

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:
Нереляционные базы данных

Практическая работа №4.2

Тема:

«Рекомендательные системы. GraphDB».

Выполнил(а): Буздуган М., группа: АБП-231
Преподаватель: Босенко Т.М

Москва
2025

Цели: Освоение работы с базой данных GraphDB в виртуальной машине с использованием Docker. Научиться загружать RDF-данные в базу, выполнять запросы с использованием SPARQL и анализировать результаты, используя функциональность GraphDB.

Для начала подключаюсь к виртуальной машине и захожу в нее, перехожу в каталог с проектом, останавливаю контейнер и запускаю контейнер с базой данных GraphDB:

```
● 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
```

После этого копирую представленную ссылку:

<https://raw.githubusercontent.com/gschmutz/nosql-workshop/master/07-working-with-graphdb/data/movies.ttl> и вставляю ее в новый репозиторий под названием «Movies».

Далее выполняю запросы, представленные в примерах и приступаю к самостоятельным задачам

1. Напишите запрос, который вернет все фильмы, у которых количество комментариев больше 200.

```

1 PREFIX schema: <http://schema.org/>
2 SELECT ?movieTitle WHERE {
3   ?movie a schema:Movie ;
4           schema:name ?movieTitle ;
5           schema:commentCount ?commentCount .
6   FILTER (?commentCount > 200)
7 }

```

	movieTitle
1	"Cinderella Man"
2	"The Pursuit of Happyness"
3	"Journey to the Center of the Earth"
4	"Sex Tape"
5	"Colombiana"
6	"The Kite Runner"
7	"The Savages"
8	"Escape from New York"
9	"The Mummy Returns"
10	"The Last Witch Hunter"
11	"The League of Extraordinary Gentlemen"
12	"Elizabeth: The Golden Age"
13	"Seven Pounds"
14	"Lucky Number Slevin"
15	"House of 1000 Corpses"
16	"25th Hour"

2. Создайте запрос для отображения всех фильмов, выпущенных в 2000-е ГОДЫ.

```

1 PREFIX schema: <http://schema.org/>
2 SELECT ?movieTitle WHERE {
3   ?movie a schema:Movie ;
4           schema:name ?movieTitle ;
5           schema:datePublished ?date .
6   FILTER (YEAR(?date) ≥ 2000 && YEAR(?date) ≤ 2009)
7 }

```

Filter query results

Compact view ☐ Hide row numbers ☐

No results. Query took 0.1s, on 2025-04-05 at 14:24.

Запрос не выдал результат, так как в изначально созданной базе данных нет никакой информации о годе выпуска фильма

3. Найдите режиссеров, которые сняли больше 5 фильмов.

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

```
SELECT ?directorName (COUNT(?movie) AS ?movieCount) WHERE {  
  ?movie a schema:Movie ;  
         schema:director ?director .  
  ?director schema:name ?directorName .  
}  
GROUP BY ?directorName  
HAVING (COUNT(?movie) > 5)
```

	directorName	movieCount
1	"James Cameron"	"7^^xsd:integer"
2	"Gore Verbinski"	"7^^xsd:integer"
3	"Sam Mendes"	"7^^xsd:integer"
4	"Christopher Nolan"	"8^^xsd:integer"
5	"Sam Raimi"	"11^^xsd:integer"
6	"Zack Snyder"	"7^^xsd:integer"
7	"Bryan Singer"	"8^^xsd:integer"
8	"Marc Forster"	"8^^xsd:integer"
9	"Barry Sonnenfeld"	"7^^xsd:integer"
10	"Peter Jackson"	"9^^xsd:integer"
11	"Ridley Scott"	"16^^xsd:integer"
12	"Peter Berg"	"6^^xsd:integer"
13	"Tim Burton"	"14^^xsd:integer"
14	"Brett Ratner"	"9^^xsd:integer"
15	"Michael Bay"	"12^^xsd:integer"
16	"Martin Campbell"	"8^^xsd:integer"
17	"M. G."	"6^^xsd:integer"

4. Выведите все фильмы, в которых снимался "Johnny Depp".

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

```
SELECT ?movieTitle WHERE {  
  ?movie a schema:Movie ;  
         schema:name ?movieTitle ;  
         schema:actor ?actor .  
  ?actor schema:name "Johnny Depp" .  
}
```

	movieTitle
1	"Pirates of the Caribbean: At World's End"
2	"Pirates of the Caribbean: Dead Man's Chest"
3	"The Lone Ranger"
4	"Pirates of the Caribbean: On Stranger Tides"
5	"Alice in Wonderland"
6	"Alice Through the Looking Glass"
7	"Charlie and the Chocolate Factory"
8	"Dark Shadows"
9	"Rango"
10	"Pirates of the Caribbean: The Curse of the Black Pearl"
11	"Public Enemies"
12	"The Tourist"
13	"Transcendence"
14	"Mortdecai"
15	"Black Mass"
16	"Into the Woods"

5. Напишите запрос, который сортирует фильмы по количеству комментариев.

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

```
SELECT ?movieTitle ?commentCount WHERE {  
  ?movie a schema:Movie ;  
         schema:name ?movieTitle ;  
         schema:commentCount ?commentCount .  
}  
ORDER BY DESC(?commentCount)|
```

	movieTitle	commentCount
1	"The Dark Knight Rises"	*813**xsd:integer
2	"Prometheus"	*775**xsd:integer
3	"Django Unchained"	*765**xsd:integer
4	"Skyfall"	*750**xsd:integer
5	"Mad Max: Fury Road"	*739**xsd:integer
6	"Gravity"	*738**xsd:integer
7	"Man of Steel"	*733**xsd:integer
8	"Avatar"	*723**xsd:integer
9	"Interstellar"	*712**xsd:integer
10	"The Avengers"	*703**xsd:integer
11	"Hugo"	*682**xsd:integer
12	"Drive"	*676**xsd:integer
13	"Batman v Superman: Dawn of Justice"	*673**xsd:integer
14	"The Hunger Games"	*673**xsd:integer
15	"Black Swan"	*669**xsd:integer
16	"Beasts of the Southern Wild"	*663**xsd:integer
17	"Star Wars: The Force Awakens"	*656**xsd:integer

Вывод: Освоена работа с базой данных GraphDB в виртуальной машине с использованием Docker. Изучен процесс загрузки RDF-данных в базу, выполнения запросов с использованием SPARQL и анализ результатов, используя функциональность GraphDB.