Основы теории систем

**Отчет по лабораторной №3**

«Формирование онтологической модели с помощью инструментов  
онтологического инжиниринга Protege и Neo4g»

Вариант 18

Выполнила:

Студент 1 курса   
Группы ИВТИИбд-12  
Романенко И.И.

Ульяновск,  
2025

Задание на лабораторную работу:

* Сформировать онтологическую модель структуры автомобиля в инструментах онтологического инжиринга Protege и Neo4j
* Провести сравнительный анализ данных инструментов
* Сделать заключение какой же инструмент является более удобным для разработки онтологических моделей

1. Разработка онтологической модели

Нужно разработать онтологическую модель структуры машины.Требования:

* Минимум 3 класса
* у каждого классса хотя бы 1 свойство
* минимум 2 типа связи
* минимум 2 экземпляра каждого класса
* минимум 2 правила логического вывода

Классы:

* Эпоха
* Страна
* Личность
* Событие
* Время

Отношения:

* Произошло в
* Принадлежит эпохе
* Родился в
* Участвовал в

Аксиомы:

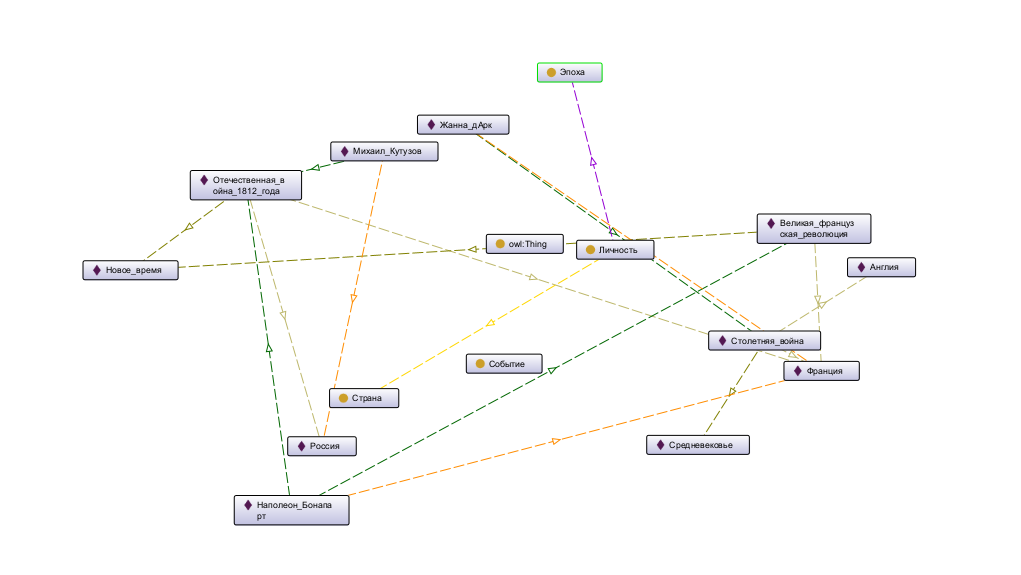
* Если Класс имеет отношение «Произошло в» с классами «Страна», то этот класс является классом «Событие»
* Если (Х произошло в Y) и (Y произошло в Z) -> (X произошло в Z)

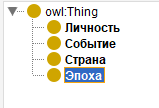
1.1 Protege

Создание классов

В данной онтологической модели существует 1 класс «Время» и его 4 подкласса: «Эпоха», «Страна», «Личность», «Событие».

Рисунок 1 – Скриншот модели.



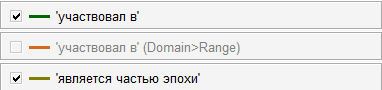
Рисунок 2 – Скриншот классов.

Создание отношений  
В Protege связи между элементами онтологии определяются с помощью отношений (properties):

Рисунок 3 — Скриншот связей







Особенности

* Наличие Reasoner – есть возможность создавать логические выводы
* На основе правил и ограничений самостоятельно создает новые факты

Плюсы

* Удобный интерфейс
* Наличие резонера – вывод данных на основе правил и ограничений в модели
* Наличие плагинов – при желании можно скачать плагины, например для визуализации классов
* Удобно проектировать онтологические модели

Минусы

* Не поддерживает работу с большим обьемом данных – это не база данных

1.2 Neo4j

Neo4j — графовая система управления базами данных (СУБД).Использует собственный язык запросов – Cypher.



Рисунок 4 –

Скриншот запроса создания нового узла

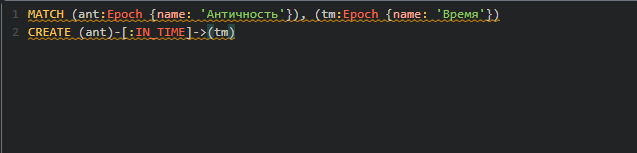


Рисунок 5 – Скриншот запроса создания отношений между узлами

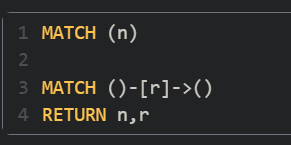
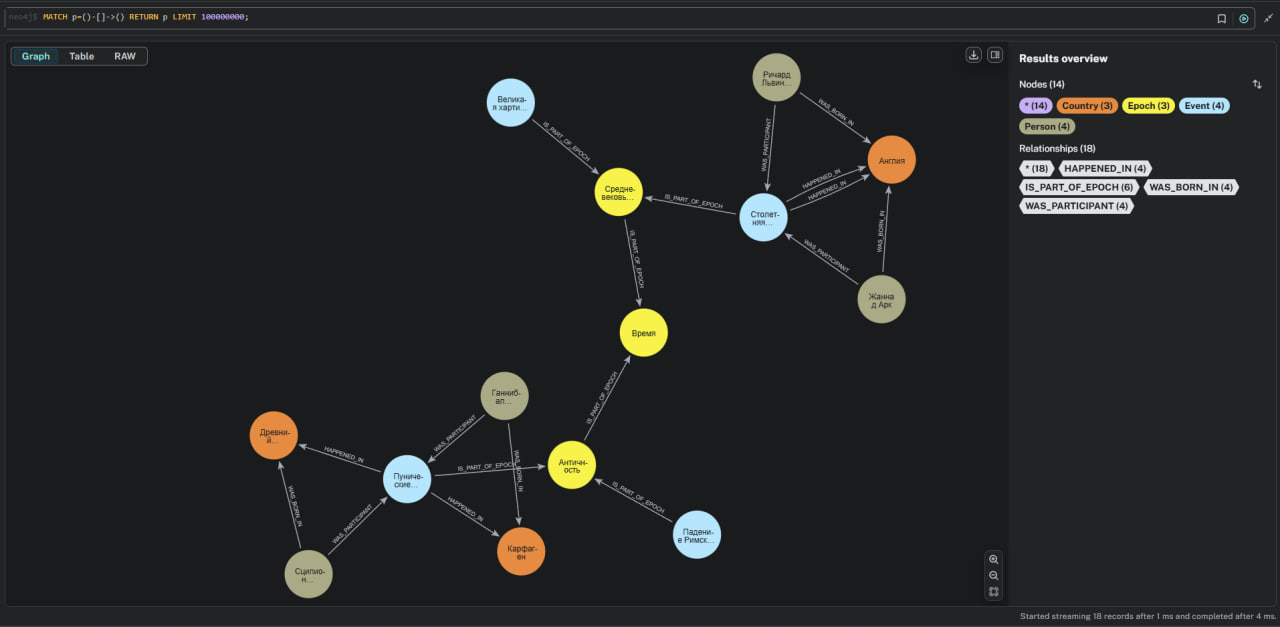


Рисунок 6 – Скриншот запроса вывода всех узлов и связей



Скриншот 7 – Скриншот графа заданной онтологической модели

Особенности

* В Neo4j нет такого понятия как «Классы», но есть Label – метки,группирующие узлы по типу, но даже они не задают структуру данных строго, так как узлы с одной и той же меткой могут иметь разные свойства
* Есть свой язык запросов – Chypter
* Гибкая система,можно добавлять узлы и связи динамически

Плюсы

* Рассчитан на работу с большим объемом данных
* Можно интегрировать в приложение, так как является Базой данных

Минусы

* требуются минимальные знания языка запросов Chypter
* Отсутствие Reasoner – нельзя делать логические выводы

1.3 Вывод

Protege – лучший вариант, если:

* нужно разработать/отредактировать онтологическую модель
* когда предпочтителен удобный интерфейс без глубоких знаний языков программирования

Neo4g используется в случае, если:

* требуется работа с большим объемом данных,структурированных как граф
* нужно хранить данные динамически и без жесткой схемы
* нужна интеграция в приложениями на разных языках программирования