УДК 004.6-024.11:004.738.5

АРХИТЕКТУРНЫЕ РЕШЕНИЯ БЫСТРОГО ПОСТРОЕНИЯ ГРАФОВОЙ БД WEB-САЙТА И АНАЛИЗА ЕГО СВОЙСТВ





М.П. Батура ***И.И. Пилецкий*** ***Н.А.Волорова*** ***П.А. Зорко*** ***А.О. Кулевич***

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: bmpbel@bsuir.by, ianmenski@gmail.com, volorova@bsuir.by, kulevich.01@gmail.com, polina.zorko16@gmail.com

М.П. Батура
Заведующий лабораторией НИЛ 8.1 «Новые обучающие технологии» БГУИР, Доктор технических наук, профессор, академик «Международной академии наук высшей школы», заслуженный работник образования Республики Беларусь. Область научных исследований: Системный анализ, управление и обработка информации в технических и организационных системах. Опубликовано более 150 научных работ.

И.И. Пилецкий
Кандидат физико-математических наук, доцент БГУИР, имеет более 100 публикаций, сфера научных интересов – разработка проектов по обработке больших объёмов данных.

Н.А. Волорова
Заведующая кафедрой информатики БГУИР, кандидат технических наук, доцент. В сфере IT более 40 лет. Имеет более 140 публикаций, сфера научных интересов – модели сложных систем.

П.А. Зорко
Студент БГУИР специальности «Информатика и технологии программирования».

А.О. Кулевич
Студентка БГУИР специальности «Информатика и технологии программирования».

Аннотация. В статье приводится описание архитектурных решений быстрого построения компонента тематического прототипа графической БД, графа знаний из открытых интернет-источников с целью глубокого анализа данных сайта, выявления скрытых зависимостей в некоторой научной области. Описываются принятые решения, демонстрируются, результаты работы компонента получение данных с web-сайта.

Ключевые слова*: интернет-источники, Big Data, анализ, графовая БД, RDF, Neo4j, RDF словари.*

Введение.

Одна из больших технических задач — это получение данных из конкретного сайта. Но, еще более сложная проблема — это анализ данных размещенных на данном сайте. Что приводит к появлению ряда новых возможностей моделирования структуры сайта, одна из которых - возможность предложить простой способ представления свойств отношений в виде графовой модели. Что позволяет анализировать свойства сайта с помощью графовых моделей. Многие сайты *формально поддерживает тройки RDF (Resource Description*

Framework), в которых тройка имеет вид *«субъект — предикат — объект»* и называется **триплетом (тройкой).** Множество RDF-утверждений образует ориентированный граф, в котором вершинами являются субъекты и объекты, а рёбра отображают отношения.

С помощью графовых технологий можно преобразовать представление сайта в графовую БД со свойствами. Что позволяет анализировать свойства сайта.

Последние данные Gartner трендов в области ИТ «Gartner Identifies Top 10 Data and Analytics Technology Trends for 2021» показывают возрастающую роль графовых технологий, так к 2025 году графовые технологии будут использоваться в 80% инноваций в области данных и аналитики по сравнению с 10% в 2021 году, что будет способствовать быстрому принятию решений в организации [1].

Графы — один из самых мощных и гибких способов представления данных. Они обладают большой выразительной силой, что позволяет их применять для моделирования различных физических систем (например, дороги, трубопроводы, электросети, воздушные авиалинии, логистику и т.д.) и социальные сетей (например, LinkedIn, MySpace, Facebook, ВКонтакте, Академические и т.д.). Граф — это универсальная и выразительная структура, позволяющая моделировать всевозможные структуры данных, например, молекулярные взаимодействия и биологические процессы, фармакология, клинические, медицинские и научные публикации. Графы позволяют моделировать сценарии постройки сложных объектов (например, самолеты, ракеты, корабли и т.д.). В настоящее время графовые модели и графовы БД широко применяются во всех областях моделирования искусственного интеллекта.

Графованые базы по сравнению с другими NoSQL БД обладает свойствами OLTP и OLAP. Графовые технологии обеспечивают организации транзакционных графовых хранилищ, интеллектуальный анализ данных и аналитическую обработку данных в реальном времени, поддерживают транзакции ACID, что не обеспечивает ни одна NoSQL БД. Графовые технологии являются основой для построения интеллектуальных приложений, для применения алгоритмов искусственного интеллекта. Важным отличие графовых баз данных является то, что они *явно описывают зависимости между узлами данных*, в то время как *другие базы данных связывают данные неявными связями* [2, 3]. Для NoSQL вместо свойств ACID ввели свойства BASE, которые значительно слабее гарантий ACID, и между ними нет прямого соответствия.

В сложных реляционных базах данных, где количество сущностей может состоять из нескольких сот [4] (например, 300 или 400 сущностей), связность сущностей приводит к увеличению соединений, которые снижают производительность и затрудняют внесение в базу данных новых обновлений. И как правило, выполнение сложных запросов будет медленными, а сама БД со временем может полностью деградировать.

Совместное применение графовых технологий, методов и алгоритмов машинного обучения позволяет получать скрытые зависимости и выполнять предиктивный анализ информации, получать ответы в режиме реального времени, реализовывать алгоритмы искусственного интеллекта.

Графовые базы данных позволяют хранить сущности и отношения между ними. Сущности моделируются узлами (nodes), которые имеют свойства. Узел интерпретируется как экземпляр объекта в приложении. Отношения моделируются ребрами (relationships или edges), которые могут иметь свойства. Neo4j БД отдельно хранит в специальном жестком формате структуру узлов и отношений, а данные (свойства) определяются как пара ключ-значение. Такое решение в базе Neo4j [5] позволяет извлекать данные без обхода графа, с определенного узла (смещение).

Основные решения для быстрого построения прототипа компонента получения и анализа данных с интернет источников.

Основными компонентами построения системы анализа данных Интернет источников являются: компонент получения данных с интернет источников; компонент графовая БД и граф знаний; а также компонент извлечения свойств из графовой БД и их анализ с помощью алгоритмов ML. Компонент извлечения свойств из графовой БД обладает дополнительной функциональностью, может использовать технологию включений (эмбеддинга), что позволяет построить векторы свойств меньшей размерности для более глубокого анализа данных. Графовые включения – это методология представления свойств сущностей (узлов) и свойств отношений в графе как вектор свойств.

Одним из сложных современных направлений является *представление знаний с помощью специальных глобальных словарей предметных областей*, мета-описаний и специальных языков, и методологий их применения. Данная методология широко применяется для создания и описания содержимого различный широко известных сайтов: Wikipedia, DBpedia, TerMef and French Bioloinc Portal, TerMef, BIOLOINC, WikiData, Scientific Research Publishing и крупных организаций.

Многие сайты используют специальную технику описания ресурса Resource Description Framework (RDF, «среда описания ресурса»). Импорт и экспорт RDF может быть в нескольких форматах (Turtle, N-Triples, JSON-LD, RDF / XML, TriG и N-Quads, TriG *).

Ресурсом в RDF может быть любая сущность — как информационная (например, веб-сайт или изображение), так и неинформационная (например, человек, город или некое абстрактное понятие). Утверждение, высказываемое о ресурсе, имеет вид «субъект — предикат — объект» и называется триплетом (тройкой**)**. Например, утверждение «ель зеленого цвета» в RDF-терминологии можно представить следующим образом: субъект — «ель», предикат — «имеет цвет», объект — «зеленый». Для обозначения субъектов, отношений и объектов в RDF используются URI (универсальный идентификатор ресурса), на рисунке 1 приведена схема триплета RDF.

Рисунок 1. Триплет RDF

Множество RDF-утверждений образует ориентированный граф, в котором вершинами являются субъекты и объекты, а рёбра отображают отношения.

Что же касается веб-сайтов, которые хранят данные в JSON-LD, то в коде страницы можно найти нужный нам скрипт JSON-LD. Если он присутствует, то данный сайт можно сериализовать в графовую базу данных способом, описанным в данной работе, если же такого скрипта нет, то понадобится искать другой способ для сериализации данных веб-сайта.

Кроме RDF и JSON, описываемых в статье, можно сериализовать и другие типы данных. Например, данные на общедоступном популярном сайте Wikipedia [6] хранятся в формате **Turtle** и тоже могут быть сериализованы в графовую базу данных, для сайта Wikidata информация о данных хранится на самом сайте [7].

Словари предметных областей.

Представление знаний с помощью специальных глобальных словарей предметных областей позволяет с помощью специального плагина Neosemantics для графовой БД Neo4j использовать RDF и связанные с ним словари, такие как OWL, RDFS, SKOS и другие для интеграции с компонентами, генерирующими / потребляющими RDF [8].

RDF словари: FOAF (Friend-of-a-Friend), SKOS (Simple Knowledge Organization System), DC (Dublin Core), BIBO (Bibliographic Ontology), SIOC (Semantically-Interlinked Online Communities), DOAP (Description of a Project), Music Ontology.

RDF словари и онтологии: OWL (Web Ontology Language) – язык представления веб-онтологий, *основан на RDF и RDFS*.

SKOS (Simple Knowledge Organization System).

Так пример представления знаний с некоторого сайта, см. ниже, приведен на рисунке 2 [9].

```
ex:psycholing rdf:type skos:Concept;<br />
skos:prefLabel "Психолингвистика”<br />
skos:altLabel "Психология языка»<br />
<skos:narrower rdf:resource=
"http://id.loc.gov/authorities/subjects/sh85125403"/>
<skos:exactMatch rdf:resource=
"http://ex.tw/auth/ps654678"/>
```

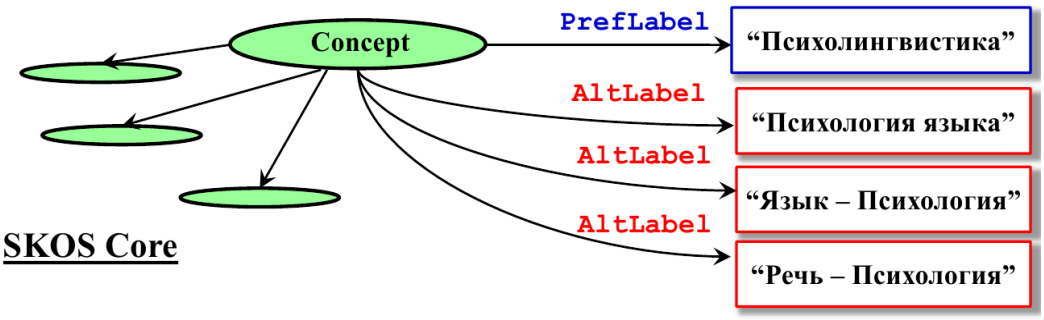

Рисунок 2. Графическое представление данных сайта

Правила преобразования троек RDF в графовую БД.

Существует три основных правила сериализации троек RDF («субъект — предикат — объект») в графовую БД Neo4j**.**

Первое правило. *Субъекты троек отображаются на узлы в Neo4j.*

Узел в Neo4j, представляющий ресурс RDF, будет помечен :Resource и будет иметь свойство uri с URI ресурса.

(S,P,O) => (:Resource {uri:S})...

Второе правило. *Предикаты троек отображаются в свойствах узла в Neo4j, если объект тройки является литералом.*

(S,P,O) && isLiteral(O) => (:Resource {uri:S, P:O})

Третье правило. *Предикаты троек отображаются* на отношения *в Neo4j, если объект тройки является ресурсом*.

(S,P,O) && !isLiteral(O) => (:Resource {uri:S})-[:P]->(:Resource {uri:O})

С помощью модулей библиотеки Neo4j можно множество RDF-утверждений преобразовать в графовую БД со свойствами, что позволяет построить прототип двух компонент системы анализа данных Интернет источников, компонента скачивания данных и компонента построения графовой базы данных и графа знаний.

3. Среда для быстрого построения графовой БД Web-сайта анализа данных Интернет источников.

В качестве ИТ платформы использовалась среда разработки Neo4j Desktop. С Neo4j Desktop можно создавать и управлять любым количеством локальных баз данных, которое

может поддерживать компьютер. Базы данных Neo4j размещаются в экземпляре системы управления базами данных (СУБД), и, начиная с Neo4j 4.0, можно иметь одну или несколько баз данных в одном экземпляре СУБД. Поскольку Desktop может запускать все поддерживаемые в настоящее время версии базы данных Neo4j, можно создать один или несколько экземпляров СУБД для поддержки разных версий Neo4j, разделить свои базы данных по типу данных, которые они содержат, или другим определённым образом для достижения конкретной желаемой конфигурации СУБД. С помощью Neo4j Desktop можно управлять конфигурацией СУБД, добавлять плагины, просматривать журналы, создавать резервные копии и восстанавливать данные, обновлять версии Neo4j и многое другое, чтобы получить полный жизненный цикл работы с Neo4j.

Для построения графовой БД Web-сайта используются специальные плагины Neosemantics (n10s) и APOC, которые нужно заранее установить.

Neosemantics (n10s) – это плагин, который позволяет использовать RDF и связанные с ним словари, такие как OWL, RDFS, SKOS и другие в Neo4j. Данный плагин используется для интеграции с компонентами, генерирующими / потребляющими RDF. Neosemantics работает как расширение базы данных Neo4j. Основными функциями данного плагина являются:

- Импорт и экспорт RDF в нескольких форматах (Turtle, N-Triples, JSON-LD, RDF / XML, TriG и N-Quads, TriG *)
- Отображение модели при импорте и экспорте
- Импорт и экспорт онтологий и таксономий в различных словарях (OWL, SKOS, RDFS)
- Проверка графа на основе ограничений SHACL
- Базовый вывод

APOC (Awesome Procedures on Cypher) – вспомогательный плагин для базы данных. Библиотека APOC считается самой большой и наиболее широко используемой библиотекой расширений для Neo4j. Она включает более 450 стандартных процедур, обеспечивающих функциональные возможности для утилит, преобразований, обновлений графов и многого другого. Они хорошо поддерживаются, и их очень легко запускать как отдельные функции или включать в запросы Cypher.

Плагин n10s имеет множество различных свойств, для начала работы с ним нужно создать конфигурацию Graph, чтобы проинструктировать Neosemantics о том, как хранить данные. Все методы, сохраняющие данные в Neo4j, имеют предварительное условие: это наличие ограничения уникальности для свойства uri узлов с меткой Resource. Поэтому с помощью специальной команды добавляется уникальность для свойства uri узлов.

Примеры технологических решений быстрого построения графовой БД Web-сайта и анализа его свойств.

В настоящее время компоненты системы анализа данных (СКА) Интернет источников реализованы в опытном варианте. В данном разделе приведены результаты работы компонента получения данных с интернет источников и быстрого построения графовой БД Web-сайта и анализа его свойств.

Тройки RDF определены как тройки, которые включают тройку как субъект или объект. Это приводит к появлению ряда новых возможностей моделирования, одна из которых – возможность предложить простой способ представления свойств отношений.

Именно такое использование поддерживается n10s: формально n10s поддерживает тройки RDF, в которых тройка является субъектом, а литерал – объектом. Внутренняя тройка представляет отношения, а внешняя тройка представляет свойство этих индивидуальных отношений.

Построение графа свойств в Neo4j на основе данных RDF.

На сайте Open Food Facts [10] имеется общедоступный набор данных с 6,2 миллионами троек по пищевым продуктам, их ингредиентам, аллергенам, фактам питания и многому другому. Импортируя тройки RDF с помощью запроса, представленного на рисунке 3, получается тематическая графовая БД.

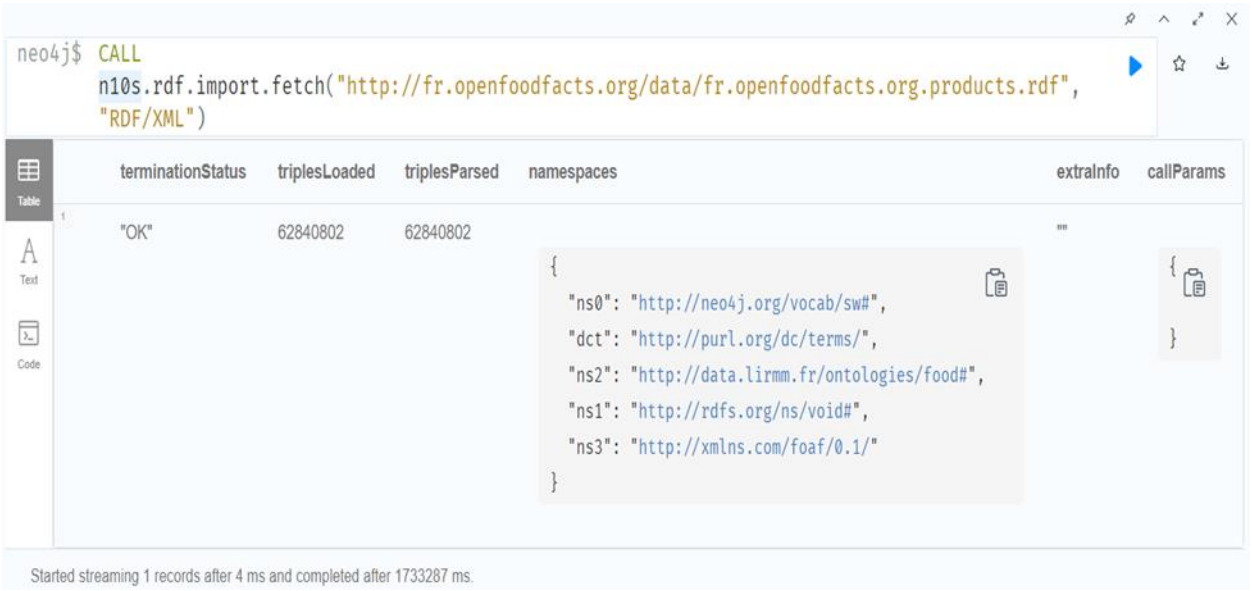

Рисунок 3. Импорт данных RDF с сайта Open Food Facts

Общее представление полученной БД представлено на рисунке 4.

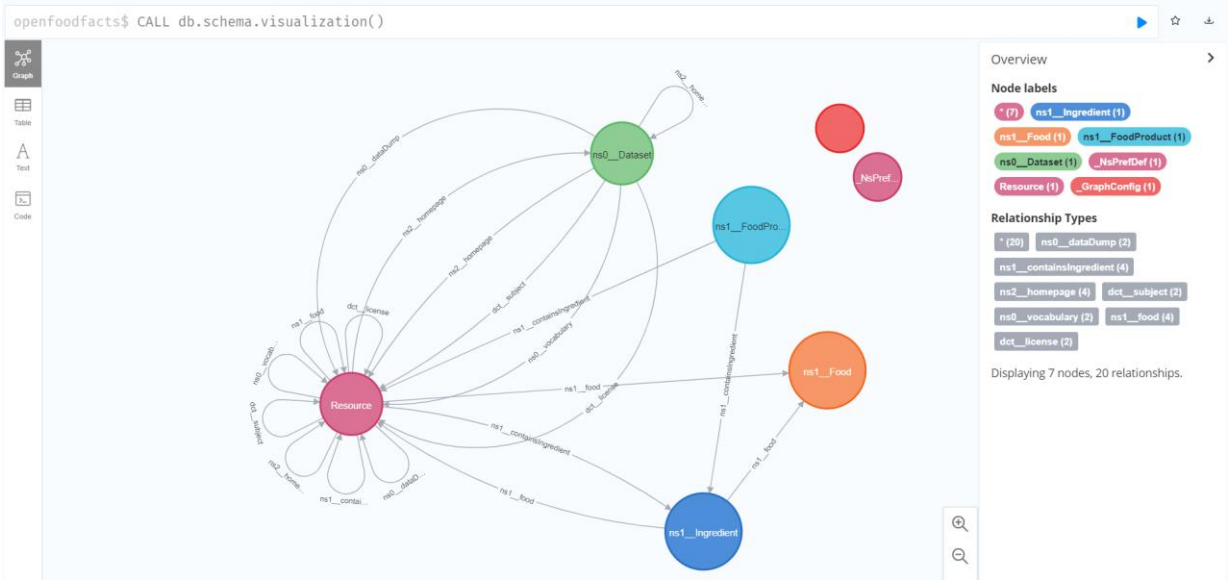

Рисунок 4. Общее представление схемы графовой базы данных

В данной БД основными типами узлов являются:

–*FoodProduct* – продукты;

–*Food* – ингредиенты, которые могут содержаться в продуктах;

–*Ingredient* – промежуточные узлы, соединяющие узлы *FoodProduct* и *Food*, в которых указывается, сколько ингредиента содержится в продукте.

Основными типами связей являются:

–*containsIngredient* – соединяет продукт *FoodProduct* и промежуточный узел *Ingredient*;

–*food* – соединяет промежуточный узел *Ingredient* и ингредиент *Food*.

В полученной базе данных можно искать различную информацию о продуктах и составе их ингредиент. К примеру, с помощью запроса, представленного ниже, можно узнать, какой набор общих ингредиентов у двух продуктов. Результат запроса показан на рисунке 5.

```
MATCH (prod1:Resource { uri: 'http://world-fr.openfoodfacts.org/produit/5053827196642/coco-pops-kellogg-s'})
MATCH (prod2:Resource { uri: 'http://world-fr.openfoodfacts.org/produit/5010358227290/2-snack-pork-pies-spar'})
MATCH (prod1)-[:ns1__containsIngredient]->(x1)-[:ns1__food]->(sharedIngredient)<-[:ns1__food]-(x2)<-[:ns1__containsIngredient]-(prod2)
RETURN prod1, prod2, x1, x2, sharedIngredient
```

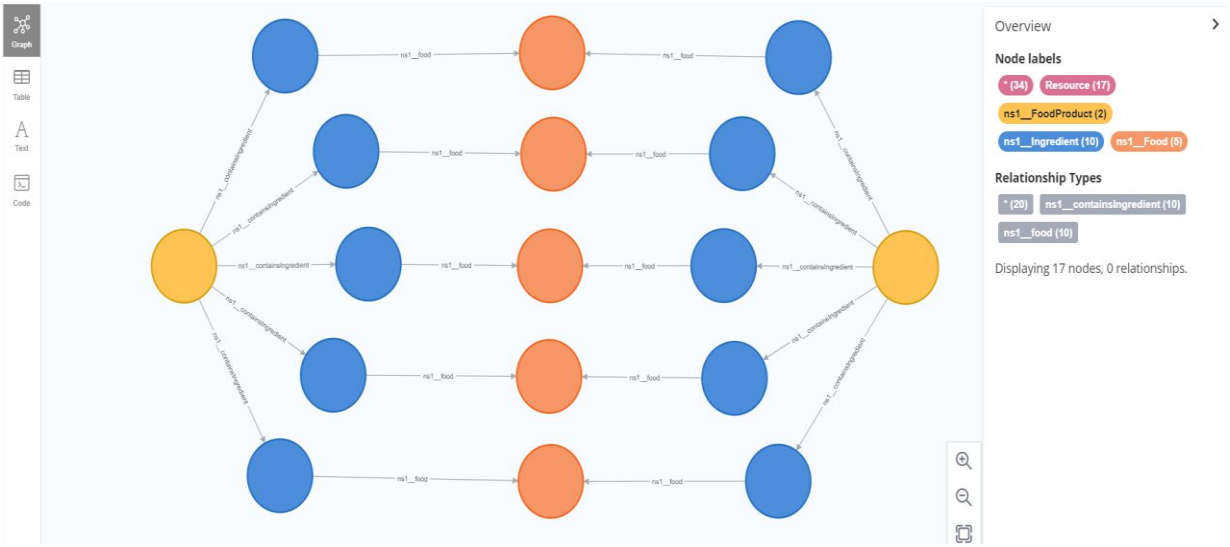

Рисунок 5. Вывод общих ингредиентов у двух продуктов

В данных RDF хранятся все данные и характеристики продуктов и ингредиентов. На рисунке 6 видно, что каждый узел продукта FoodProduct имеет набор свойств, таких как идентификационный номер узла, название продукта, содержание различных микроэлементов в продукте и т.д., каждый узел ингредиента имеет идентификационный номер, название ингредиента и uri, а промежуточный узел между продуктом и ингредиентом Ingredient имеет идентификационный номер, свойство, указывающее на то, сколько ингредиента содержится в продукте и uri.

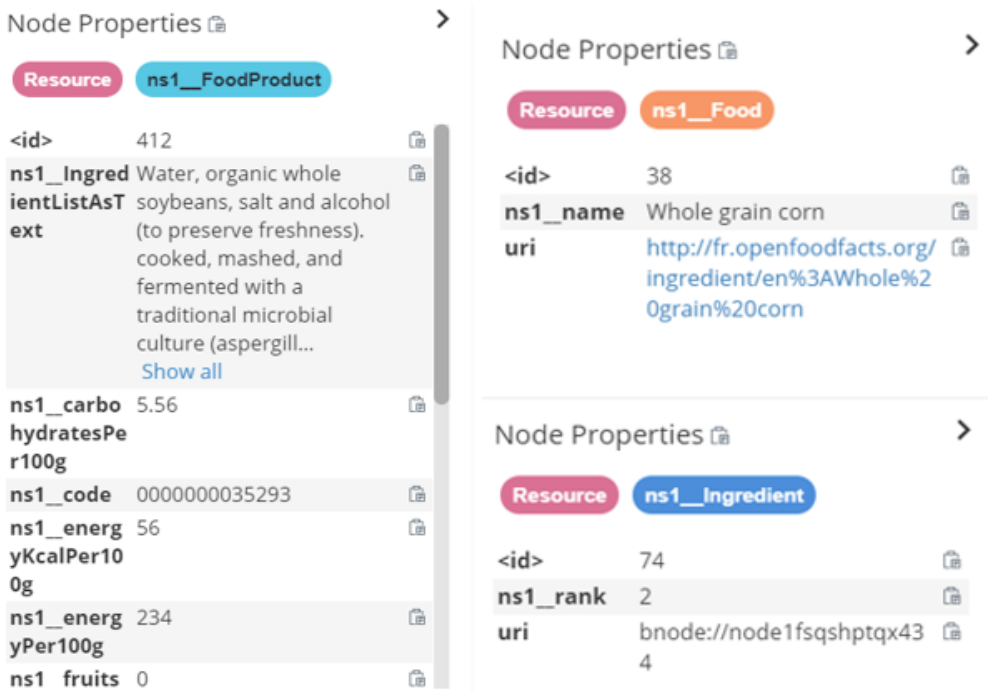

Рисунок 6. Свойства узлов графовой БД с продуктами питания

Таким образом, мы получаем универсальный инструмент для быстрого поиска информации о нужном нам продукте, сортировке и выборке продуктов по определённому свойству и др.

К примеру, вы можем найти все продукты, в которых содержится пастеризованное молоко и в которых на 100 г продукта содержится меньше 100 ккал. Для этого требуется выполнить следующий запрос. Результат запроса представлен на рисунке 7.

```
MATCH (f:ns1__FoodProduct)
WHERE ((any(ingridient in split(f.ns1__IngredientListAsText,',') WHERE ingridient = 'Pasteurized milk')) and (toFloat(f.ns1__energyKcalPer100g) < 100))
RETURN f LIMIT 100
```

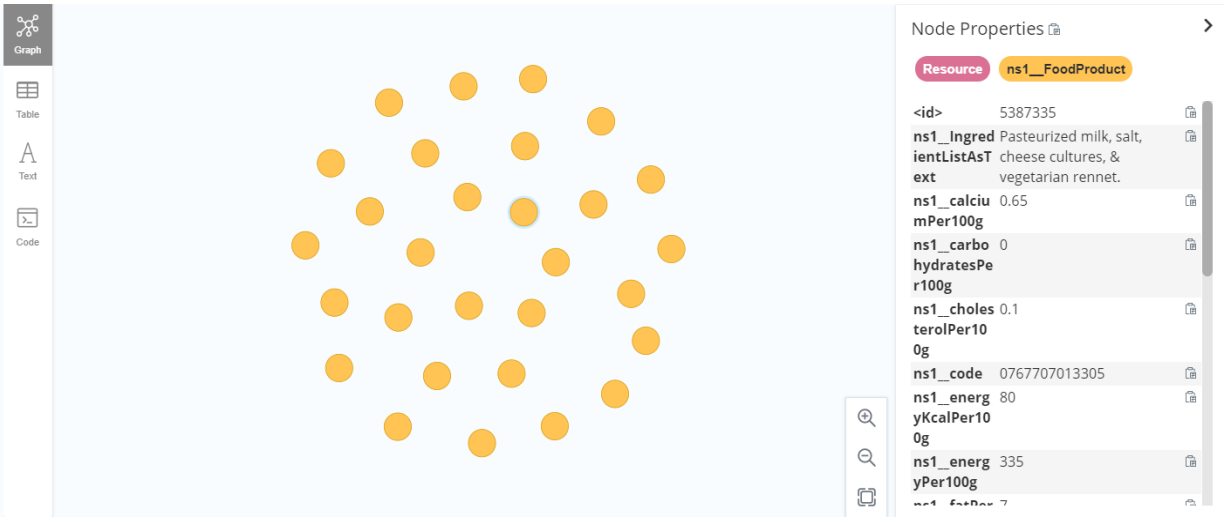

Рисунок 7. Продукты, содержащие пастеризованное молоко и содержащие меньше 100 ккал на 100 г

По данному запросу, в нужную выборку продуктов попало 29 продуктов питания.

Рекомендуемая норма соли для человека в день не более 5 грамм. А пальмовое масло для человека не очень полезное. С помощью следующего не простого запроса можем найти все продукты, которые не содержат пальмовое масло и содержания соли которых на 100 г меньше 0,5 г. Результат запроса представлен на рисунке 8.

```
MATCH (f:ns1__FoodProduct)
WHERE ((any(ingridient in split(f.ns1__IngredientListAsText,',') WHERE ingridient <> 'palm oil')) and (toFloat(f.ns1__saltPer100g) < 0.5))
RETURN f LIMIT 100
```

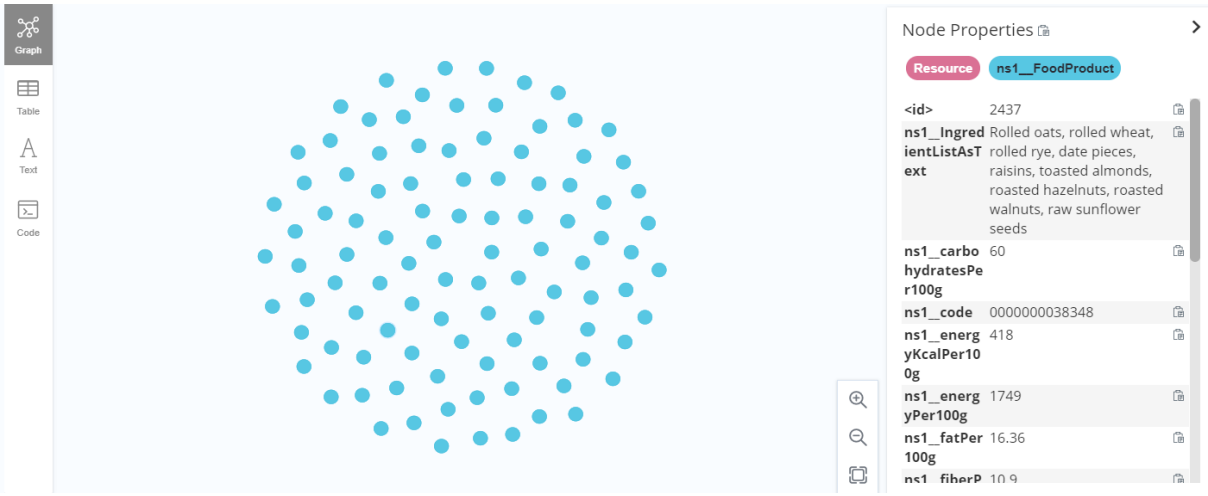

Рисунок 8. Продукты, не содержащие пальмового масла и содержащие меньше 0,5 г соли на 100 г

Построение графа свойств на основе данных, встроенных в веб-страницы (JSON).

При сериализации данных, наиболее распространённым типом данных является JSON. Он часто используется на сайтах со статьями и публикациями. Рассмотрим сайт со статьями на научные темы Oxford Academic [11]. Как правило, все элементы, нужные для сериализации и построения графовой БД находятся в коде страницы сайта в script элементе с типом *application/ld+json*.

Здесь в специальном формате представлены необходимые данные об узлах, их свойствах и связях между ними будущей графовой базы данных. Для извлечения информации с веб-страницы можно использовать процедуру APOC **apoc.load.html**, представленную ниже. Для этого требуется URL-адрес страницы и CSS-подобный селектор, чтобы выбрать конкретный элемент, который вам нужен. Результат запроса представлен на рисунке 9.

```
CALL apoc.load.html("https://academic.oup.com/journals", { jsonld: 'head script[type="application/ld+json"]'}) YIELD value
```

Рисунок 9. Извлечение информации с веб-страницы с помощью APOC

Для визуализации и анализа RDF используется плагин Neosemantics. С помощью комбинированного запроса, представленного ниже, можно сериализовать данные главной страницы сайта и трёх его статей в RDF. Результат сериализации представлен на рисунке 10.

```
UNWIND ["https://academic.oup.com/journals", "https://academic.oup.com/jat/advance-article/doi/10.1093/jat/bkab114/6422696?searchresult=1", "https://academic.oup.com/nargab/article/3/4/lqab103/6423165?searchresult=1", "https://academic.oup.com/pcp/advance-article/doi/10.1093/pcp/pcab162/6422744?searchresult=1"] as page
CALL apoc.load.html(page, { jsonld: 'head script[type="application/ld+json"]'}) YIELD value
CALL n10s.rdf.import.inline(value.jsonld[0].data, "JSON-LD")  yield terminationStatus, triplesLoaded, triplesParsed, extraInfo
RETURN page, terminationStatus, triplesLoaded, triplesParsed, extraInfo
```

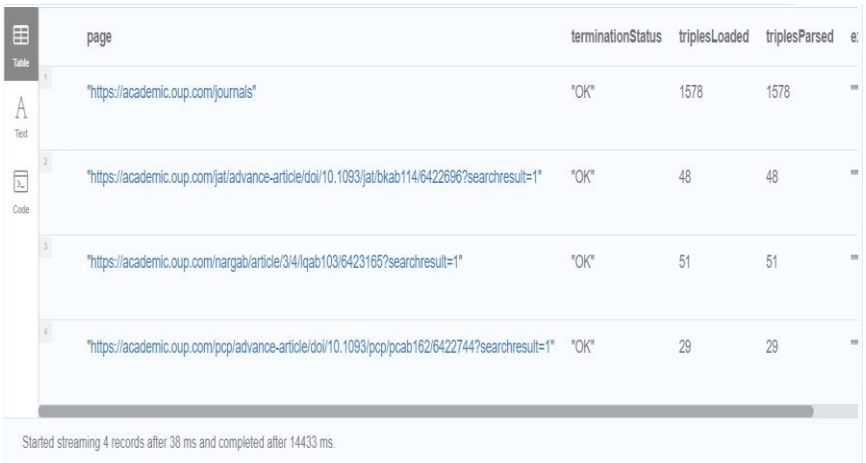

Рисунок 10. Сериализация главной страницы и трёх статей сайта Oxford Academic

На рисунке 11, приведено общее представление графовой базы данных, полученной из сайта с помощью технологии быстрого построения тематической графовой БД.

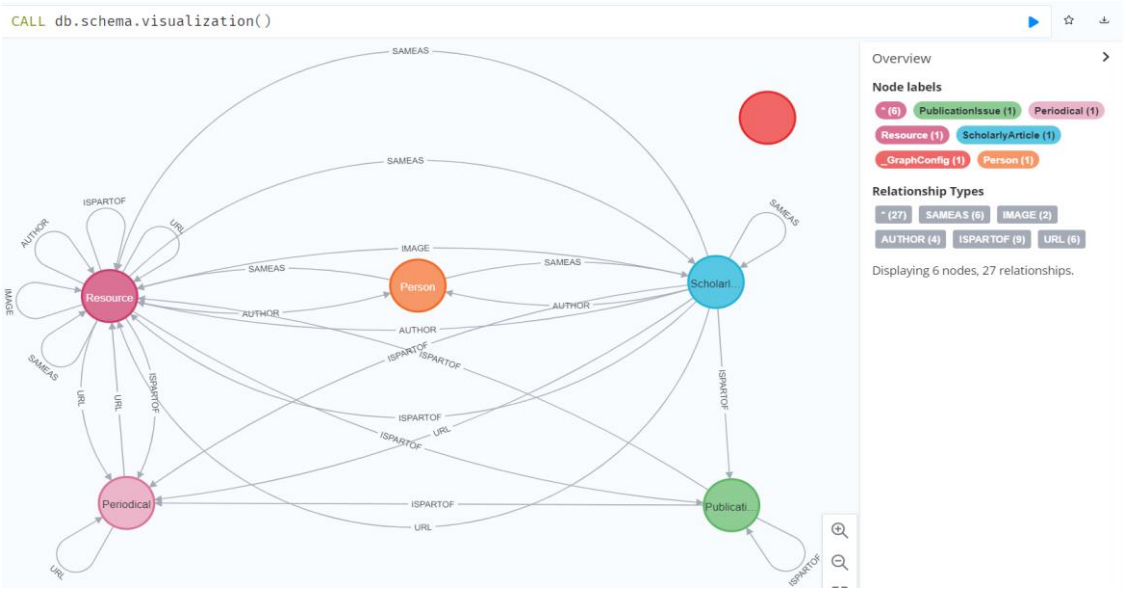

Рисунок 11. Общее представление графовой БД сайта Oxford Academic

После сериализации образовались узлы 6 видов и связи 5 видов. Фрагмент JSON-LD использует Schema.org словарь [12], откуда берутся все элементы в схеме графа. Каждый элемент имеет определённый тип и свойства, характерные только ему.

В случае получившегося запроса имеются следующие виды узлов:

– *Periodical* – тип сайта, означающий, что на сайте имеются публикации, выпущенные в последовательных частях с числовыми или хронологическими обозначениями и предназначенные, например, для журнала, научного журнала или газеты на неопределенный срок;

– *ScholarlyArticle* – научная статья или публикация;

– *Person* – человек, в случае со статьями, автор или издатель статьи;

– *PublicationIssue* – часть последовательно издаваемой публикации, такой как периодическое издание или том публикации, часто пронумерованная, обычно содержащая группу работ, таких как статьи;

– *_GraphConfig* – узел, содержащий в себе все настройки и конфигурацию плагина Neosemantics;

– *Resource* – все остальные узлы, которые не попали под какую-либо из категорий.

Имеются следующие виды связей:

– *sameAs* – связь, которая указывает на идентичность элементов;

– *image* – изображение предмета. Это может быть URL-адрес или полностью описанный ImageObject;

– *author* – автор какой-либо публикации или статьи;

– *isPartOf* – указывает на элемент, частью которого является этот элемент

– *url* – URL-адрес элемента.

Созданная ИТ среда Neo4j Desktop позволяет анализировать какие веб-страницы и элементы связаны друг с другом и какой связью, а также объединять несколько похожих графовых баз данных (к примеру, различные сайты со статьями) в одну графовую базу данных. При этом повторяющиеся узлы, которые будут присутствовать в нескольких баз данных, при объединении так же будут объединяться.

Сериализуем данные (главную страницу и три статьи) с ещё одного сайта со статьями Medium [13]. Для этого выполним запрос, указанный ниже. На рисунке 12 результат запроса.

```
UNWIND ["https://medium.com", "https://medium.com/zoey-writes/my-husbands-white-mistress-cut-our-black-daughter-s-hair-without-my-consent-6bbaed606130", "https://baos.pub/5-books-i-have-recommended-over-100-times-a62049a8e90e", "https://entrylevelrebel.medium.com/science-just-confirmed-elon-musks-favorite-interview-question-is-brilliant-2a1e328592f5"] as page
CALL apoc.load.html(page, { jsonld: 'head script[type="application/ld+json"]'}) YIELD value
CALL n10s.rdf.import.inline(value.jsonld[0].data, "JSON-LD")  yield terminationStatus, triplesLoaded, triplesParsed, extraInfo
RETURN page, terminationStatus, triplesLoaded, triplesParsed, extraInfo
```

	page	terminationStatus	triplesLoaded	triplesParsed	extraInfo
1	"https://medium.com"	"OK"	10	10	""
2	"https://medium.com/zoey-writes/my-husbands-white-mistress-cut-our-black-daughter-s-hair-without-my-consent-6bbaed606130"	"OK"	30	30	""
3	"https://baos.pub/5-books-i-have-recommended-over-100-times-a62049a8e90e"	"OK"	30	30	""
4	"https://entrylevelrebel.medium.com/science-just-confirmed-elon-musks-favorite-interview-question-is-brilliant-2a1e328592f5"	"OK"	30	30	""

Рисунок 12. Сериализация главной страницы и трёх статей сайта Medium

На рисунке 13, приведено общее представление графовой базы данных, полученной из сайта Medium.

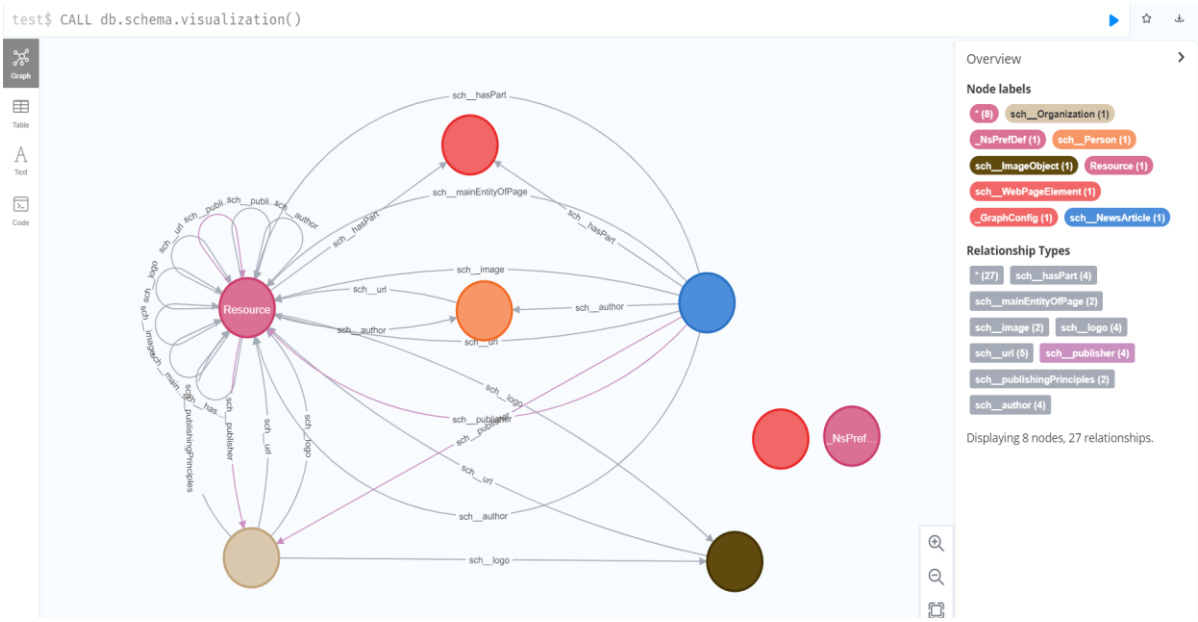

Рисунок 13. Общее представление графовой БД сайта Medium

С помощью таких запросов можно сериализовать данные с разных сайтов в одну графовую БД. На рисунке 14 представлено общее представление графовой БД сайтов Medium и Oxford Academic. На данном представлении видно, что узлы (Person, NsPreDef, Resource, GraphConfig), которые были и в первой, и во второй графовой базах данных, объединились, как и повторяющиеся типы связей (image, url, author).

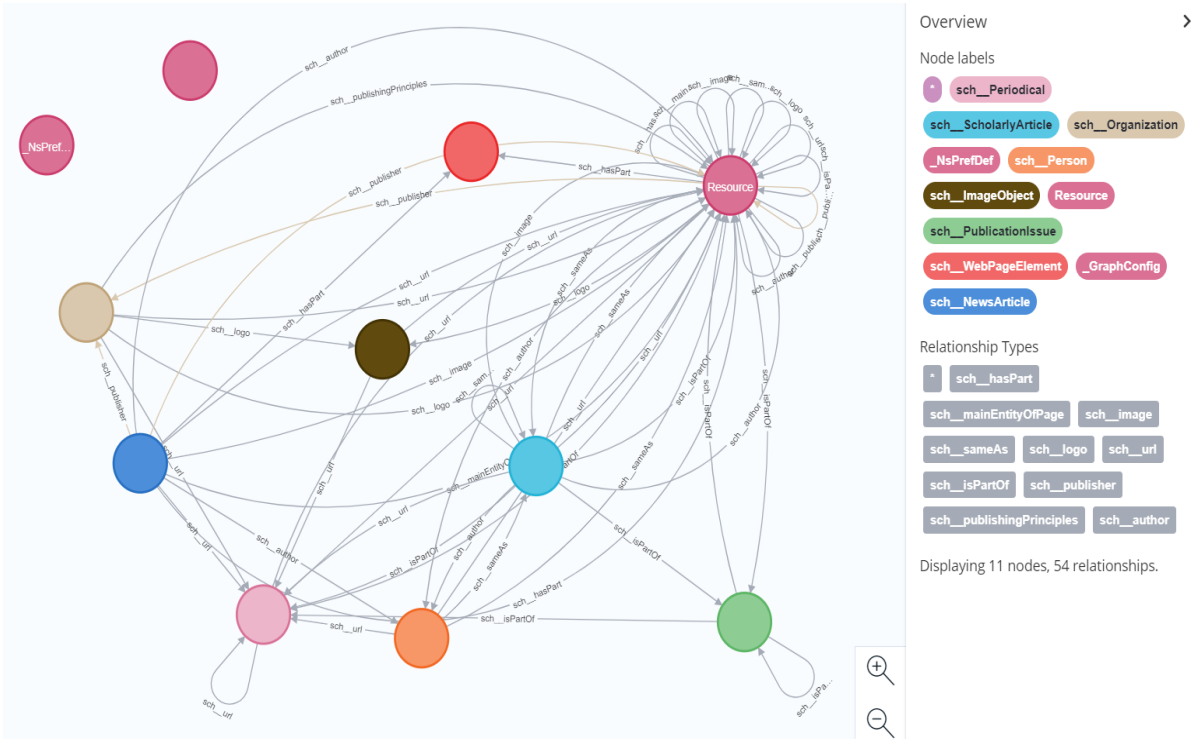

Рисунок 14. Общее представление графовой БД сайтов Medium и Oxford Academic

Построение тематической графовой БД на основе данных Turtle.

В качестве примера технологии построения графовой БД будут использоваться данные на общедоступном популярном сайте Wikipedia [6]. Эти данные хранятся в формате **Turtle** и могут быть сериализованы в графовую базу данных с помощью комбинирования запросов SPARQL и возможностей Neo4j (информация о данных хранится на сайте Wikidata [7]).

С помощью следующего запроса, представленного на рисунке 15, сериализуем данные о населении Беларуси за последние 12 лет, начиная с 1 января 2010 года.

```
WITH 'PREFIX neo: <neo4j://voc#>  1
CONSTRUCT { 2
 ?country a neo:Country . 3
 ?country neo:countryName ?countryLabel . 4
 ?country neo:inContinent ?continent . 5
 ?continent neo:continentName ?continentLabel . 6
 ?country neo:hasPopulationCount [ neo:population ?population ; neo:onDate ?date ] . 7
 ?population a neo:PopulationCount
}
WHERE {
  ?country wdt:P17 wd:Q184 ;
   rdfs:label ?countryLabel .
   filter(lang(?countryLabel) IN ("en",  "ru")) .
 ?country wdt:P30 ?continent .
 ?continent rdfs:label ?continentLabel .
 filter(lang(?continentLabel) IN ("en",  "ru")) .
 ?country p:P1082 ?populationStatement .
 ?populationStatement ps:P1082 ?population;
   pq:P585 ?date .
 filter(?date > "2010-01-01"^^xsd:dateTime)
}
LIMIT 20' AS sparql

CALL n10s.rdf.preview.fetch(
 'https://query.wikidata.org/sparql?query='+ apoc.text.urlencode(sparql),
 'Turtle' ,
  { headerParams: { Accept: "application/x-turtle" } }
)
YIELD nodes, relationships
RETURN nodes, relationships
```

Рисунок 15. Запрос для сериализации данных о населении Беларуси

Перевод данных в RDF осуществляется запросом SPARQL CONSTRUCT, в результате выполнения которого поток троек субъекта, предиката и объекта, которые вместе представляют граф RDF.

SPARQL CONSTRUCT состоит из следующий строк: 1 – данный оператор определяет пространство имен neo4j; 2 – раздел CONSTRUCT запроса определяет триплеты; 3 – neo:Country заменяется на wdt:P17 (страна) wd:Q184 (Беларусь); 4 – узел страны будет иметь свойство countryName; 5 – страна будет иметь inContinent отношение к своему континенту; 6 – континент будет иметь continentName свойство; 7 – для подсчёт населения создаётся тройка, чтобы представить отношение к новому узлу со свойствами для даты и подсчета.

После выполнении запроса получится результат, представленный на рисунке 16.

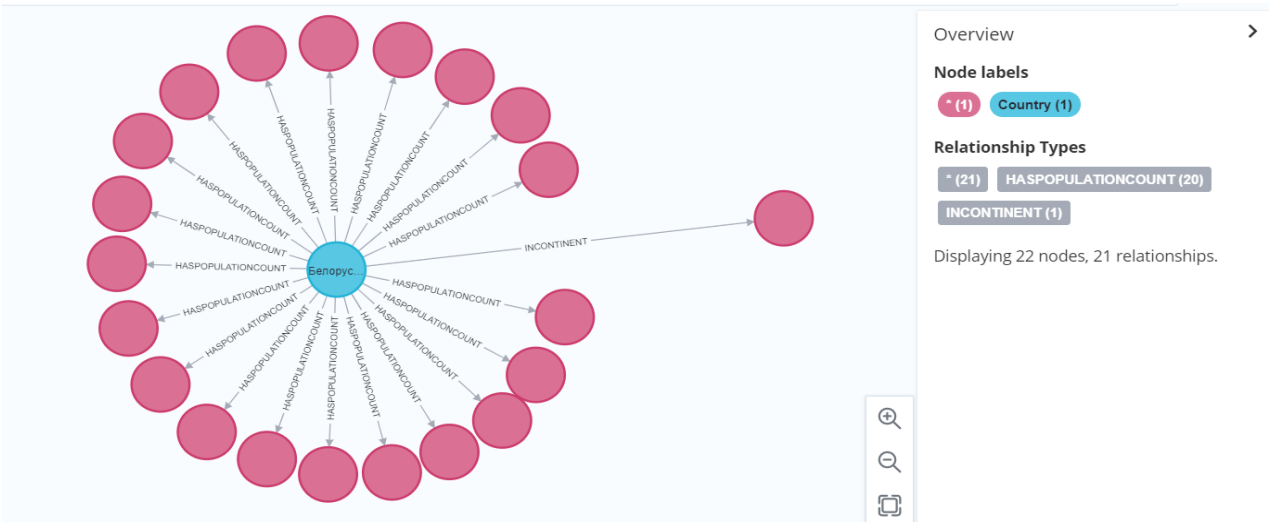

Рисунок 16. Результат выполнения запроса

На данном графе образовалось два типа связей. Связь *INCONTINENT* связывает главный узел *Белоруссия* с узлом *Европа*, а связи HASPOPULATIONCOUNT связывает главный узел *Белоруссия* с узлами, значения которых показывают население Беларуси в разные годы. Свойства некоторых из узлов представлены на рисунке 17.

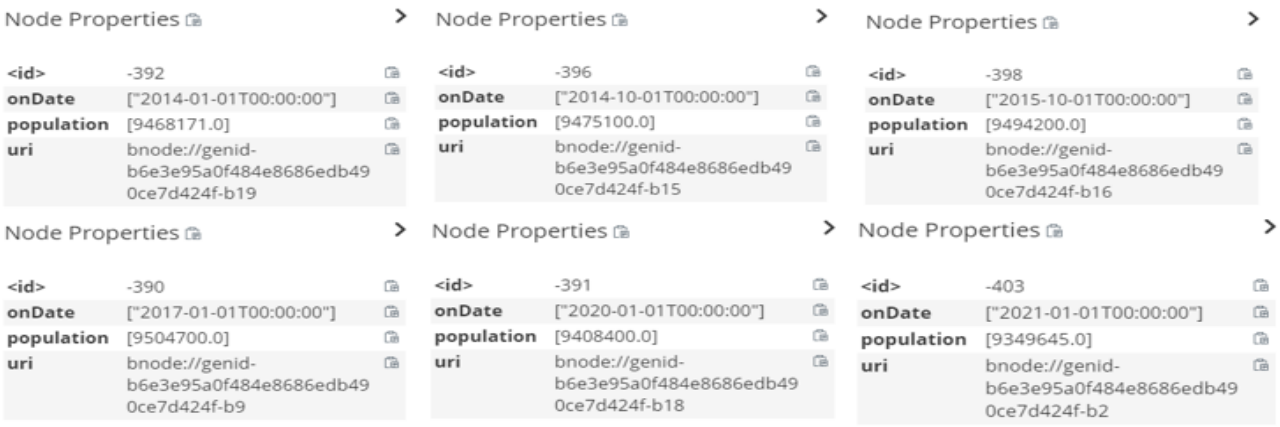

Рисунок 17. Свойства узлов Населения

Из полученных свойств узлов можно узнать численность населения Беларуси в разное время (см. таблицу 1).

Таблица 1. Численность населения Беларуси

Дата	01.01.2014	01.10.2014	01.10.2015	01.01.2017	01.01.2020	01.01.2021
Население, чел	9468171	9475100	9494200	9504700	9408400	9349645

Заключение.

Результаты приведенные в статье представляют собой инновационный научно-образовательный проект БГУИР. Результаты выполнения проекта используются при обучении студентов и магистрантов по тематике «Обработка больших объемов информации».

В данной статье проанализировано довольно сложное современное направление – как представление знаний с помощью специальных глобальных словарей предметных областей, мета-описаний и специальных языков, и методологий их применения. Данная методология широко применяется для создания и описания содержимого различный широко известных сайтов: Wikipedia, DBpedia, TerMef and French Bioloinc Portal, TerMef, BIOLOINC, WikiData, Scientific Research Publishing и крупных организаций. Основная проблема состоит в том, как получить данные с данных сайтов для дальнейшего анализа, решение которой представляет не простую аналитическую и техническую задачу.

В работе была рассмотрена модель представления знаний с помощью технологии описания данных Resource Description Framework (RDF, «среда описания ресурса»), выполнены работы по созданию ИТ среды для представления и анализа данных сайтов с помощью графовой БД Neo4j, библиотек APOC и Neosemantics, их использование для сериализации данных разного формата (Turtle, N-Triples, JSON-LD, RDF / XML, TriG и N-Quads, TriG *) с различных сайтов и десериализации.

Выполнен сбор данных сайтов содержащие данные в RDF, JSON-LD и HTML представлении и импортирован в тематические графовые БД Neo4j. С помощью не простых команд получены графы знаний и проведен анализ полученных результатов.

Список использованных источников

[1] Gartner [Электронный ресурс] / Режим доступа: https://www.gartner.com/en/newsroom/ press-releases/2021-03-16-gartner-identifies-top-10-data-and-analytics-technologies-trends-for-2021 Дата доступа: 25.02.22.

[2] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," NIPS 2017, pp. 1024–1034, 2017.

[3] T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto, "Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach," in IJCAI 2017, 2017, pp ICLR 2017, 2017.

[4] Пилецкий И.И., «Эволюционная технология разработки баз данных» //Доклады БГУИР, №3 (41), Минск, БГУИР, 2009 г. - С.107 -112.

[5] What Is Neo4J? // [Электронный ресурс] – Режим доступа: https://neo4j.com/product/neo4j-graph-database/ Дата доступа: 25.02.22.

[6] Wikipedia // [Электронный ресурс] – Режим доступа: https://en.wikipedia.org/wiki/Wikipedia/ Дата доступа: 25.02.22.

[7] Wikidata [Электронный ресурс] / Режим доступа: https://www.wikidata.org Дата доступа: 25.02.22.

[8] Среда Описания Ресурса (RDF): Понятия и Абстрактный Синтаксис [Электронный ресурс] // Режим доступа: https://www.w3.org/2007/03/rdf_concepts_ru/ Overview.html / Дата доступа: 25.02.22.

[9] Авторитетные данные в Семантическом вебе. // [Электронный ресурс] Режим доступа: http://www.rusmarc.ru/publish/auth_semweb.pdf / Дата доступа: 25.02.2022.

[10] Open Food Facts [Электронный ресурс] / Режим доступа: https://world.openfoodfacts.org Дата доступа: 25.02.22.

[11] Oxford Academic [Электронный ресурс] / Режим доступа: https://academic.oup.com/journals Дата доступа: 25.02.22.

[12] Schema.org [Электронный ресурс] / Режим доступа: https://schema.org Дата доступа: 25.02.22.

[13] Medium [Электронный ресурс] / Режим доступа: https://medium.com Дата доступа: 25.02.22.

ARCHITECTURAL SOLUTIONS FOR QUICK CONSTRUCTION OF A GRAPHIC DB WEB-SITE AND ANALYSIS OF ITS PROPERTIES

M.P. Batura

Head of the Laboratory of Research Laboratory 8.1 "New Educational Technologies" BSUIR, Doctor of Technical Sciences, Professor, Academician of the "International Academy of Sciences of Higher Education", Honored Worker of Education of the Republic of Belarus. Research area: System analysis, management and information processing in technical and organizational systems.

I.I. Piletski

PhD, Associate Professor of BSUIR.

H.A.Volorova
Head of the Department of Informatics at BSUIR, PhD, Associate Professor.

P.A. Zorka
Student of BSUIR Faculty of Computer Science and Programming Technologies..

A.O. Kulevich
Student of BSUIR Faculty of Computer Science and Programming Technologies.

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
E-mail: bmpbel@bsuir.by, ianmenski@gmail.com, volorova@bsuir.by, kulevich.01@gmail.com, polina.zorko16@gmail.com

Abstract. The article provides a description of architectural solutions for the rapid construction of a component of a thematic prototype of a graphical database, a knowledge graph from open Internet sources for the purpose of in-depth analysis of site data, identifying hidden dependencies in a certain scientific field. The decisions made are described, the results of the work of the component for obtaining data from the website are demonstrated.

Keywords: Internet sources, Big Data, analysis, graph database, RDF, Neo4j, RDF dictionaries.