**Санкт-Петербургский государственный электротехнический университет**

**«ЛЭТИ» им. В.И.Ульянова (Ленина)**

**(СПбГЭТУ «ЛЭТИ»)**

|  |  |  |
| --- | --- | --- |
| **Направление** | 09.03.02– Информационные системы и технологии | |
| **Профиль** | Информационные системы и технологии | |
| **Факультет** | КТИ | |
| **Кафедра** | ИС | |
| *К защите допустить* |  | |
| Зав. кафедрой |  | Цехановский В.В. |

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

БАКАЛАВРА

Тема: **Разработка подсистемы поиска тематических данных в графовой базе данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент(ка) |  |  |  | Порошин М.А. |
|  |  | *подпись* |  |  |
| Руководитель |  |  |  | Зорин К.М. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
| Консультанты |  |  |  | Иванов И.И. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
|  |  |  |  | Иванов И.И. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
|  |  |  |  | Иванов И.И. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |

Санкт-Петербург

2017**ЗАДАНИЕ**

**на выпускную квалификационную работу**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой ИС |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Цехановский В.В. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Студент: | Порошин М.А. | | | |  | Группа | 3372 |
| Тема работы: Разработка подсистемы поиска тематических данных в графовой базе данных. | | | | | | | |
| Место выполнения ВКР: СПбГЭТУ «ЛЭТИ» | | | | | | | |
| Исходные данные (технические требования):  Разработка прототипа подсистемы, способной находить информацию в графовой базе данных по заданным ключам и представлять ее в понятном пользователю виде | | | | | | | |
| Содержание ВКР:  Создание прототипа модуля анализа информации о событиях, разделѐнное на исследование вопроса, разработку системы сбора информации и анализ данных через работу с графовой базой. | | | | | | | |
| Перечень отчетных материалов: текст ВКР, иллюстративный материал | | | | | | | |
| Дополнительные разделы: технико-экономическое обоснование проекта. | | | | | | | |
|  | | | | | | | |
| Дата выдачи задания | | | Дата представления ВКР к защите | | | | |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | | | |
|  | | |  | | | | |
| Студент | |  | | Порошин М.А. | | | |
| Руководитель | |  | | Зорин К.М. | | | |

**календарный план выполнения**

**выпускной квалификационной работы**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой ИС |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Цехановский В.В. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент: | Порошин М.А. |  | Группа | 3372 |
| Тема работы: Разработка подсистемы поиска тематических данных в графовой базе данных. | | | | |

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование работ | Срок выполнения |
| 1 | Обзор литературы по теме работы | 00.00 – 00.00 |
| 2 | Наименование раздела | 00.00 – 00.00 |
| 3 | Наименование раздела | 00.00 – 00.00 |
| 4 | Наименование раздела | 00.00 – 00.00 |
| 5 | Оформление пояснительной записки | 00.00 – 00.00 |
| 6 | Оформление иллюстративного материала |  |

|  |  |  |
| --- | --- | --- |
| Студент: |  | Порошин М.А. |
| Руководитель |  | Зорин К.М. |

*(Уч. степень, уч. звание)***РЕФЕРАТ**

Пояснительная записка 00 стр., 00 рис., 00 табл., 00 ист., 00 прил.

Объектом разработки является система поиска тематических данных в графовой базе данных.

Цель работы – поиск в графовой базе событий по определенным ключам с целью ответа на вопрос пользователя и вывод ответа на человеческом языке.

Одним из основных источников хранения информации служат базы данных. Однако не все умеют писать к ним запросы. Более общим и удобным способом для получения информации была бы возможность задавать вопросы компьютеру на естественном языке. Такую задачу решают вопрос-ответные системы. В данной работе описана разработка прототипа модуля поиска данных по определенным ключам в графовой бд для системы, принимающей вопрос и дающей ответ на естественном языке. Задачей модуля является конструирование запросов к базе и интерпретация ответа. В качестве данных были выбраны события, для хранения данных была использована база neo4j, естественным языком выбран английский. Несмотря на то, что на некоторых шагах были получены неполные результаты, в целом рабочий прототип задуманной системы удалось создать.

**ABSTRACT**

Briefly (10-15 lines) the content of graduating work is specified

**содержание**

|  |  |  |
| --- | --- | --- |
|  | Введение | 8 |
| 1 | Современное состояние вопроса | 9 |
| 1.1. | История вопроса | 10 |
| 1.2. | Современные проблемы | 15 |
| 1.3. | Пути решения проблем | 20 |
| 2 | Описание исследования (разработки) | 25 |
| 2.1. | Принципы, материалы и методы | 30 |
| 2.2. | Теоретические исследования и расчеты | 35 |
| 2.3. | Модель, чертежи, электрические схемы | 40 |
| 3 | Результаты исследования (разработки) | 45 |
| 3.1. | Методика эксперимента | 50 |
| 3.2. | Результаты эксперимента | 55 |
| 3.3. | Сравнение и оценка результатов | 60 |
| 4 | Название дополнительного раздела | 65 |
| 4.1. | Название 1-го подраздела дополнительного раздела | 66 |
| 4.2. | Название 2-го подраздела дополнительного раздела | 66 |
| 4.3. | Название 3-го подраздела дополнительного раздела | 67 |
|  | Заключение | 68 |
|  | Список использованных источников | 69 |
|  | Приложение А. Название приложения | 72 |

*Наименования разделов приведены в качестве примера.*

**определения, обозначения и сокращения**

В настоящей пояснительной записке применяют следующие термины с соответствующими определениями:

UML – унифицированный язык моделирования.

Neo4j – реализованная на языке java система управления графовыми базами данных.

SQL – формальный непроцедурный язык программирования, применяемый для работы с данными в реляционных базах данных.

NoSQL – «не только SQL», ряд подходов, направленных на реализацию хранилищ баз данных, имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL.

Cypher – декларативный NOSQL язык для работы с данными в графовых базах данных.

QA – (Question answering system) информационная система c естественно языковым интерфейсом, способная принимать вопросы и отвечать на них на естественном языке.

**Введение**

С давних пор информация играет важную роль в нашей жизни; нам нужно иметь определенную информацию, перед тем как принять решение. Недавно с развитием технологий, таких как компьютеры, гаджеты и интернет, у нас появилась возможность получать практически где угодно и какую угодно информацию.

Одним из главных источников информации служат базы данных. Они содержат набор связанных данных. Для того чтобы извлечь информацию из БД нужно сформулировать запрос понятный для компьютера. Однако не все умеют писать такие запросы. Более общим и удобным способом для получения информации была бы возможность задавать вопросы компьютеру на естественном языке, однако, такие вопросы для компьютера будут всего лишь последовательностью символов. Поэтому в настоящее время создаются различные системы, облегчающие поиск информации, дабы восполнить этот пробел.

На данный момент существуют вопросно – ответные или Question Answering (QA) системы, способные принимать вопросы и отвечать на них на естественном языке. Она позволяет пользователю задавать фактические вопросы на естественном языке (Где находится лувр?) и получить точный конкретный ответ (В Париже, Франция), в отличие от варианта с поисковиком, который выведет страницу со ссылками, среди которых пользователь будет вынужден сам искать необходимую информацию. В этом отношении QA можно рассматривать как следующее поколение удобных инструментов для поиска в большом количестве текстовых данных, например в сети интернет. Кроме того QA системы и их отдельные модули могут быть использованы для создания ботов для социальных сетей или интерфейсы для управления устройствами или например умным домом.

Среди известных систем можно выделить IBM Watson – суперкомпьютер, оснащённый вопросно-ответной системой искусственного интеллекта или Siri (Speech Interpretation and Recognition Interface) —персональный помощник и вопросно-ответная система, программный клиент которой входит в состав програмных продуктов компании Apple. Из отечественных аналогов стоит отметить систему «Нигма.РФ», первую в русском интернете интеллектуальную метапоисковую система, созданую при поддержке факультетов ВМК и психологии МГУ, а также Стэнфордского университета.

В данном, конкретном случае реализации, для упрощения задачи, система была разбита на 3 модуля: модуль извлечения информации из источников (статей), модуль естественно языкового интерфейса и модуль поиска тематических данных в графовой бд. В работе будет подробно рассмотрен модуль поиска тематических данных и графовые базы данных.

**Глава 1. СОВРЕМЕННОЕ СОСТОЯНИЕ ВОПРОСА**

* 1. **История вопроса**

С давних пор информация играет важную роль в нашей жизни; нам нужно иметь определенную информацию, перед тем как принять решение. Недавно с развитием технологий, таких как компьютеры, гаджеты и интернет, у нас появилась возможность получать практически какую угодно информацию. Одним из главных источников информации служат базы данных. Для того чтобы извлечь информацию из БД нужно сформулировать запрос понятный для компьютера. Однако не все умеют писать такие запросы. Более общим и удобным способом для получения информации была бы возможность задавать вопросы компьютеру на естественном языке, однако, такие вопросы для компьютера будут всего лишь последовательностью символов. Поэтому системы, облегчающие поиск информации, с возможностью разбора «человеческого» языка создавались и совершенствуются по сей день.

Естественный язык к базам данных NLIDB (Natural Language Interface to DataBases) представляют собой системы, которые переводят запросы с человеческого языка в запрос понятный БД. NLIDB может рассматриваться как классическая задача в области обработки естественного языка. Несмотря на то, что самое раннее исследование началось с конца шестидесятых годов, исследования продолжаются, ввиду того что пока не удалось создать систему полностью на естественном языке, интуитивно понятную даже для пользователя, далекого от таких вещей. Создание полноценной NLIDB несет за собой множество полезных следствий, таких например как иное восприятие пользователями баз данных, удобства работы с ними, а следовательно их используемость в целом.

Также существуют вопросно-ответные системы Question Answering (QA), способные принимать вопросы и отвечать на них на естественном языке. Вопросно- ответная система многое переняла от NLIDB и является более современной. Она позволяет пользователю задавать фактические вопросы на естественном языке (Where is the Louvre Museum located?) и получить точный конкретный ответ (In Paris, France), в отличие от варианта с поисковиком, который выведет страницу со ссылками, среди которых пользователь будет вынужден сам искать необходимую информацию. В этом отношении QA можно рассматривать как следующее поколение удобных инструментов для поиска в больших объемах текстовых данных, например в сети интернет. В общем случае представляет собой информационную систему, содержащую в себе комплекс интеллектуальных систем и набор справочный материалов. Вопрос-ответные системы могут быть как узкоспециализированными так и общего назначения, старающиеся ответить на любой вопрос пользователя.

Для системы, работающей с большими объемами произвольно связанных данных необходимо их грамотно хранить и иметь механизмы для поиска, обладающие большой производительностью. Большинство подобных систем было реализовано на реляционных базах данных. С появлением графовой модели данных и графовых субд стали появляться аналоги, также использующие графы для поиска информации или машинного обучения через диалог с пользователем. На мой взгляд, из вариантов решения данной задачи может стать использование графовых баз данных.

* 1. **Современные проблемы**

Современные QA-системы построены на принципах NLP (Natural Language Processing), что позволяет им образовывать классы интеллектуальных систем информационного поиска. Системы имеют как общие принципы обработки естественного языка, так и различия в способе внутреннего представления семантического содержимого и в алгоритмах для приведения текста на естественном языке к внутреннему представлению системы. Следовательно, результаты работы системы напрямую зависят от реализуемых алгоритмов компьютерной лингвистики. Кроме того, для большинства национальных языков таковая задача установки семантических ролей не поставлена в отличие от английского, где подготовлена существенная база аннотированных текстов. Этим обусловлен выбор английского в качестве естественного языка при разработке прототипа.

Такой системе необходимо хранить в базе данных множество объектов с нефиксированным количеством полей и различными связями между ними. Важно чтобы модель была гибкой. А значит, возможность добавлять большие объемы объектов (возможно с отличным набором полей и с новыми видами отношений) за короткое время и без вреда для самой модели. Также немаловажно, чтобы система имела высокую производительность, т.к. осуществляется работа с большими объемами данных.

* 1. **Пути решения проблем**

Вариантом решения может стать использование графовых баз данных. Графовая база представляет собой разновидность бд с реализацией сетевой модели в виде графа и его обобщений. Компоненты базы – узлы и ребра, которые могут быть дополнены набором полей. Этот вид баз данных отличается от реляционных и, в некоторый случаях, имеет рад преимуществ перед ними.

В реляционных базах данных ссылки на другие строки и таблицы указаны ссылками на их (первичные) ключи через внешние ключи столбцов.

Например, если в базе много отношений многие-ко-многим, то имея реляционную базу, вам придется использовать join или создавать распределительную таблицу с внешними ключами на обе связанные таблицы.

При большом количестве таких соединений, каждое последующее соединение может существенно увеличивать затрату ресурсов памяти программы, к тому же, соединения происходят во время запроса к базе, что может также замедлить работу программы или даже стать причиной сбоя. Такие сложные запросы, соединяющие информацию из многих таблиц, обычно пишутся в виде функций. Такой подход позволяет сэкономить ресурсы, однако не приемлем для решения данной задачи. Пользователь задает вопрос на естественном языке, а не выбирает из списка предложенных, это значит, что для поиска ответа, программа должна динамически сгенирировать запрос к базе данных и написать процедуры заранее невозможно.

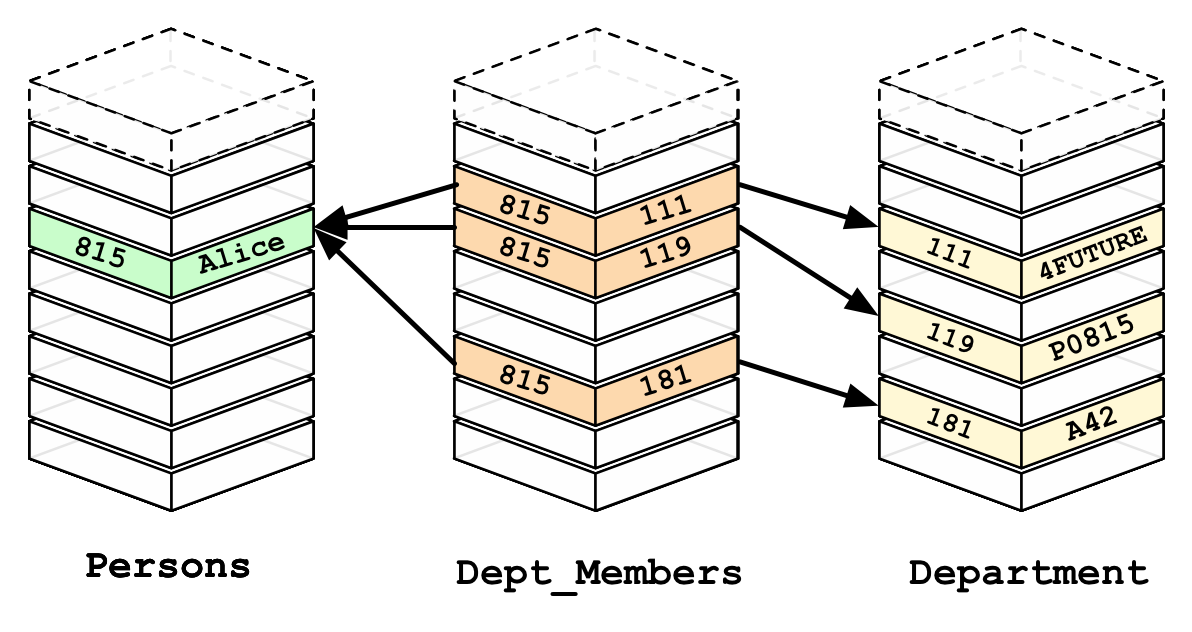


Рисунок 1.1 - соединение таблиц в реляционной базе данных.

В графовых базах отношения являются гражданами первого класса, в отличие от реляционных, в которых необходимо сделать выводы о связях между объектами и ввести определенные ключи. Каждый узел (субъект или атрибут) в модели базы данных графа непосредственно и физически содержит список записей отношений, которые представляют его связи с другими узлами. Эти записи-отношения сгруппированы по типу и направлению может содержать дополнительные атрибуты. Всякий раз, при запуске эквивалента операции присоединения, база данных просто использует этот список и имеет прямой доступ к подключенным узлам, затрачивая на это меньше ресурсов, чем у реляционных баз на операции присоединения. Эта способность обеспечивает графовым бд выигрыш в производительности, особенно на больших объемах данных.

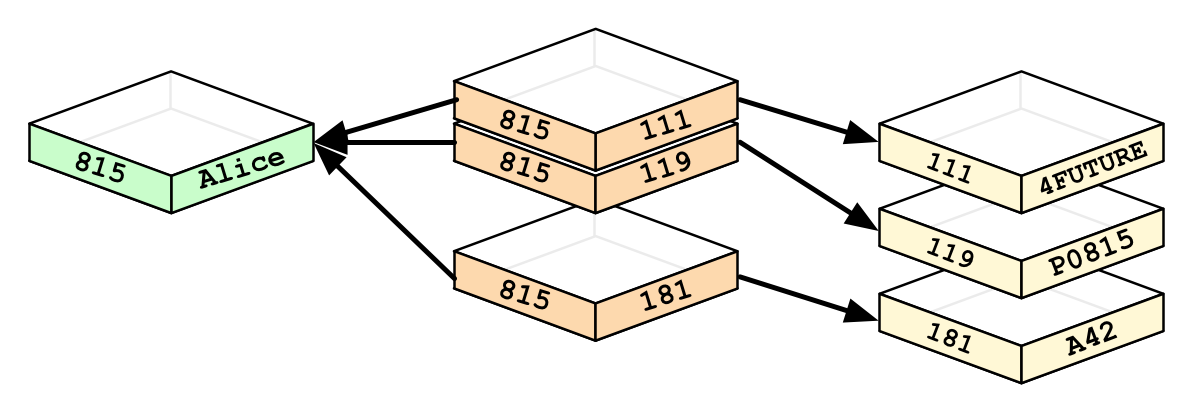


Рисунок 1.2 – модель с графовой базе данных.

В отличие от модели данных на рисунке 1.1 (реляционная), модель на рисунке 1.2 выглядит гораздо проще и выразительнее. Графовые базы данных поддерживают гибкую, мелкозернистую модель данных. Это позволяет добавлять даже существенные изменения, без необходимости модернизировать систему и модель данных. Это свойство также необходимо для базы данных вопросно-ответной системы, поскольку в систему постоянно необходимо загружать новые данные о событиях, которые могут быть связаны с уже имеющимся в базе особым образом.

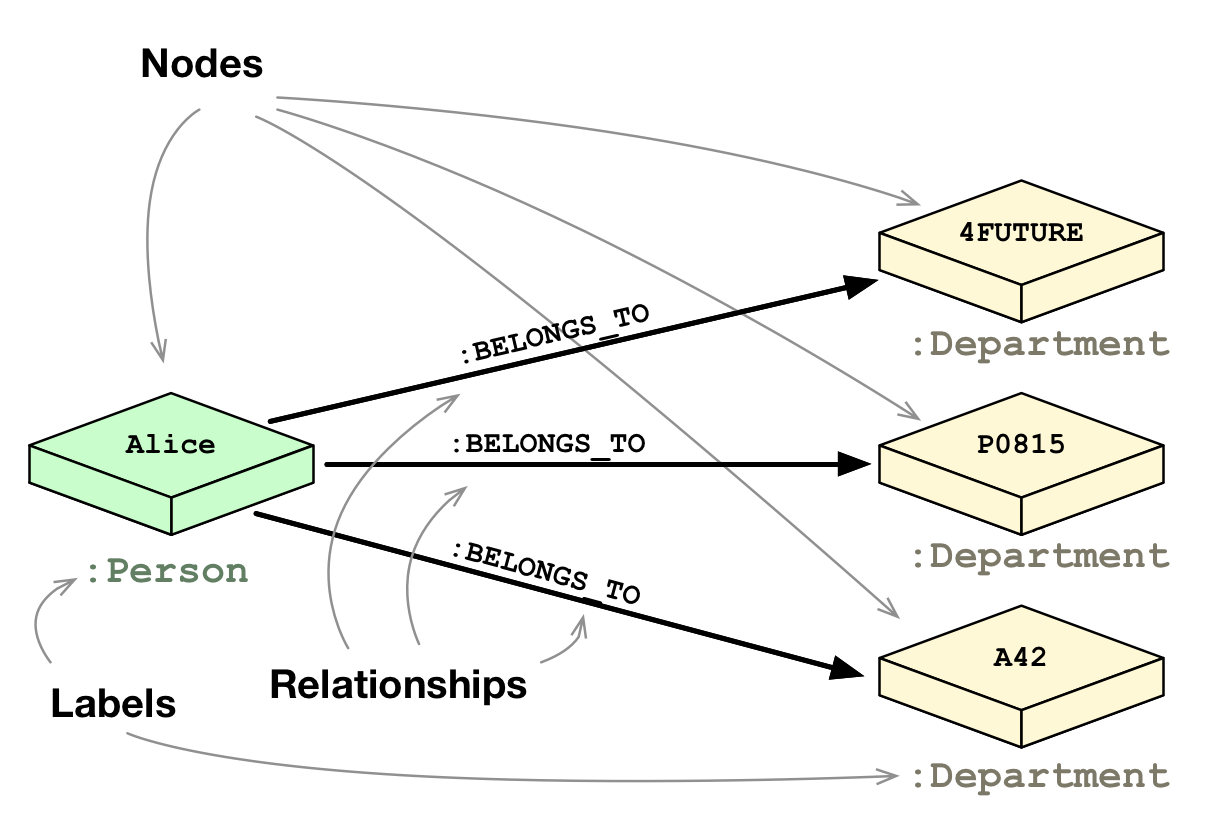


Рисунок 1.3 – узлы и связи в графовой бд

**Глава 2. ИССЛЕДОВАНИЕ**

Для того чтобы разработать прототип системы хранения поиска и представления информации, необходимо разобраться в ней на всех этапах построения и составить задание, учитывающее особенности системы и процесса ее построения. Основные шаги включают в себя выбор, в каком виде система будет получать входные параметры, создание информационной модели базы данных, алгоритмов поиска тематических данных и представления их для пользователя. Данная база должна содержать информацию о множестве событий, людях и связях между ними. В целом процесс формирование задания будет проходить образом, описанным далее.

* 1. **База данных**

Как уже было отмечено, выбор базы данных должен соответствовать поставленной задаче и требованиям. Данные о событиях, людях, местах и прочие данные, которые можно получить из новостных статей могут быть разнообразными, связи между ними произвольными, а значит, база должна удовлетворять следующим требованиям:

* Высокая производительность при больших объемах данных и связей между ними, в основном для сложных запросов.
* Возможность вносить крупные изменения без необходимости менять модель данных.
* База не должна быть сложной в заполнении и обслуживании (так как предполагается, что новые данные в базу будет заносить программа, на основе новостных статей)

Всем вышеперечисленным условия соответствуют графовые бд, это класс NoSQL. Среди графовых баз была выбрана субд Neo4j, так как по заявлениям разработчиков, ее логическая модель в то же время является и физической, что позволяет быстро вносить изменения в базу. Язык субд - Cypher представлением похож на SQL и прост в изучении, поддерживаются ряд API и сообщество, которое поддерживает создание справочных материалов и тематических форумов. Кроме всего перечисленного, Neo4j является достаточно надежной и производительной субд.

Рассмотренные альтернативы Neo4j не обладали таким количеством справочных материалов и удобством использования и представления информации.

**2.2 Информационная модель**