

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

Кузьмина Дарья Юрьевна БД-241м

Инструменты хранения и анализа больших данных

Лабораторная работа 2.1. Часть 3.
Рекомендательные системы. GraphDB
Вариант 11

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,
доктор экономических наук

Москва
2025

Содержание

Введение	2
Основная часть	2
Заключение	12

Введение

Цель

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.

Процесс начала работы:

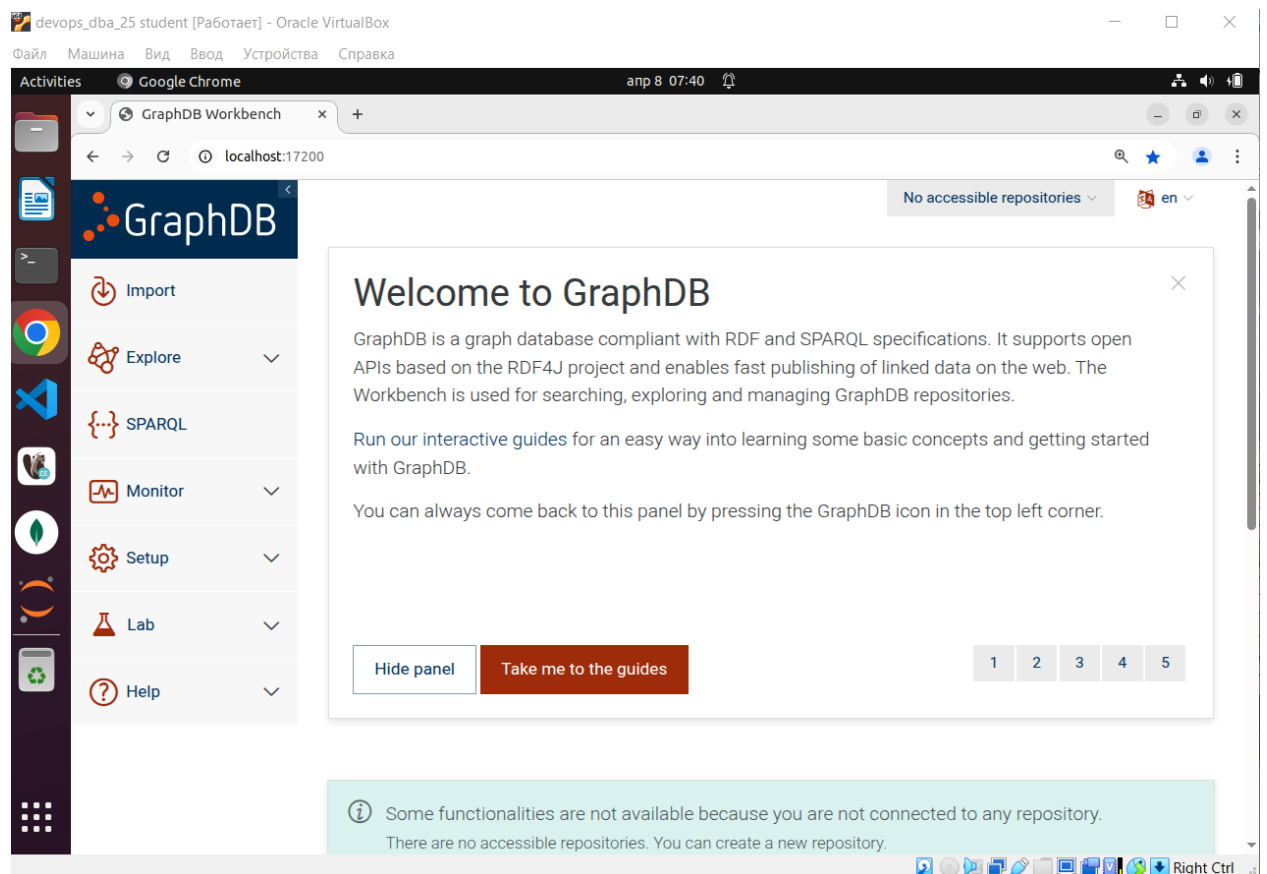
1. Подключитесь к виртуальной машине и войдите в нее.
2. Перейдите в каталог с проектом:
3. `cd ~/Downloads/dba/nonrel/graphdb`
4. Остановите контейнер, если он работает:
5. `sudo docker compose stop`
6. Запустите контейнер с базой данных GraphDB:
7. `sudo docker compose start`

Основная часть

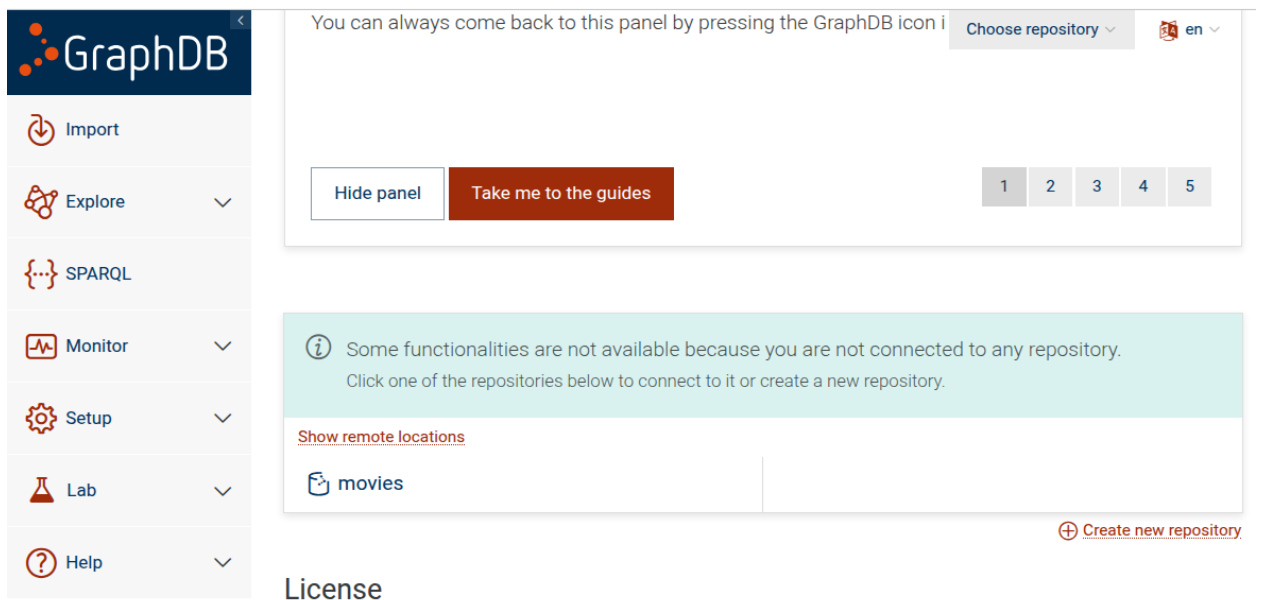
Задача 1.

Запускаем GraphDB

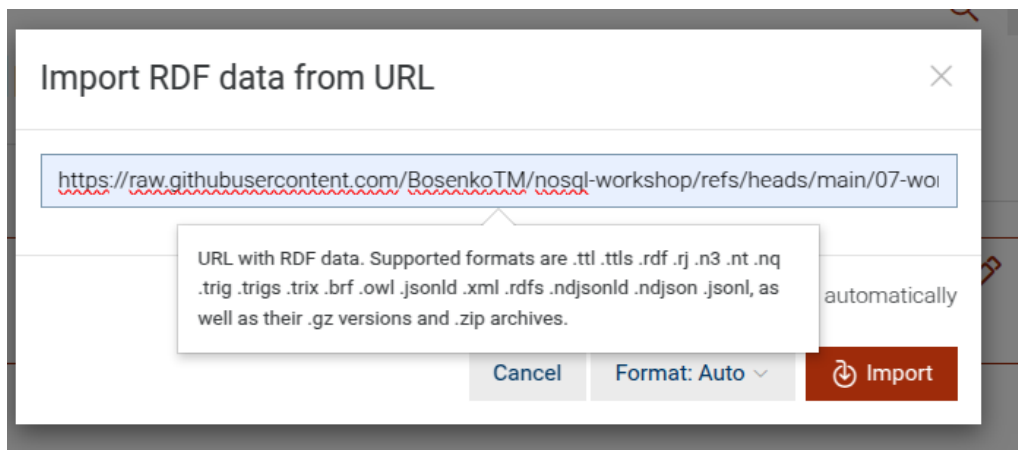
- `dev@dev-vm:~/Downloads/dba/nonrel$ cd graphdb`
- `dev@dev-vm:~/Downloads/dba/nonrel/graphdb$ docker stop $(docker ps -q) b7804ebeb56d a5bb7206afbd ec4919b750da`
- `dev@dev-vm:~/Downloads/dba/nonrel/graphdb$ sudo docker compose up -d`
[sudo] password for dev:
[+] Running 1/1
✓ Container graphdb-1 Started
- `dev@dev-vm:~/Downloads/dba/nonrel/graphdb$`



Раходим в GraphDB



Создаем базу данных



Импортируем данные из источника, тут можно указать свой гит.

После проведения практической части. Я добавила свой датасет на гит:

https://raw.githubusercontent.com/Iezekiss/BDSAD_MGPU/refs/heads/main/Lab2.1/Lab2.1.3/TMDB_all_movies.csv

Кстати, там можно посмотреть рейтинг и дату выхода.

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



Type to filter



<https://raw.githubusercontent.com/BosenkoTM/nosql-workshop/refs/heads/main/07-working-with-graphdb/data/movies.ttl>
X ? Imported successfully in 43s.



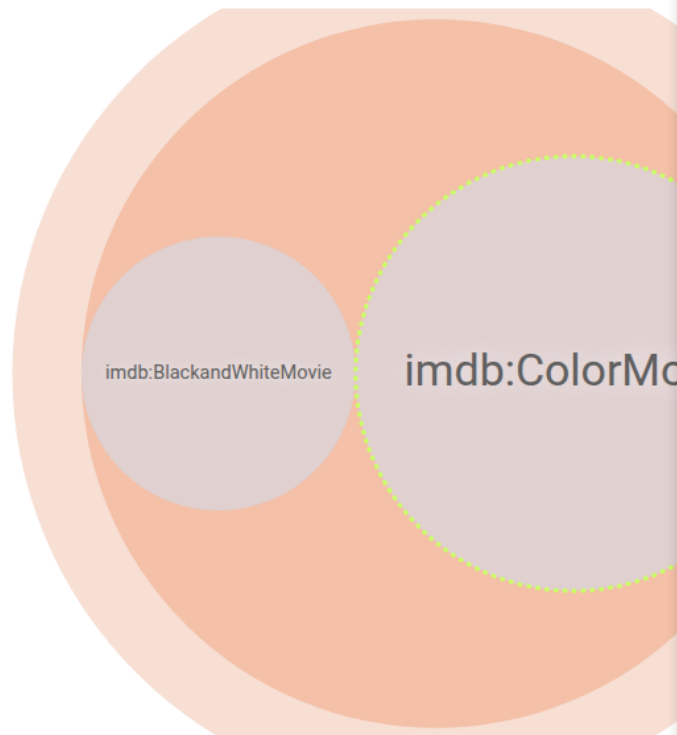
Import

Импортировала

The screenshot shows the GraphDB interface with the 'Class hierarchy' view selected. The left sidebar contains navigation options: Import, Explore, Graphs overview, Class hierarchy (selected), Class relationships, Visual graph, Similarity, SPARQL, Monitor, Setup, and Lab. The main area displays a 'Class Count' of 3 and a nested circle diagram for 'schema:Movie'. The diagram consists of three concentric circles: an outer light orange circle, a middle dark orange circle, and an inner grey circle. The text 'schema:Movie' is centered in the inner circle. The bottom status bar indicates 'GraphDB 10.6.2 • RDF4J 4.3.8 • Connectors 16.2.6 • Workbench 2.6.2 • © 2002–2025 Ontotext AD. All rights reserved.'

Смотрим схемы.

Class hierarchy



movies en

imdb:ColorMovie

Domain-Range graph

[View all 4,690 instances in SPARQL](#)

Search first 1000 class instances

- imdb:title/TheBourneUltimatum
- imdb:title/K19TheWidowmaker
- imdb:title/TrueLies
- imdb:title/Superman
- imdb:title/BenHur
- imdb:title/Mulan
- imdb:title/TheSimpsonsMovie

Смотрим первичные графы.

TheMatrix

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

subject predicate object context

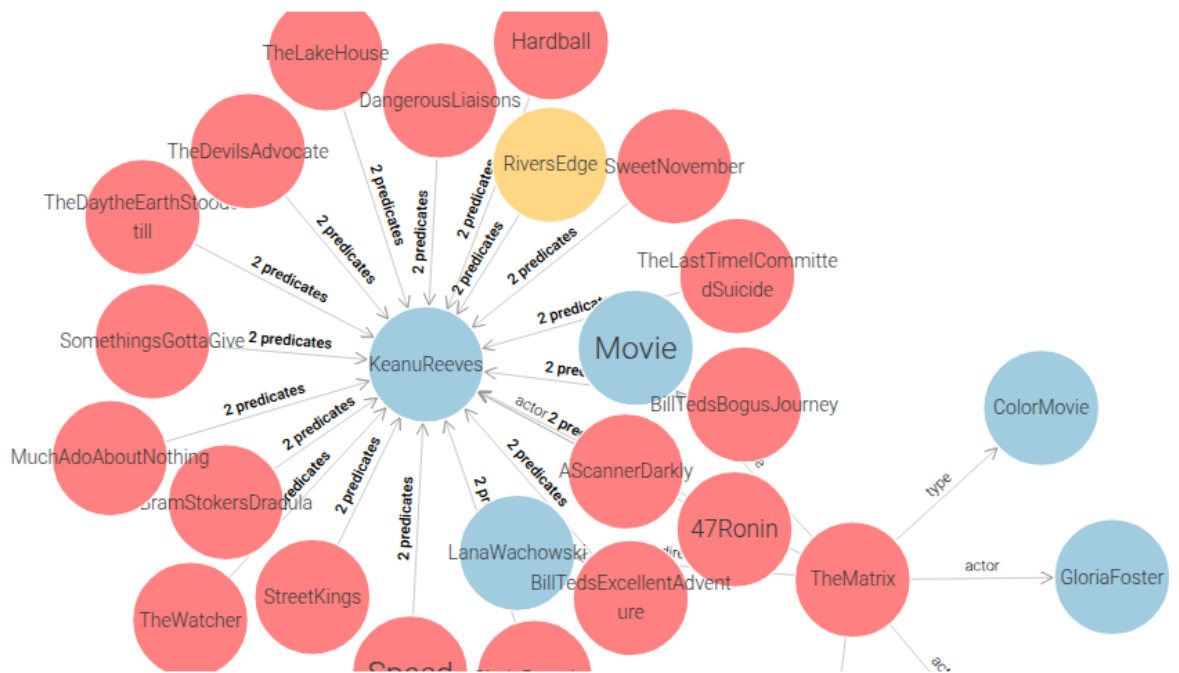
Explicit only Show Blank Nodes Download as Visual graph

all

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/	http://www.ontotext.com/

Посмотрим, какими графами описан объект.



Рассмотрим субъект, объект и предикат.

Переходим в SPARQL.

	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
10	owl:equivalentClass	rdf:type	owl:SymmetricProperty
11	owl:equivalentClass	rdf:type	owl:TransitiveProperty

Выполняем первый запрос, ограничивающий количество графов до 100

Далее мы нашли набор связанных графов из источников из семантической сети и выполнили дополнительные задания на отработку поисковых запросов.

Индивидуальное задание **Вариант 11**

Напишите запрос для получения всех фильмов с их количеством комментариев.	Создайте запрос для поиска всех фильмов с рейтингом выше 9,0.	Найдите все фильмы, в которых сыграл "Keanu Reeves" и режиссером был "Lana Wachowski".	Напишите запрос для поиска всех фильмов, выпущенных в 1990-е годы.	Найдите все фильмы, у которых актер и режиссер — один и тот же человек.
---	---	--	--	---

Задание 1

Получение всех фильмов с количеством комментариев

Что делает запрос:

Находит все фильмы и подсчитывает, сколько комментариев есть у каждого. Сортирует по убыванию количества комментариев.

Пример вывода:

- Фильм "Матрица" — 150 комментариев
- Фильм "Крестный отец" — 120 комментариев


```

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

```

	movie	movieName	commentCount
1	imdb.title/TheDarkKnightRises	"The Dark Knight Rises"	"813"
2	imdb.title/Prometheus	"Prometheus"	"775"
3	imdb.title/DjangoUnchained	"Django Unchained"	"765"
4	imdb.title/Skyfall	"Skyfall"	"750"
5	imdb.title/MadMaxFuryRoad	"Mad Max: Fury Road"	"739"
6	imdb.title/Gravity	"Gravity"	"738"
7	imdb.title/ManofSteel	"Man of Steel"	"733"
8	imdb.title/Avatar	"Avatar"	"723"
9	imdb.title/Interstellar	"Interstellar"	"712"
10	imdb.title/TheAvengers	"The Avengers"	"703"
11	imdb.title/Hugo	"Hugo"	"682"

Задание 3

Фильмы с Киану Ривзом в роли актера и Ланой Вачовски в роли режиссера

Что делает запрос:

Ищет фильмы, где одновременно:

- Актер — "Keanu Reeves"
- Режиссер — "Lana Wachowski"

Пример вывода:

- "Матрица"
- "Матрица: Перезагрузка"

```
1
2 * PREFIX schema: <http://schema.org/>
3
4 SELECT ?movie ?title
5 * WHERE {
6     ?movie a schema:Movie;
7           schema:name ?title;
8           schema:actor [schema:name "Keanu Reeves"];
9           schema:director [schema:name "Lana Wachowski"].
10 }
11
```

Run

keyboard shortcuts



Download as

Filter query results

Compact view ☐ Hide row numbers ☐

Showing results from 0 to 1 of 1. Query took 0.3s, moments ago.

	movie	title
1	imdb:title/TheMatrix	"The Matrix"

Задание 5

Фильмы, где актер и режиссер — один и тот же человек

Что делает запрос:

Ищет фильмы, в которых один человек был и **актером**, и **режиссером**.

Показывает название фильма и имя этого человека.

Пример вывода:

- "Гражданин Кейн" — Орсон Уэллс

- "Одержимость" — Клинт Иствуд

```
Unnamed × Unnamed 1 × Unnamed 2 × Unnamed 3 × Unnamed 4 ×
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
```

Run

keyboard shortcuts

	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
9	imdb:person/WarrenBeatty	"Warren Beatty"	"1"^^xsd:integer
10	imdb:person/JacquesPerrin	"Jacques Perrin"	"1"^^xsd:integer
11	imdb:person/SylvesterStallone	"Sylvester Stallone"	"1"^^xsd:integer
12	imdb:person/KevinSpacey	"Kevin Spacey"	"1"^^xsd:integer

Получить данные по заданию 2 и заданию 4 невозможно, поскольку в моем файле нету данных о рейтинге и дате выпуска фильма, однако можно, например, отсортировать по стране выпуска, цветной или не цветной и другие варианты, однако предполагаемые запросы выглядели бы так:

2. Поиск фильмов с рейтингом выше 9.0

Что делает запрос:

Выбирает фильмы, у которых рейтинг больше 9.0, и показывает их названия и сам рейтинг.

Пример вывода:

- "Побег из Шоушенка" — 9.3
- "Крестный отец" — 9.2

```

1
2 * PREFIX schema: <http://schema.org/>
3
4 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5 SELECT ?movie ?title ?rating
6 WHERE {
7   ?movie a schema:Movie;
8           schema:name ?title;
9           schema:ratingValue ?rating.
10  FILTER (xsd:decimal(?rating) > 9.0)
11 }
12

```

Run

keyboard shortcuts

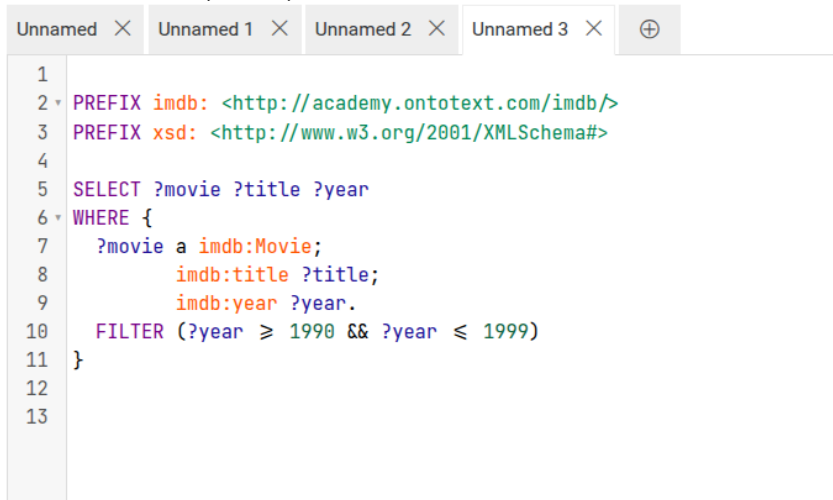
4. Фильмы, выпущенные в 1990-е годы

Что делает запрос:

Находит все фильмы, которые вышли с 1990 по 1999 год, и показывает их названия и год выпуска.

Пример вывода:

- "Криминальное чтиво" (1994)
- "Титаник" (1997)



```
1
2 PREFIX imdb: <http://academy.ontotext.com/imdb/>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4
5 SELECT ?movie ?title ?year
6 WHERE {
7   ?movie a imdb:Movie;
8         imdb:title ?title;
9         imdb:year ?year.
10  FILTER (?year ≥ 1990 && ?year ≤ 1999)
11 }
12
13
```

Заключение

Вывод:

Освоена работа с GraphDB: загрузка данных, настройка репозитория.
Реализованы запросы на SPARQL для анализа связей между сущностями.
Получены навыки фильтрации, агрегации и работы с типами данных.