**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика, искусственный интеллект и системы управления КАФЕДРА Системы обработки информации и управления

**Лабораторная работа №4**

**«Создание графовой базы данных и работа с ней на примере СУБДNeo4j»**

ИСПОЛНИТЕЛИ: Гаврилов Л.Я.

ФИО

группа ИУ5-24М

подпись

"" март 2024 г.

ПРЕПОДАВАТЕЛЬ: Виноградова М.В.

ФИО

подпись

" " 2024 г.

Москва – 2024

1. **Цель работы**

* Изучить модель представления данных и способы работы с графовыми БД;
* Освоить методы создания графовой БД и языки запросов к ней;
* Получить навыки работы с графовой СУБД Neo4j

1. **Задание**

1. Создать в среде Neo4j базу данных по теме «Подбор фильмов на основе глубокого обучения». Определить набор узлов, задать их свойства и метки.

2. Продемонстрировать (вывести на экран) содержимое БД (узлы и их свойства),

используя команды Match/Where/Return.

3. Создать набор узлов БД. Определить для них несколько меток и свойств. Создать

отношение между новыми узлами. Создать отношение между существующими узлами.

4. Продемонстрировать удаление узлов и связей. Продемонстрировать удаление и

изменение свойств и меток.

5. Продемонстрировать содержимое БД (фильтрация по узлам, отношениям, меткам и

связям).

6. Выполнить запросы к базе данных на языке Cypher:

* с условием NOT NULL
* операторами AND, OR
* с сортировкой
* с условием на направление отношения
* с параметрами отношения

<https://stackoverflow.com/questions/23310114/how-to-reset-clear-delete-neo4j-database>

**Шаг 1: Создание базы данных и определение набора узлов**

Предположим, что нам нужны узлы следующих типов:

* Movie (фильм)
* Genre (жанр)
* Director (режиссёр)
* Actor (актёр)
* User (пользователь)

Каждый узел будет иметь следующие свойства:

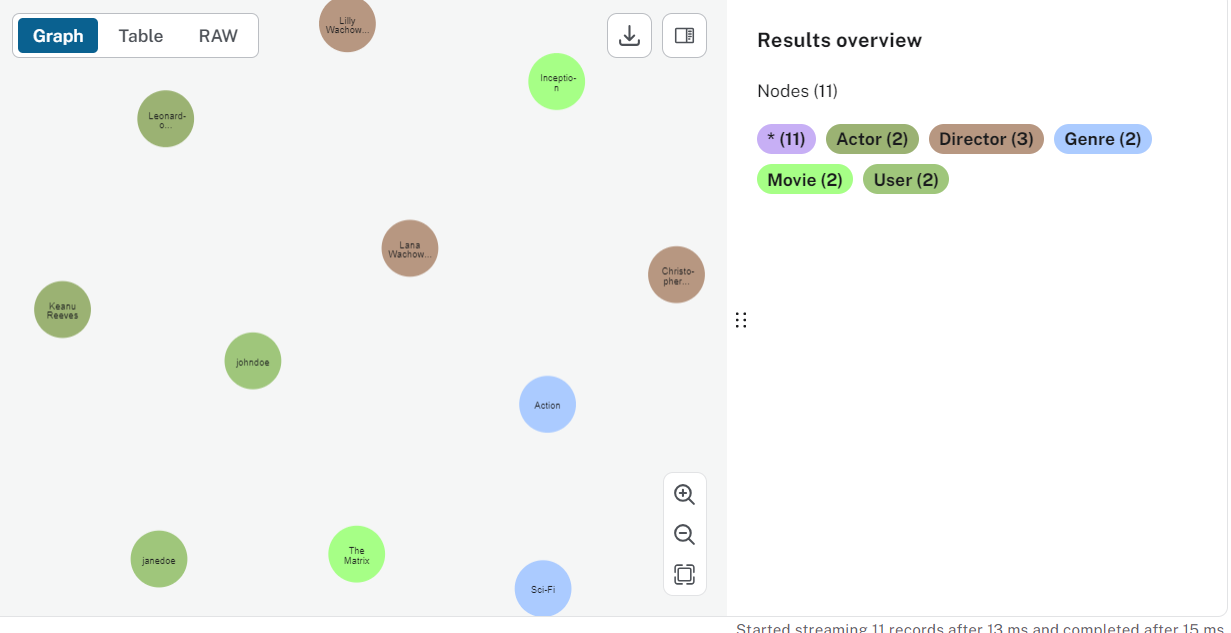
* Movie:
  + title (название)
  + release\_year (год выпуска)
  + rating (рейтинг)
  + description (описание)
* Genre:
  + name (название)
* Director:
  + name (имя)
* Actor:
  + name (имя)
* User:
  + username (имя пользователя)
  + email (электронная почта)

**Шаг 2: Вывод содержимого БД (узлы и их свойства)**

// Вывод всех узлов и их свойств

MATCH (n)

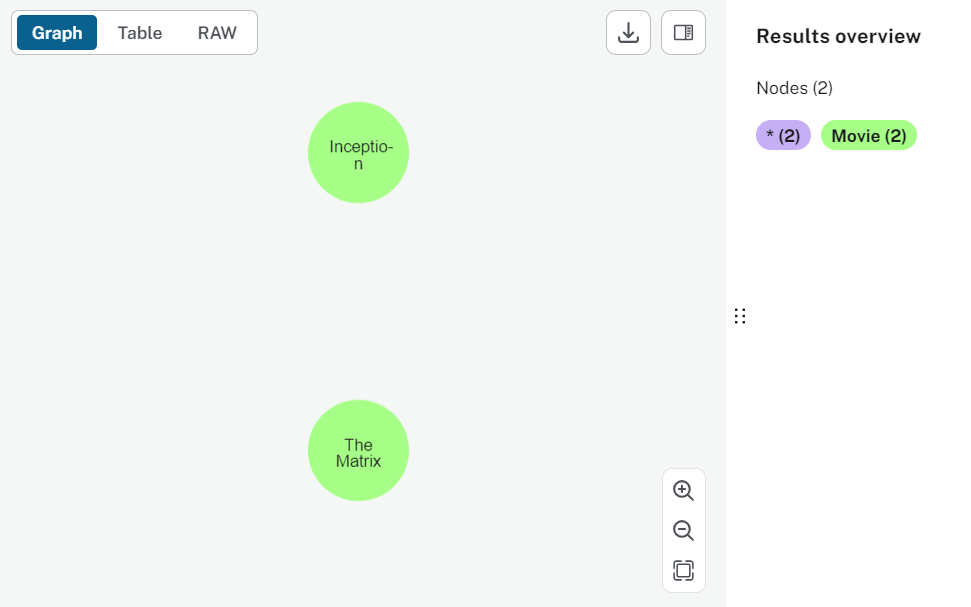
RETURN n;



// Вывод всех узлов типа Movie и их свойств

MATCH (m:Movie)

RETURN m;

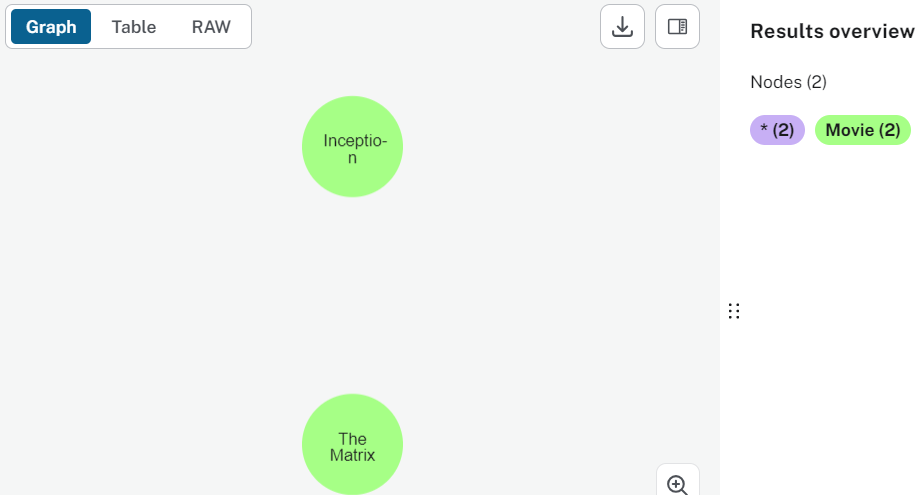


// Вывод всех фильмов с рейтингом выше 8

MATCH (m:Movie)

WHERE m.rating > 8

RETURN m;

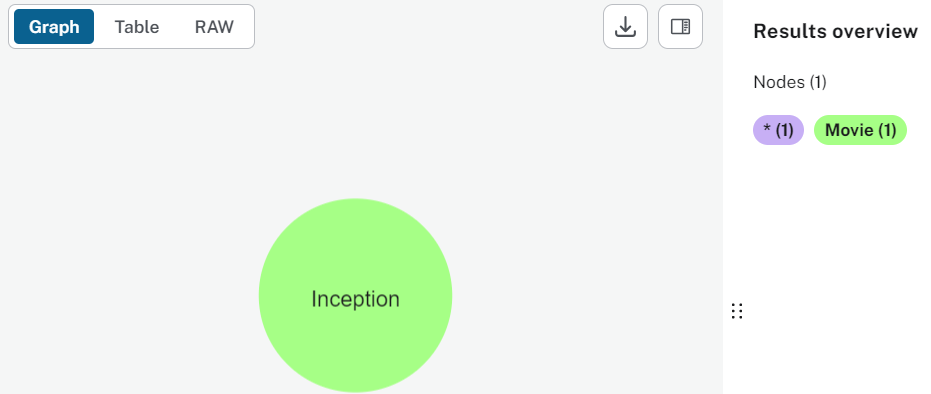


// Вывод всех фильмов жанра Sci-Fi, выпущенных в 2010 году

MATCH (m:Movie)-[:BELONGS\_TO]->(g:Genre)

WHERE g.name = 'Sci-Fi' AND m.release\_year = 2010

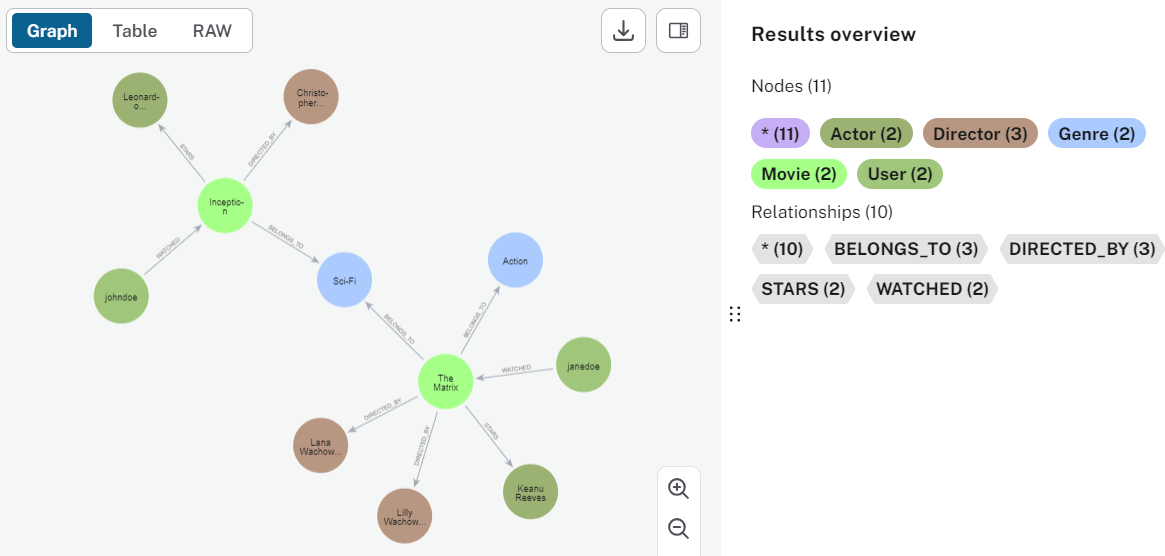
RETURN m;



// Вывод всех узлов и их отношений

MATCH (n)-[r]->(m)

RETURN n, r, m;



**Шаг 3: Создание набора узлов и отношений**

**Создание узлов:**

// Создание узлов типа Movie

CREATE (m:Movie {title: 'Inception', release\_year: 2010, rating: 8.8, description: 'A thief who steals corporate secrets through the use of dream-sharing technology is given the inverse task of planting an idea into the mind of a CEO.'});

CREATE (m:Movie {title: 'The Matrix', release\_year: 1999, rating: 8.7, description: 'A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.'});

// Создание узлов типа Genre

CREATE (g:Genre {name: 'Sci-Fi'});

CREATE (g:Genre {name: 'Action'});

// Создание узлов типа Director

CREATE (d:Director {name: 'Christopher Nolan'});

CREATE (d:Director {name: 'Lana Wachowski'});

CREATE (d:Director {name: 'Lilly Wachowski'});

// Создание узлов типа Actor

CREATE (a:Actor {name: 'Leonardo DiCaprio'});

CREATE (a:Actor {name: 'Keanu Reeves'});

// Создание узлов типа User

CREATE (u:User {username: 'johndoe', email: 'johndoe@example.com'});

CREATE (u:User {username: 'janedoe', email: 'janedoe@example.com'});

**Создание отношений между узлами:**

// Связь между фильмами и жанрами

MATCH (m:Movie {title: 'Inception'}), (g:Genre {name: 'Sci-Fi'})

CREATE (m)-[:BELONGS\_TO]->(g);

MATCH (m:Movie {title: 'The Matrix'}), (g:Genre {name: 'Sci-Fi'})

CREATE (m)-[:BELONGS\_TO]->(g);

MATCH (m:Movie {title: 'The Matrix'}), (g:Genre {name: 'Action'})

CREATE (m)-[:BELONGS\_TO]->(g);

// Связь между фильмами и режиссёрами

MATCH (m:Movie {title: 'Inception'}), (d:Director {name: 'Christopher Nolan'})

CREATE (m)-[:DIRECTED\_BY]->(d);

MATCH (m:Movie {title: 'The Matrix'}), (d:Director {name: 'Lana Wachowski'})

CREATE (m)-[:DIRECTED\_BY]->(d);

MATCH (m:Movie {title: 'The Matrix'}), (d:Director {name: 'Lilly Wachowski'})

CREATE (m)-[:DIRECTED\_BY]->(d);

// Связь между фильмами и актёрами

MATCH (m:Movie {title: 'Inception'}), (a:Actor {name: 'Leonardo DiCaprio'})

CREATE (m)-[:STARS]->(a);

MATCH (m:Movie {title: 'The Matrix'}), (a:Actor {name: 'Keanu Reeves'})

CREATE (m)-[:STARS]->(a);

// Связь между пользователями и фильмами (например, просмотренные фильмы)

MATCH (u:User {username: 'johndoe'}), (m:Movie {title: 'Inception'})

CREATE (u)-[:WATCHED]->(m);

MATCH (u:User {username: 'janedoe'}), (m:Movie {title: 'The Matrix'})

CREATE (u)-[:WATCHED]->(m);

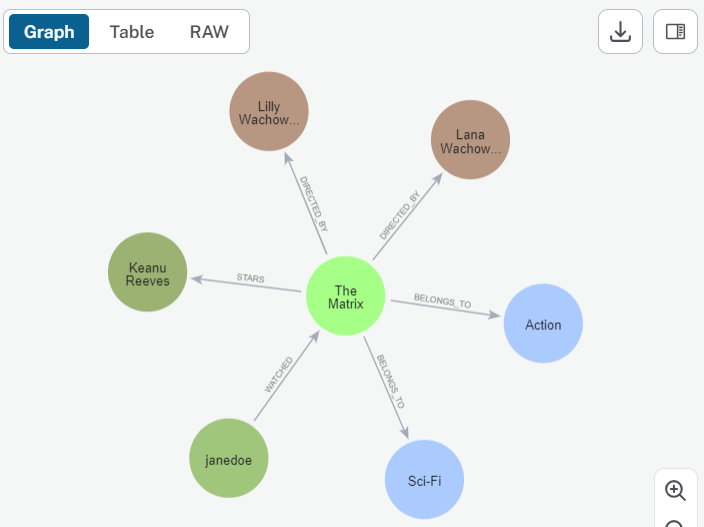
**Шаг 4: Удаление узлов и связей, изменение свойств и меток**

**Удаление узлов и связей:**

// Удаление узла и всех его связей

MATCH (m:Movie {title: 'Inception'})

DETACH DELETE (m);



// Удаление определенной связи

MATCH (u:User {username: 'johndoe'})-[r:WATCHED]->(m:Movie {title: 'The Matrix'})

DELETE r;

**Изменение свойств и меток:**

// Изменение свойства узла

MATCH (m:Movie {title: 'The Matrix'})

SET m.rating = 9.0;

// Добавление метки к узлу

MATCH (m:Movie {title: 'The Matrix'})

SET m:Classic;

// Удаление метки с узла

MATCH (m:Movie {title: 'The Matrix'})

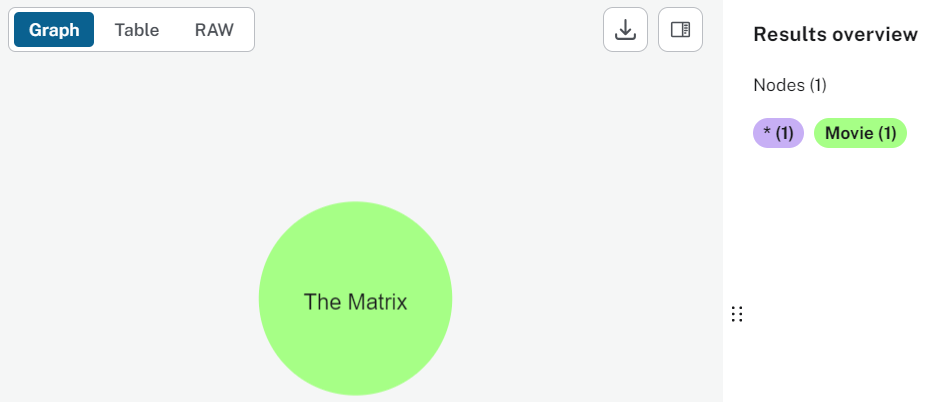
REMOVE m:Classic;

**Шаг 5: Фильтрация по узлам, отношениям, меткам и связям**

// Вывод всех фильмов

MATCH (m:Movie)

RETURN m;



// Фильтрация по метке

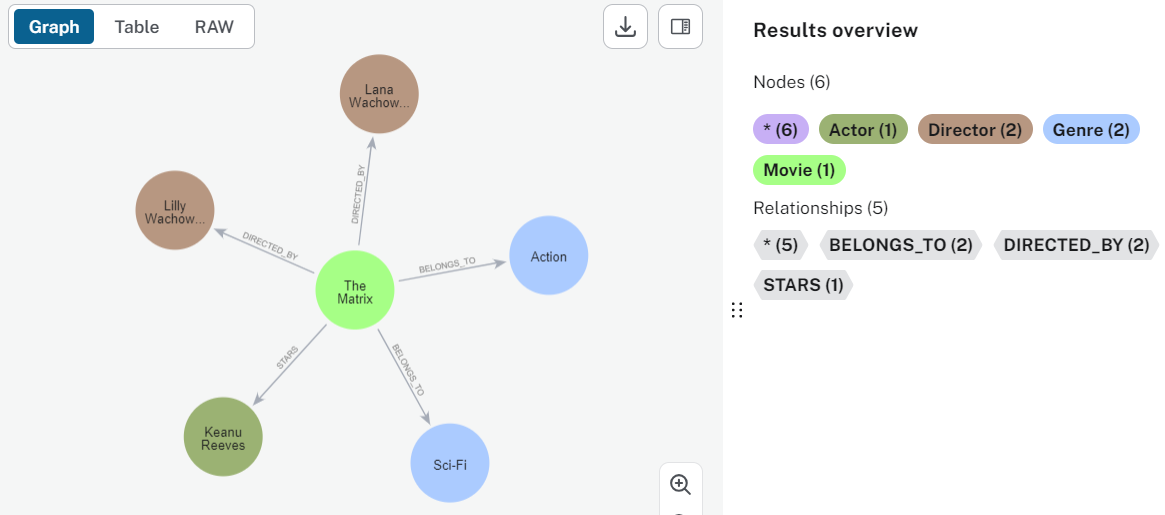
MATCH (m:Movie:Classic)

RETURN m;

// Вывод всех отношений для фильма 'The Matrix'

MATCH (m:Movie {title: 'The Matrix'})-[r]->(n)

RETURN m, r, n;



**Шаг 6: Запросы с условиями NOT NULL, операторами AND, OR, сортировкой и направлением отношений**

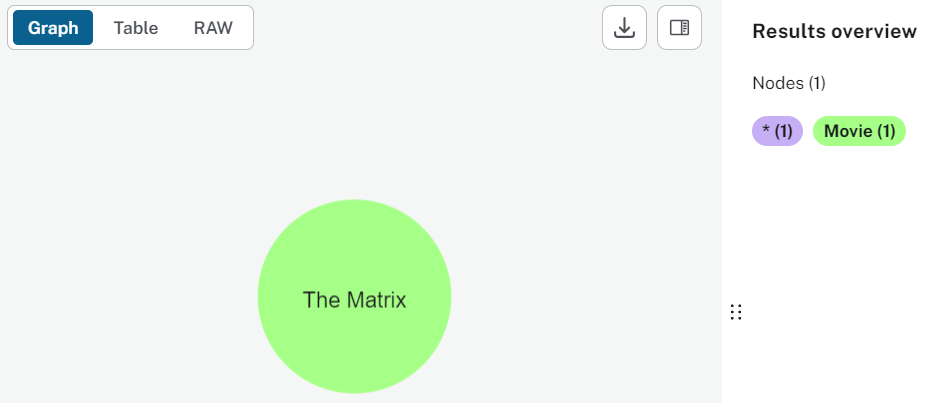
**Условие NOT NULL:**

// Фильмы с ненулевым рейтингом

MATCH (m:Movie)

WHERE m.rating IS NOT NULL

RETURN m;



**Операторы AND, OR:**

// Фильмы определенного года и с определенным рейтингом

MATCH (m:Movie)

WHERE m.release\_year = 2010 AND m.rating > 8

RETURN m;

// Фильмы определенного жанра или режиссёра

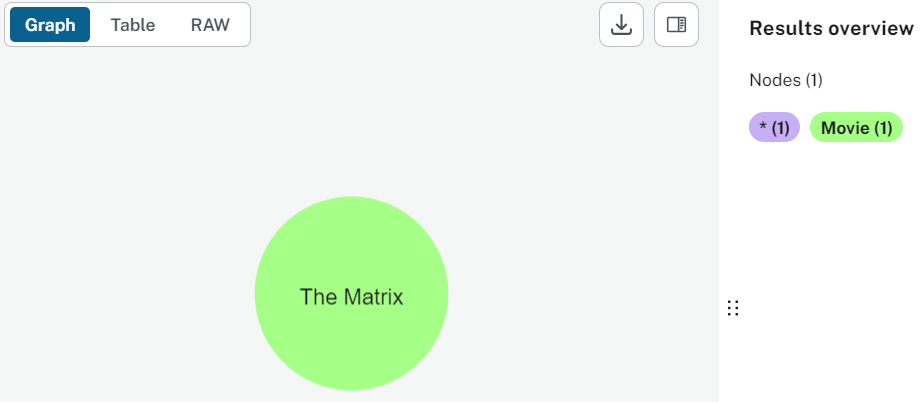
MATCH (m:Movie)-[:BELONGS\_TO]->(g:Genre {name: 'Sci-Fi'})

RETURN m

UNION

MATCH (m:Movie)-[:DIRECTED\_BY]->(d:Director {name: 'Christopher Nolan'})

RETURN m;



**Сортировка:**

// Сортировка фильмов по рейтингу

MATCH (m:Movie)

RETURN m

ORDER BY m.rating DESC;

**Условие на направление отношения:**

// Фильмы, снятые определенным режиссёром

MATCH (d:Director {name: 'Christopher Nolan'})<-[:DIRECTED\_BY]-(m:Movie)

RETURN m;

**Условия на параметры отношения:**

// Пользователи, просмотревшие фильмы с рейтингом выше 8

MATCH (u:User)-[:WATCHED]->(m:Movie)

WHERE m.rating > 8

RETURN u, m;

