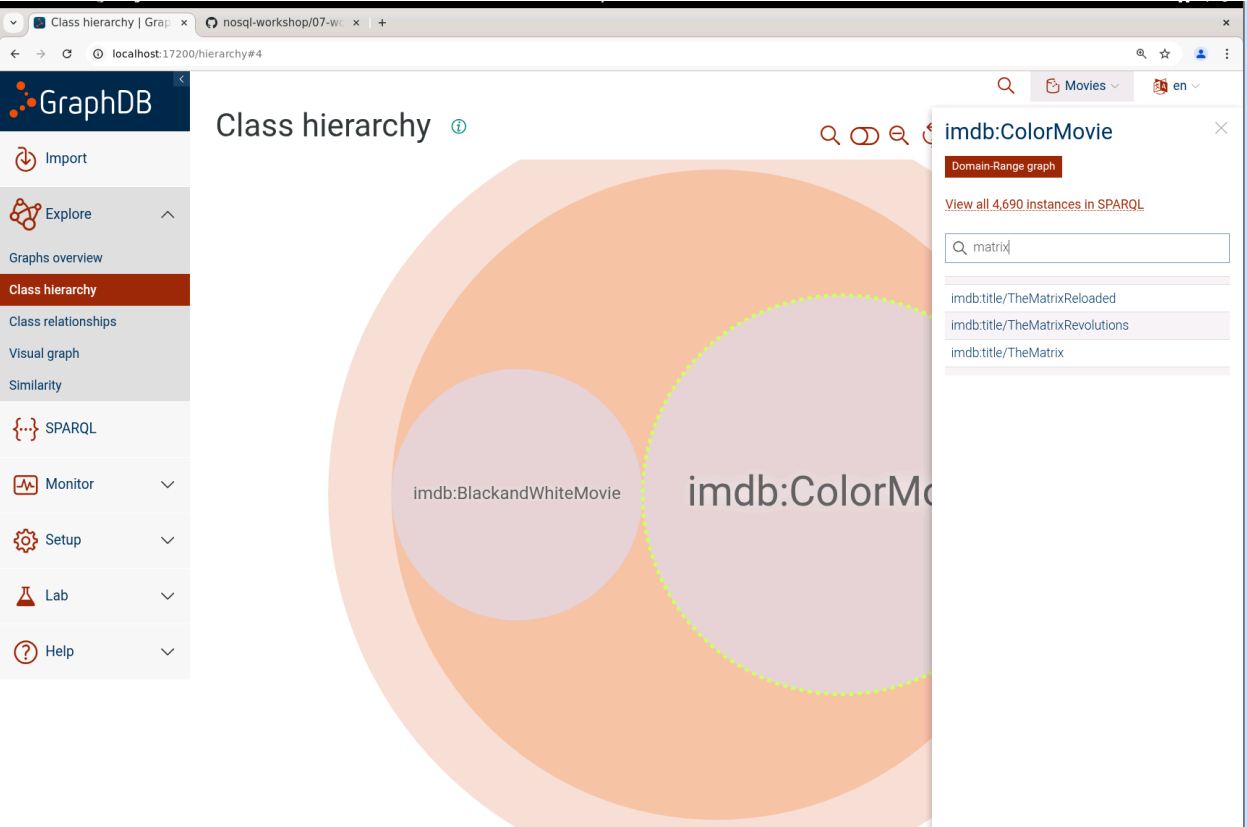




Подкласс «imdb:ColorMovie»(цветные фильмы)



Фильм Matrix



Связи фильма Matrix

TheMatrix

Source: <http://academy.ontotext.com/imdb/title/TheMatrix>

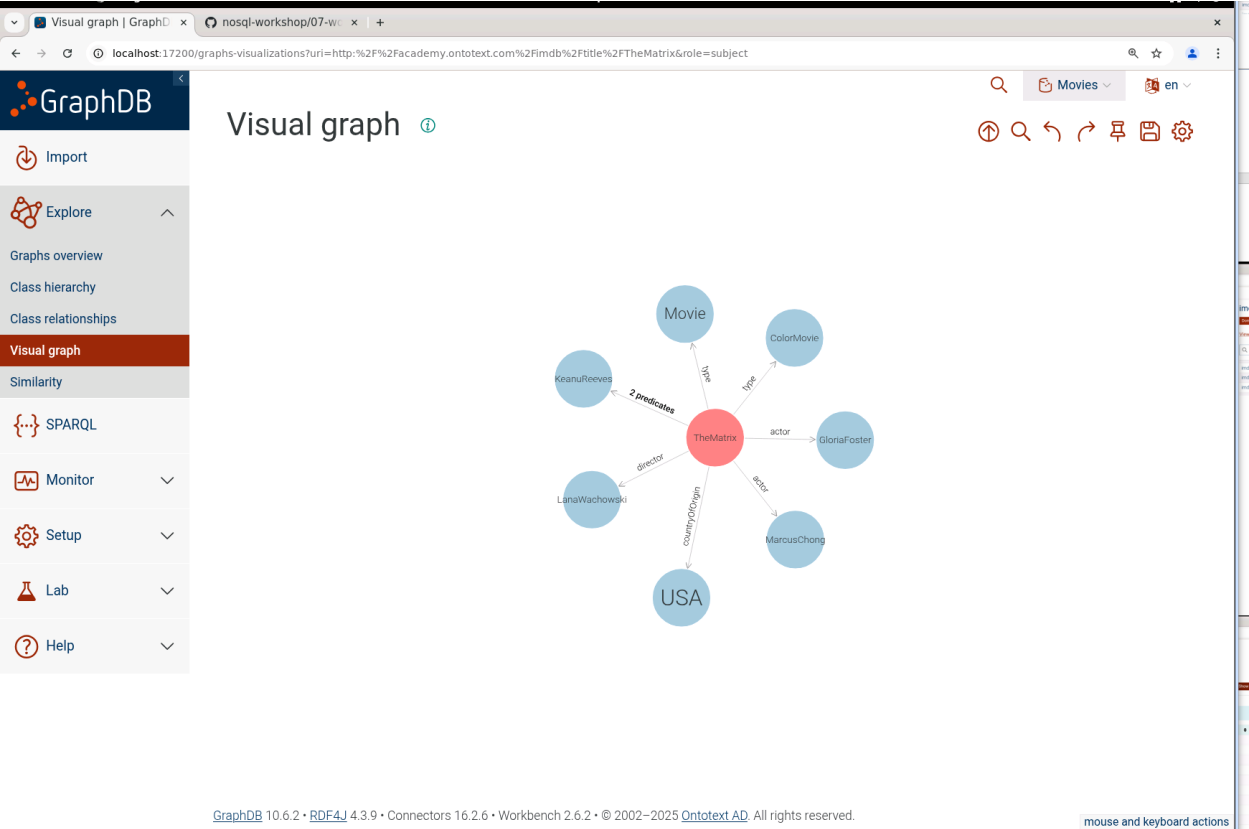
subject predicate object context all

Explicit only Show Blank Nodes Download as Visual graph

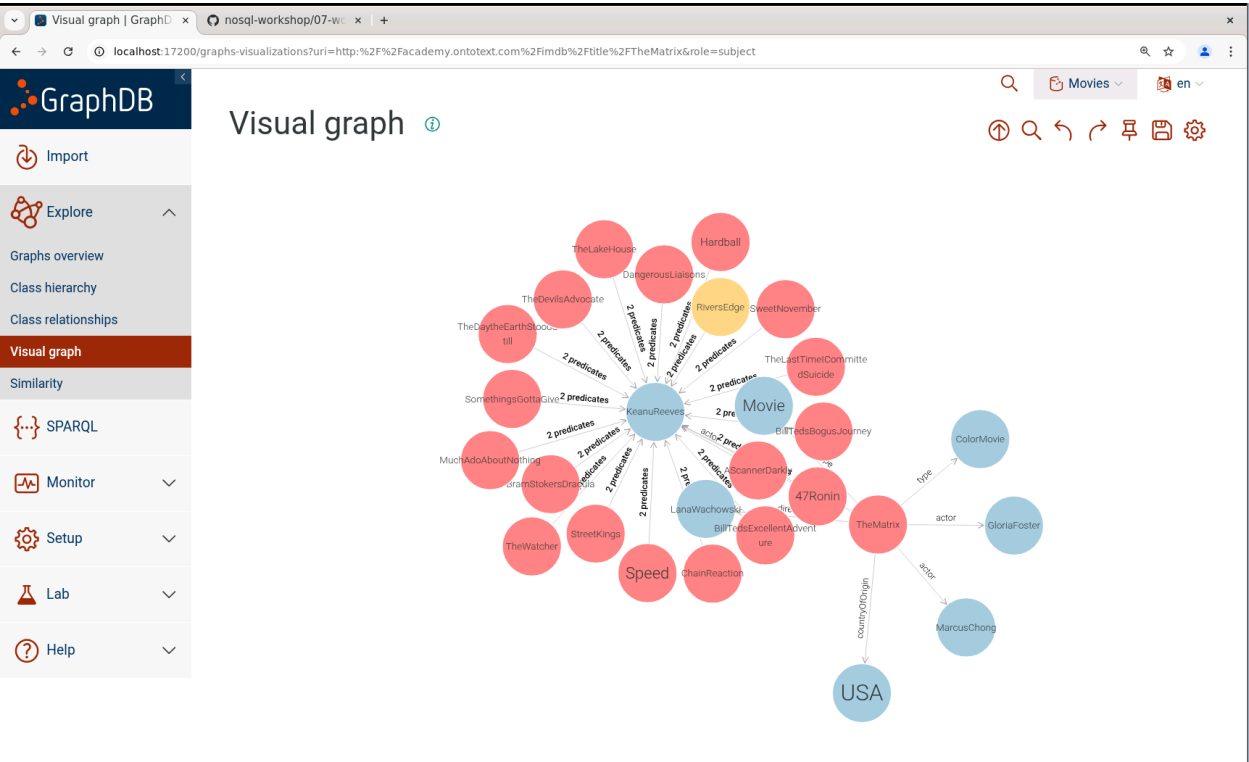
Filter query results Compact view Hide row numbers

	subject	predicate	object	context
1	imdb:title/TheMatrix	imdb:leadActor	imdb:person/KeanuReeves	http://www.ontotext.com/explicit
2	imdb:title/TheMatrix	schema:actor	imdb:person/GloriaFoster	http://www.ontotext.com/explicit
3	imdb:title/TheMatrix	schema:actor	imdb:person/MarcusChong	http://www.ontotext.com/explicit
4	imdb:title/TheMatrix	schema:commentCount	"313"^^xsd:integer	http://www.ontotext.com/explicit
5	imdb:title/TheMatrix	schema:countryOfOrigin	imdb:country/USA	http://www.ontotext.com/explicit
6	imdb:title/TheMatrix	schema:director	imdb:person/LanaWachowski	http://www.ontotext.com/explicit
7	imdb:title/TheMatrix	schema:genre	"Action"	http://www.ontotext.com/explicit
8	imdb:title/TheMatrix	schema:genre	"Sci-Fi"	http://www.ontotext.com/explicit
9	imdb:title/TheMatrix	schema:name	"The Matrix"	http://www.ontotext.com/explicit
10	imdb:title/TheMatrix	rdf:type	imdb:ColorMovie	http://www.ontotext.com/explicit

Visual Graph фильма Matrix



Keanu Reeves



SPARQL

Все триплеты в графе (100 записей)

The screenshot shows the GraphDB SPARQL Query & Update interface. The query editor contains the following SPARQL query:

```
1 select * where {
2   ?s ?p ?o .
3 } limit 100
```

The results are displayed in a table format with columns s, p, and o. The results show various RDF properties and their relationships.

	s	p	o
1	rdf:type	rdf:type	rdf:Property
2	rdfs:subPropertyOf	rdf:type	rdf:Property
3	rdfs:subPropertyOf	rdf:type	owl:TransitiveProperty
4	rdfs:subClassOf	rdf:type	rdf:Property
5	rdfs:subClassOf	rdf:type	owl:TransitiveProperty
6	rdfs:domain	rdf:type	rdf:Property
7	rdfs:range	rdf:type	rdf:Property
8	owl:equivalentProperty	rdf:type	owl:SymmetricProperty
9	owl:equivalentProperty	rdf:type	owl:TransitiveProperty

Все триплеты, связанные с фильмом “Pirates of the Caribbean: At World's End”

The screenshot shows the GraphDB SPARQL Query & Update interface. The query editor contains the following SPARQL query:

```
1 select * where {
2   <http://academy.ontotext.com/imdb/title/PiratesoftheCaribbeanAtWorldsEnd> ?p ?o .
3 }
```

The results are displayed in a table format with columns p and o. The results show various RDF properties and their relationships related to the movie.

	p	o
1	rdf:type	schema:Movie
2	rdf:type	imdb:ColorMovie
3	imdb:leadActor	imdb:person/JohnnyDepp
4	schema:actor	imdb:person/JohnnyDepp
5	schema:actor	imdb:person/OrlandoBloom
6	schema:actor	imdb:person/JackDavenport
7	schema:name	"Pirates of the Caribbean: At World's End"
8	schema:genre	"Action"
9	schema:genre	"Adventure"

С использованием префикса

GraphDB

Import

Explore

SPARQL

Monitor

Setup

Lab

Help

SPARQL Query & Update

Editor onlyEditor and resultsResults only

1 • PREFIX imdb: <http://academy.ontotext.com/imdb/>
2
3 • select * where {
4 imdb:title/PiratesoftheCaribbeanAtWorldsEnd ?p ?o .
5 }

Run

keyboard shortcuts

TableRaw responsePivot TableGoogle Chart

Filter query resultsCompact viewHide row numbersShowing results from 0 to 13 of 13. Query took 0.1s, moments ago.

	p	o
1	rdf:type	schema:Movie
2	rdf:type	imdb:ColorMovie
3	imdb:leadActor	imdb:person/JohnnyDepp
4	schema:actor	imdb:person/JohnnyDepp
5	schema:actor	imdb:person/OrlandoBloom
6	schema:actor	imdb:person/JackDavenport
7	schema:name	"Pirates of the Caribbean: At World's End"
8	schema:genre	"Action"
9	schema:genre	"Adventure"

Цветные фильмы, их названия и количество комментариев в порядке убывания (100 записей)

GraphDB

Import

Explore

SPARQL

Monitor

Setup

Lab

Help

SPARQL Query & Update

Editor onlyEditor and resultsResults only

1 • PREFIX imdb: <http://academy.ontotext.com/imdb/>
2 • PREFIX schema: <http://schema.org/>
3
4 • SELECT ?movie ?name ?commentCount
5 • WHERE {
6 ?movie a imdb:ColorMovie ;
7 schema:name ?name ;
8 schema:commentCount ?commentCount .
9 }
10 ORDER BY DESC(?commentCount)
11 LIMIT 100

Run

keyboard shortcuts

TableRaw responsePivot TableGoogle Chart

Filter query resultsCompact viewHide row numbersShowing results from 0 to 100 of 100. Query took 0.1s, moments ago.

	movie	name	commentCount
1	imdb:title/TheDarkKnightRises	"The Dark Knight Rises"	"813"^^xsd:integer
2	imdb:title/Prometheus	"Prometheus"	"775"^^xsd:integer
3	imdb:title/DjangoUnchained	"Django Unchained"	"765"^^xsd:integer
4	imdb:title/Skyfall	"Skyfall"	"750"^^xsd:integer
5	imdb:title/MadMaxFuryRoad	"Mad Max: Fury Road"	"739"^^xsd:integer
6	imdb:title/Gravity	"Gravity"	"738"^^xsd:integer
7	imdb:title/ManofSteel	"Man of Steel"	"733"^^xsd:integer
8	imdb:title/Avatar	"Avatar"	"723"^^xsd:integer
9	imdb:title/Interstellar	"Interstellar"	"712"^^xsd:integer

SPARQL Query & Update

```
1. PREFIX imdb: <http://academy.ontotext.com/imdb/>
2. PREFIX schema: <http://schema.org/>
3.
4. SELECT * {
5.   ?movie a imdb:ColorMovie ;
6.         schema:name ?movieName ;
7.         schema:commentCount ?commentCount .
8. } ORDER BY DESC(?commentCount)
```

Press Alt+Enter to autocomplete

Table Raw response Pivot Table Google Chart

Filter query results Compact view Hide row numbers Showing results from 0 to 1000 of 4648. Query took 0.1s, moments ago.

	movie	movieName	commentCount
1	imdb:title/TheDarkKnightRises	"The Dark Knight Rises"	"813"^^xsd:integer
2	imdb:title/Prometheus	"Prometheus"	"775"^^xsd:integer
3	imdb:title/DjangoUnchained	"Django Unchained"	"765"^^xsd:integer
4	imdb:title/Skyfall	"Skyfall"	"750"^^xsd:integer
5	imdb:title/MadMaxFuryRoad	"Mad Max: Fury Road"	"739"^^xsd:integer
6	imdb:title/Gravity	"Gravity"	"738"^^xsd:integer
7	imdb:title/ManOfSteel	"Man of Steel"	"733"^^xsd:integer
8	imdb:title/Avatar	"Avatar"	"723"^^xsd:integer
9	imdb:title/Interstellar	"Interstellar"	"712"^^xsd:integer

RDF-утверждения, где один и тот же человек одновременно является и режиссёром и главным актером.

SPARQL Query & Update

```
1. PREFIX schema: <http://schema.org/>
2. PREFIX imdb: <http://academy.ontotext.com/imdb/>
3.
4. SELECT * {
5.   ?movie schema:director ?person ;
6.         imdb:leadActor ?person .
7. } ORDER BY ?person
```

Table Raw response Pivot Table Google Chart

Filter query results Compact view Hide row numbers Showing results from 0 to 63 of 63. Query took 0.1s, moments ago.

	movie	person
1	imdb:title/TaxiToTheDarkSide	imdb:person/AlexGibney
2	imdb:title/FacingTheGiants	imdb:person/AlexKendrick
3	imdb:title/MutualAppreciation	imdb:person/AndrewBujalski
4	imdb:title/FunnyHaHa	imdb:person/AndrewBujalski
5	imdb:title/WaltzWithBashir	imdb:person/AriFolman
6	imdb:title/SpaceCowboys	imdb:person/ClintEastwood
7	imdb:title/AbsolutePower	imdb:person/ClintEastwood
8	imdb:title/BloodWork	imdb:person/ClintEastwood

Подсчет фильмов

SPARQL Query & Update

nosql-workshop/07-wi

localhost:17200/sparql

GraphDB

Import

Explore

SPARQL

Monitor

Setup

Lab

Help

SPARQL Query & Update

Editor only

Editor and results

Results only

en

Unnamed

1 PREFIX schema: <http://schema.org/>

2 PREFIX imdb: <http://academy.ontotext.com/imdb/>

3

4 SELECT ?person (COUNT(?movie) as ?numMovies) {

5 ?movie schema:director ?person ;

6 imdb:leadActor ?person .

7 } GROUP BY ?person ORDER BY DESC(?numMovies)

Press Alt+Enter to autocomplete

keyboard shortcuts

Run

Table

Raw response

Pivot Table

Google Chart

Download as

Filter query results

Compact view

Hide row numbers

Showing results from 0 to 38 of 38. Query took 0.1s, moments ago.

	person	numMovies
1	imdb:person/ClintEastwood	*10**xsd:integer
2	imdb:person/WoodyAllen	*10**xsd:integer
3	imdb:person/MichaelMoore	*4**xsd:integer
4	imdb:person/TomHanks	*2**xsd:integer
5	imdb:person/LeonardNimoy	*2**xsd:integer
6	imdb:person/DenzelWashington	*2**xsd:integer
7	imdb:person/AndrewBujalski	*2**xsd:integer
8	imdb:person/JamelDebbouze	*1**xsd:integer
9	imdb:person/WarrenBeatty	*1**xsd:integer

SPARQL Query & Update

nosql-workshop/07-wi

localhost:17200/sparql

GraphDB

Import

Explore

SPARQL

Monitor

Setup

Lab

Help

SPARQL Query & Update

Editor only

Editor and results

Results only

en

Unnamed

1 PREFIX imdb: <http://academy.ontotext.com/imdb/>

2 PREFIX schema: <http://schema.org/>

3

4 SELECT ?person ?personName (COUNT(?movie) AS ?numMovies)

5 WHERE {

6 ?movie schema:name ?name ;

7 schema:director ?person ;

8 imdb:leadActor ?person .

9 ?person schema:name ?personName .

10 }

11 GROUP BY ?person ?personName

12 ORDER BY DESC(?numMovies)

13 LIMIT 100

keyboard shortcuts

Run

Table

Raw response

Pivot Table

Google Chart

Download as

Filter query results

Compact view

Hide row numbers

Showing results from 0 to 38 of 38. Query took 0.1s, moments ago.

	person	personName	numMovies
1	imdb:person/ClintEastwood	"Clint Eastwood"	*10**xsd:integer
2	imdb:person/WoodyAllen	"Woody Allen"	*10**xsd:integer
3	imdb:person/MichaelMoore	"Michael Moore"	*4**xsd:integer
4	imdb:person/TomHanks	"Tom Hanks"	*2**xsd:integer
5	imdb:person/LeonardNimoy	"Leonard Nimoy"	*2**xsd:integer
6	imdb:person/DenzelWashington	"Denzel Washington"	*2**xsd:integer
7	imdb:person/AndrewBujalski	"Andrew Bujalski"	*2**xsd:integer
8	imdb:person/JamelDebbouze	"Jamel Debbouze"	*1**xsd:integer

Индивидуальное задание. Вариант 13

1. Найдите все фильмы, снятые до 1970 года.
2. Напишите запрос для получения всех фильмов с рейтингом 7 и выше.
3. Найдите все фильмы, в которых снимались два актера: "Scarlett Johansson" и "Chris Evans".
4. Создайте запрос для получения всех фильмов, снятых в жанре триллер.
5. Напишите запрос для получения всех фильмов и их жанра, отсортировав по году.

Задание 1. Все фильмы, снятые до 1970 года

В файле с данными по [ссылке](#), нет информации о годе выпуска фильма ("year", "schema:year", "imbd:year")

Поэтому запрос на поиск фильмов до 1970 года выдаст пустой результат

Предполагаемый запрос:

GraphDB 10.6.2 • RDF4J 4.3.9 • Connectors 16.2.6 • Workbench 2.6.2 • © 2002–2025 Ontotext AD. All rights reserved.

Задание 2. Все фильмы с рейтингом 7 и выше

Нет информации о рейтинге фильмов

Предполагаемый запрос:

GraphDB

Import

Explore

SPARQL

Monitor

Setup

Lab

Help

SPARQL Query & Update

Editor onlyEditor and resultsResults only

1 PREFIX imdb: <http://academy.ontotext.com/imdb/>

2 PREFIX schema: <http://schema.org/>

3

4 SELECT ?movie ?name ?rating

5 WHERE {

6 ?movie schema:name ?name ;

7 schema:rating ?rating .

8 FILTER(?rating >= 7)

9 }

10 ORDER BY DESC(?rating)

Press Alt+Enter to autocomplete

Run

keyboard shortcuts

TableRaw responsePivot TableGoogle Chart

Filter query resultsCompact viewHide row numbers

No results. Query took 0.1s, moments ago.

movie

name

rating

No data available in table

GraphDB 10.6.2 • RDF4J 4.3.9 • Connectors 16.2.6 • Workbench 2.6.2 • © 2002–2025 Ontotext AD. All rights reserved.

Задание 3. Все фильмы, в которых снимались два актера: "Scarlett Johansson" и "Chris Evans"

GraphDB

Import

Explore

SPARQL

Monitor

Setup

Lab

Help

SPARQL Query & Update

Editor onlyEditor and resultsResults only

1 PREFIX imdb: <http://academy.ontotext.com/imdb/>

2 PREFIX schema: <http://schema.org/>

3

4 SELECT DISTINCT ?movie ?title

5 WHERE {

6 ?scarlett schema:name "Scarlett Johansson" .

7 ?chris schema:name "Chris Evans" .

8

9 ?movie a ?type ;

10 schema:name ?title ;

11 schema:actor ?scarlett , ?chris .

12

13 }

Run

keyboard shortcuts

TableRaw responsePivot TableGoogle Chart

Filter query resultsCompact viewHide row numbers

Showing results from 0 to 2 of 2. Query took 0.1s, moments ago.

movie

title

1 imdb:title/CaptainAmericaCivilWar

"Captain America: Civil War"

2 imdb:title/CaptainAmericaTheWinterSoldier

"Captain America: The Winter Soldier"

GraphDB 10.6.2 • RDF4J 4.3.9 • Connectors 16.2.6 • Workbench 2.6.2 • © 2002–2025 Ontotext AD. All rights reserved.

Задание 4. Все фильмы, снятые в жанре триллер.

GraphDB

Import

Explore

SPARQL

Monitor

Setup

Lab

Help

SPARQL Query & Update

Editor onlyEditor and resultsResults only

1PREFIX imdb: <http://academy.ontotext.com/imdb/>
2PREFIX schema: <http://schema.org/>
3
4SELECT ?movie ?title
5WHERE {
6 ?movie a ?type ;
7 schema:name ?title ;
8 schema:genre "Thriller" .
9 }
10 }

Run

keyboard shortcuts

TableRaw responsePivot TableGoogle Chart

Download as

123

Filter query resultsCompact viewHide row numbersShowing results from 0 to 1000 of 2710. Query took 0.1s, moments ago.

	movie	title
1	imdb:title/Spectre	"Spectre"
2	imdb:title/Spectre	"Spectre"
3	imdb:title/TheDarkKnightRises	"The Dark Knight Rises"

Задание 5. Все фильмы и их жанры, отсортированные по году.
*Так как информации о годе нет, были отсортированы по “стране производства”, в алфавитном порядке

GraphDB

Import

Explore

SPARQL

Monitor

Setup

Lab

Help

SPARQL Query & Update

Editor onlyEditor and resultsResults only

1PREFIX imdb: <http://academy.ontotext.com/imdb/>
2PREFIX schema: <http://schema.org/>
3
4SELECT ?countryName ?movieTitle (GROUP_CONCAT(DISTINCT ?genre; SEPARATOR=", ") AS ?genres)
5WHERE {
6 ?movie a ?type ;
7 schema:name ?movieTitle ;
8 schema:genre ?genre ;
9 schema:countryOfOrigin ?country .
10
11 ?country schema:name ?countryName .
12
13 FILTER(?type IN (imdb:ColorMovie, imdb:BlackandWhiteMovie))
14 }
15 GROUP BY ?countryName ?movieTitle
16 ORDER BY ?countryName ?movieTitle

Run

keyboard shortcuts

TableRaw responsePivot TableGoogle Chart

Download as

12345

Filter query resultsCompact viewHide row numbersShowing results from 0 to 1000 of 4897. Query took 0.3s, moments ago.

	countryName	movieTitle	genres
1	"Afghanistan"	"Osama"	"Drama"
2	"Argentina"	"Live-In Maid"	"Drama"
3	"Argentina"	"Nine Queens"	"Thriller, Drama, Crime"
4	"Argentina"	"The Holy Girl"	"Drama"
5	"Argentina"	"The Secret In Their Eyes"	"Thriller, Mystery, Drama"
6	"Aruba"	"Knock Off"	"Action, Thriller, Comedy"
7	"Aruba"	"Knock Off"	"Action, Thriller, Comedy"

Заключение

В результате выполненной работы были успешно освоены основные принципы работы с базой данных GraphDB в виртуальной машине с использованием Docker. В ходе выполнения задания изучены методы загрузки и обработки RDF-данных, а также составления и выполнения SPARQL-запросов для извлечения и анализа информации.