Викиданные: обзор системы

Оглавление

Введение

Семантический Вэб (The Semantic Web) — это часть глобальной концепции развития сети Интернет, целью которой является реализация возможности машинной обработки информации, доступной во Всемирной паутине. Термин был впервые введен в статье, опубликованной 17 мая 2001 г. в журнале «Scientific American» Тимом Бернерса-Ли, Джеймсом Хэндлером и Орой Лассила «The Semantic Web». Однако Semantic Web был задуман консорциумом W3 достаточно давно, с середины 90-х уже писались статьи и заметки, которые не привлекали внимания широкой общественности.

Для реализации Семантического Вэба необходимо разработать и внедрить множество технологий. Одной из них является технология Связанных данных (Linked Data). В 2007 году были созданы DBpedia и Freebase, в 2008 была запущена база знаний YAGO. В 2012 году запущен новый проект Фонда Викимедиа Викиданные (Wikidata). Основной целью проекта было решить некоторые аспекты, связанные с улучшением качества языковых версий Википедии. Однако стоит отметить, что проект Викиданные берет свое начало из идей Самантического Вэба, а именно реализует технологию Открытых Связанных Данных (Open Linked Data).

Основной целью данной работы является общий обзор системы Викиданные. Его можно разделить на две больших части: обзор БД Викиданные и обзор сервиса запросов к БД Викиданные.

Определены следующие задачи работы:

1. Дать описание проекта Викиданные.
2. Рассмотреть базу данных Викиданные и ее модель данных.
3. Описать функциональность встроенного сервиса запросов Викиданных.
4. Рассмотреть основные конструкции языка запросов Викиданных.
5. Подготовить примеры запросов к БД Викиданные.

Раздел 1. Общий обзор о состоянии и использовании системы Викиданные

1.1. Викиданные: определение и концепции

Викиданные (Wikidata) — это свободная, совместно наполняемая, многоязычная, вторичная база данных, в которой собрана структурированная информация для обеспечения поддержки Википедии, Викисклада, а также других вики-проектов движения Викимедия — по всему миру.

Рассмотрим более подробно существенные конструктивные решения, которые характеризуют подход, принятый в базе данных Викиданные:

- Открытое редактирование. Викиданные позволяют каждому пользователю сайта вносить и редактировать информацию в БД, даже без создания учетной записи. Интерфейс на основе форм делает редактирование понятным и простым.
- Контроль сообщества. Информация вносится и поддерживается редакторами Викиданных, которые определяют правила внесения и обработки информации. Под контролем сообщества вкладчиков находятся не только фактологические данные, но и схема данных. Авторы, редактирующие численность населения Рима, в первую очередь на их взгляд вносят самое правильное число.
- Множественность. Многие факты оспариваются, либо не определены, поэтому Викиданные позволяют противоречивым данным сосуществовать и обеспечивают механизмы для организации такого множества данных.
- Вторичные данные. Викиданные собирают факты, опубликованные в первичных источниках, вместе со ссылками на эти источники, а также их связи с другими базами данных. Викиданные не стремятся ответить на вопрос о том, насколько верно некоторое утверждение, но только лишь — соответствует ли оно информации, приведённой в источнике.
- Многоязычные данные. Большинство данных не привязаны к одному языку: цифры, даты и координаты имеют универсальное значение, а другая информация может быть переведена на любой язык. Викиданные — это многоязычный проект. Есть только один сайт Викиданные, в то время как Википедия имеет независимые издания для каждого языка, т.е. Википедия имеет множество сайтов.

- Структурированные данные. Высокая степень структурированности обеспечивает лёгкость повторного использования этих данных как проектами Викимедиа, так и внешними сервисами, и позволяет компьютерам легко обрабатывать и «понимать» такие данные.

- Свободные данные и легкий доступ. Одна из целей Викиданных — предоставлять данные другим внешним приложениям. Данные экспортируются в нескольких форматах, включая XML, JSON, RDF. Данные в Викиданных публикуются как общественное достояние по лицензии CC0, позволяющей максимальное широкое повторное использование информации самыми разными способами.

1.2. Краткая история проекта и результаты сегодня

Википедия — открытая энциклопедия знаний, где каждый человек может вносить и редактировать статьи; это один из популярнейших вэб-сайтов мира. С развитием системы накапливалось и накапливается все больше информации, включая даты, координаты и многие типы отношений от родословных деревьев до таксономии типов. Эти данные являются огромной ценностью, потому что могут быть применены в различных областях науки и культуры. Однако Википедия не обеспечивает прямого доступа к большинству этих данных, ни через сервисы запросов, ни через выгружаемый экспорт данных. По сути, данные спрятаны в огромном количестве статей, написанных к тому же на разных языках. Извлекать такие данные весьма затруднительно. Другая проблема заключается в том, что информация, касающаяся одного предмета, может появляться в статьях на разных языках и во многих статьях в пределах одного языка, но при этом она будет различаться. Численность населения Рима, например, можно найти в английской и итальянской статье о Риме, но также и в английской статье о городах Италии. Все эти цифры могут быть отличны друг от друга. Для решения всех этих проблем, Фонд Викимедиа создал новый проект Викиданные.

Викиданные были запущены 30 октября 2012 года. Тогда редакторы могли только создавать элементы (items) и соединять их со статьями Википедии. В январе 2013 года три Википедии, сначала венгерская, затем еврейская (на иврите) и итальянская, подключились к Викиданным. Между тем, сообщество уже создало более трех миллионов элементов. В феврале присоединилась английская Википедия, в марте 2013 года уже все существующие Википедии были подключены к БД Викиданные. По состоянию на февраль 2014 года Викиданные получали информацию от более чем 40 тыс. участников. Начиная с мая 2013 года, с Викиданными постоянно работали более 3.5 тыс. активных участников — это те вкладчики, которые делают, по крайней мере, пять изменений в течение месяца.

В марте 2013 года в качестве языка сценариев Википедии введен язык Lua, который может использоваться для автоматического создания и обогащения некоторых частей статьи в Википедии, например, инфобоксов. Скрипты Lua

могут получить доступ к Викиданным, позволяя редакторам Википедии извлекать, обрабатывать и отображать эти данные.

На данный момент достигнуты следующие результаты:

- централизация связей между разноязычными изданиями Википедии и другими сайтами проекта Викимедиа. К примеру, все статьи Википедии об "энциклопедии" (на любом языке) связаны с одним элементом Викиданных с идентификатором Q5292. Эти так называемые ссылки на сайты и другие данные о сущности, известной как "энциклопедия", можно посмотреть на странице https://www.wikidata.org/wiki/Q5292;
- централизация инфобоксов. Все больше и больше измененных вручную инфобоксов, таблиц с основной, фактической информацией по теме статьи, используют Викиданные в качестве базы данных серверной части, поэтому отображаемая информация будет одинакова во всех изданиях Википедии;
- обеспечение интерфейса для различных запросов. Содержание Викиданных можно запросить через открытый интерфейс SPARQL на сервисе https://query.wikidata.org. В дальнейшем результаты запроса планируется интегрировать на страницы в Википедии и других проектов, как списки, таблицы, карты и другие формы.

1.3. Система Викиданные и модель данных

Проект Викиданных расширил традиционный вики (wiki) подход: теперь пользователи не только изменяют содержимое веб-сайта через браузер, но и одновременно пополняют и редактируют базу знаний. «Вики» в переводе с гавайского языка означает «быстрый». Уорд Каннингем, создавший первую вики в 1995 году, выбрал это слово, чтобы подчеркнуть, что его содержимое его сайта можно быстро менять. Для того чтобы поддерживать вики-сайты, существует множество различных программных обеспечений, решающих различные задачи.

Одной из самых популярных систем, обеспечивающих вики, является серверное программное обеспечение МедиаВики (MediaWiki), которое позволяет обрабатывать миллионы обращений к сайту, а также обеспечивает хранение предыдущих версий. Викиданные используют МедиаВики (MediaWiki) с расширением Wikibase, которое позволяет хранить и управлять структурированными данными в центральном репозитории (хранилище), извлекать и отображать данные из репозитория на вики-страницу, а также реализует сервис запросов к хранилищу на языке SPARQL.

Таким образом, Викиданные — это централизованный репозиторий данных, доступ к которым могут получать подключённые к хранилищу сайты, и в частности вики-сайты, управляемые Фондом Викимедиа.

При упоминании в статьях «модель Викиданных» технически подразумевают модель данных репозитория Wikibase. В данной работе будет рассматриваться более упрощенная модель данных, а именно концептуальная схема, с которой работают пользователи Викиданных, когда вносят и редактируют данные на сайте.

Отдельный объект или предмет, о котором Викиданные имеют структурированные данные, называется **сущностью** (entity). В системе основными типами сущностей являются *элементы* (items) и *свойства* (properties). Каждая сущность имеет уникальный идентификатор сущности, представляющий собой число с буквенным префиксом (Q в идентификаторах элементов, P у свойств). Каждая сущность также обладает уникальным URI (*Uniform Resource*

Identifier), который следует шаблону http://www.wikidata.org/entity/ID, где ID — идентификатор сущности.

Элемент (item) соответствует какому-то объекту реального мира, концепту или событию, получившему идентификатор и сохранённому в Викиданнных вместе с информацией о нём. Каждый элемент имеет страницу, на которой пользователи могут просматривать и вводить данные. Так, например, страницу элемента Q131149 можно увидеть по ссылке: https://www.wikidata.org/wiki/Q131149 (рис. 1).

Элемент Викиданных определяется не только уникальным идентификатором или внешней ссылкой, но и уникальным сочетанием *метки* и *описания*, иными словами они уникальны для каждого элемента.

Метки (labels) — это имя или название, которое наиболее точно отражает элемент. Они необязательно должны быть уникальными, так как неоднозначность устраняется различием в *описании*.

Описания (descriptions) используются для устранения неоднозначности меток и предоставляют более подробную информацию предмете. Элементы с одной и той же меткой могут иметь различные описания, так как представляют собой различные объекты мира. Например, «статья в Википедии» и «Американский поэт, эссеист, натуралист…» — это описания для разных элементов, имеющих одну метку «Генри Дэвид Торо».

Элемент может иметь также *синонимы*, *ссылки на сайты* и *утверждения*.

Синонимы или псевдонимы (aliases) — это альтернативные имена для элемента, например, псевдоним для человека или научное имя для животного.

Ссылки на сайты (sitelinks) связывают каждый элемент с соответствующими статьями на всех вики-проектах Викимедиа.

Утверждения (statements) описывают детальные характеристики для каждого элемента. Каждое утверждение состоит из свойства (property) и его значения (value).

Каждая страница элемента отражает структуру данных, то есть содержит следующие основные части (см. рис. 1):

— название (например, “Henry David Thoreau”);

— краткое описание (например, “American poet, essayist, naturalist and abolitionist …”)

— список псевдонимов или синонимов (например, “Thoreau”);

— список утверждений (самая обширная часть данных, см. далее);

— список ссылок на сайты (ссылки на страницы Википедии и другие проекты).

Первые три части данных (название, описание, псевдонимы) известны под общим названием термины. Они в основном используются для поиска и отображения элементов. Элемент может иметь название на любом языке, поддерживаемом Викиданными. То, что отображается на страницах, зависит от настройки языка.

Рис. 1. Пример страницы Викиданных с данными об элементе Q131149

1.4. Свойства и утверждения

Свойство (property) — это сущность, которая описывает связь между элементом и *значением* свойства (*value*). Свойства, как и элементы, описаны на страницах и имеют идентификаторы, начинающиеся с «P». Свойства также имеют названия, псевдонимы, описания и утверждения, но у них нет ссылок на сайты. Также в отличие от элементов, свойства могут иметь тип данных (data type) и ограничения (constraints).

Ограничение — это правило использования конкретного свойства. К примеру, у каждого элемента, соответствующего конкретному человеку, должно быть только одно свойство date of birth (P569), так что у этого свойства есть ограничение единственного значения.

Тип данных в модели данных Wikibase — это тоже сущность, которая определяет тип и формат значений свойств. Типы данных определяют структуру значений, принимаемых свойствами. Свойство может иметь простое значение (например, строка или элемент) или комплексное значение, которое требует несколько полей, как для времени, сферических координат и количества. Существуют также специальные значения: «неизвестное значение» (*unknown value*) и «нет значения» (*no value*). Каждый тип данных должен отдельно обрабатываться программным обеспечением (например, пользовательский интерфейс будет отличаться в зависимости от типа редактируемых данных). Таким образом, типы данных, которые поддерживаются Викиданными, могут быть расширены только разработчиками программного обеспечения, а не редакторами на сайте.

Утверждения (statements) — это основное средство для представления фактической информации об элементе. Стоит отметить, что Wikibase моделируют не сам элемент, а скорее утверждения или заявления о нем. Не говорится, что в Берлине проживает 3,5 млн. человек, но есть утверждение о том, что население Берлина составляет 3,5 млн. человек по состоянию на 2011 год в соответствии с немецким статистическим ведомством.

Утверждение состоит из пары свойство-значение и может также содержать квалификаторы или уточнители (qualifiers) и ссылки (references).

Ссылка предлагает источник в поддержку того или иного утверждения, а **квалификаторы** предоставляют дополнительную контекстную информацию для данного утверждения. На рис. 2 представлено утверждение, где есть квалификаторы.

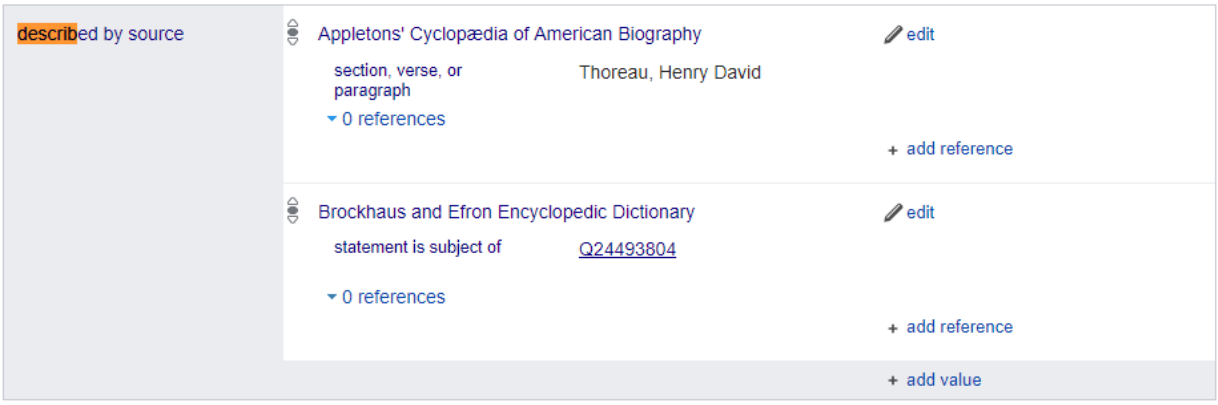

Рис. 2 Сложное утверждение о Генри Дэвиде Торо с квалификатором

В этом утверждении две основные пары свойство-значение:

- описан в источнике: Appletons' Cyclopædia of American Biography (P1343: Q12912667);
- описан в источнике: Brockhaus and Efron Encyclopedic Dictionary (P1343: 602358).

Уточнители на рис. 2 — это «раздел, стих или параграф: Thoreau, Henry David» и «тема утверждения: Q24493804». Квалификаторы представляют собой пары свойство-значение и относятся к основной части утверждения, а не к элементу на странице (Генри Дэвид Торо). Второй квалификатор менее очевидный: свойство «тема утверждения» означает «элемент, посвященный теме данного утверждения» и принимает значения типа элемент (item). Q24493804 — это элемент Викиданных, представляющий энциклопедическую статью с названием «Торо, Генри Давид» в томе XXXIIIa энциклопедического словаря Брокгауза и Ефрона.

Тройка свойство, значение и квалификаторы называется **заявлением** (**claim**). Заявление со ссылкой образует само утверждение.

Поскольку существует потенциально большое количество утверждений для элемента или свойства, необходимо выделять некоторые из утверждений от остальных, чтобы возвращать их при запросах. Например, Викиданные содержат много исторических данных с подходящими уточнителями, например, численность населения городов в разное время. Такие данные имеют множество применений, но простой запрос для населения города не должен возвращать длинный список чисел. Чтобы упростить базовую фильтрацию данных, утверждениям Викиданных можно присвоить один из трех рангов: **рангов** (**Ranks**):

- предпочтительный (preferred). Если такие утверждения существуют, то они возвращаются в ответ на запрос. Редакторы могут отметить несколько утверждений как предпочтительные, например, для обозначения разногласий или для выражения понятия наличия нескольких значений (в случае таких свойств, как «дети»).
- нормальный (normal). Этот ранг используется по умолчанию. В случае если нет предпочтительных утверждений или явно указано в запросе возвращать утверждения этого ранга, то эти утверждения возвращаются.
- нерекомендуемый (deprecated). Используется для утверждений, которые обсуждаются или считаются ошибочными, но все же перечислены для целей завершения или для предотвращения их постоянного добавления и удаления. Данные утверждения появляются в результатах, если они явно добавлены в запрос.

Раздел 2. Сервис запросов Викиданных

2.1 Доступ к данным

Есть несколько способов получения и редактирования данных из Викиданных. Можно работать с отдельными элементами или наборами данных, такими как дампы. Есть возможность скачать еженедельную копию всех данных, представленных в Викиданных, на странице *https://dumps.wikimedia.org/wikidatawiki/entities/*.

Поэлементный доступ к данным может осуществляться по уникальным унифицированным идентификаторам ресурсов URI, которые определяют элементы и свойства как сущности (concept URI). URI любого элемента или свойства получается добавлением его идентификатора к основному пространству имён Викиданных: *http://www.wikidata.org/entity/*. Здесь следует отметить разницу между самой сущностью и данными об этой сущности, которые есть в Викиданных. Если запросить *http://www.wikidata.org/entity/Q131149*, то сработает перенаправление HTTP, которое передаст клиенту *data URL*, указывающий на сведения Викиданных о Г.Д.Торо. Этот data URL будет иметь вид *http://www.wikidata.org/wiki/Special:EntityData/Q131149*.

Пространство имён Викиданных для данных о сущностях имеет вид *http://www.wikidata.org/wiki/Special:EntityData/*. Добавление к этому префиксу идентификатор сущности создает форму URL-адреса данных об этой сущности. При запросе URL Special:EntityData происходит согласование содержимого, чтобы определить формат вывода Викиданных. Таким образом, когда в веб-браузере запрашивается с помощью URI сущность из Викиданных, то ответом на запрос будет html-страница с URL-адресом *http://www.wikidata.org/wiki/Q131149*, содержащая данные о Г. Д. Торо, потому что html — удобный формат для браузера. Связанные клиентские сервисы будут получать из Викиданных данные о сущности в другом формате, например JSON или RDF, в зависимости от значения Accept: в HTTP-заголовке их запроса. Можно явно указывать формат для получения данных о сущности, дополнив URL суффиксом, указывающим на

интересующий тип содержания: .json, .rdf, .ttl или .nt. К примеру, *http://www.wikidata.org/wiki/Special:EntityData/Q131149.json* ведёт к экспорту элемента Q131149 в JSON.

Помимо этого, данные могут быть получены через MediaWiki action API — это веб-служба, которая обеспечивает доступ к некоторым вики-функциям, таким как аутентификация, операции со страницами и поиск.

В данной работе будет подробно рассмотрен доступ к данным через сервис запросов.

2.2. Сервис запросов Викиданных

Wikidata Query Service (WDQS) представляет собой пакет программного обеспечения и публичный сервис https://query.wikidata.org, предназначенный для выполнения SPARQL-запросов, позволяющий запрашивать данные из базы данных Викиданные. WDQS предоставляет публичную точку доступа SPARQL.

Точка доступа SPARQL (SPARQL-endpoint) — это служба, которая поддерживает протокол запросов SPARQL. Она позволяет пользователю делать запросы к базе знаний. Сервер обрабатывает запрос и возвращает ответ в некотором, обычно машиночитаемом, формате. Таким образом, точки доступа SPARQL в первую очередь являются API к базам знаний.

Запросы SPARQL могут быть отправлены непосредственно в точку доступа SPARQL методом GET к *https://query.wikidata.org/sparql?query*. Например:

https://query.wikidata.org/sparql?query=SELECT%20?item%20WHERE%20{?item%20wdt:P31%20wd:Q146.}

Результат возвращается в файле в формате XML по умолчанию, или JSON, если установлен либо параметр запроса format=json, либо заголовок Accept: application/sparql-results+json. Формат JSON является стандартным. В настоящее время точкой доступа SPARQL поддерживаются следующие форматы вывода результата запросов: XML, JSON, TSV, CSV, бинарный RDF.

Более удобным средством является графический интерфейс пользователя (GUI) на домашней странице http://query.wikidata.org/ позволяет редактировать и отправлять запросы SPARQL механизму выполнения запросов. Результаты отображаются в виде html-таблицы. Каждый запрос имеет уникальный URL-адрес, который можно пометить для последующего использования. Переход к этому URL вносит запрос в окно редактирования, но без его выполнения (для его выполнения нужно нажать кнопку "Выполнить").

Можно также генерировать короткий URL для текущего запроса через сервис укорачивания URL, выбрав опцию *Short URL to result* на ссылке *Link*. Также по этой ссылке имеется еще две полезные опции: SPARQL-endpoint (точка доступа SPARQL), по которой можно получить результирующий XML-файл

текущего запроса и Embed result (встроить результат), когда по полученному коду результат текущего запроса можно непосредственно вставлять в вики-разметку вновь создаваемых или редактируемых страниц приложений. Кнопка "Добавить префиксы" формирует заголовок, содержащий стандартные префиксы для SPARQL-запросов. Наиболее распространенные префиксы работают в автоматическом режиме.

При выполнении запроса в GUI можно выбрать вид представления его результатов, указав в начале запроса комментарий: #defaultView:viewName. Результаты запросов могут быть представлены в виде таблицы, карты, сетки изображений, временной шкалы, графа, линейной диаграммы, гистограммы, точечной диаграммы.

2.3. RDF

Сервис запросов к Викиданным работает на множестве данных из wikidata.org, представленных в RDF.

RDF (Resource Description Framework — Среда Описания Ресурсов) представляет собой формальный язык для описания данных. Тройка «субъект-предикат-объект» представляет собой утверждение о ресурсе и является главным элементом в языке. Ресурс — эта любая сущность как информационная, (вэб-сайт, изображение, документ в сети) так и неинформационная (человек, Земля, некое абстрактное понятие). Для обозначения субъектов, отношений и объектов в RDF используются URI. В терминах Викиданных тройка (или триплет) представляется в виде элемент-свойство-значение (item-property-value). Утверждение «Генри Дэвид Торо учился в Гарварде» состоит из субъекта «Генри Дэвид Торо» (Q131149), предиката «учился в» (P69) и объекта «Гарвард» (Q13371).

RDF сам по себе является не форматом файла, а всего лишь абстрактной моделью представления данных. Для хранения и передачи информации, уложенной в модель RDF, существует целый ряд форматов записи. Например, система запросов Викиданных работает с данными в формате RDF Dump Format, описанный на странице https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format.

Это RDF формат, который используется для передачи (экспорта) Викиданных. Соответственно, модель данных RDF Dump Format должна быть тесно связана с моделью данных Wikibase, поэтому сразу остановимся на наиболее интересных моментах.

RDF формат представляет утверждения в двух формах: правдивые и полные.

Правдивые утверждения (truthy statement) — это утверждения, имеющие лучший ранг для данного свойства. Например, «Столица США — Нью-Йорк» это правда, но в историческом контексте, поэтому это утверждение не будет считаться системой «правдивым». То есть, если есть предпочтительные утверждения для свойства, то только они будут считаться правдивыми. В

противном случае, утверждения с нормальным рангом считаются правдивыми. Предикаты для правдивых утверждений будут иметь свой префикс **wdt:**. В общем случае, правдивые утверждения можно понимать, как актуальные факты.

Утверждение может иметь дополнительную информацию, даты начала и конца и т.д. (уточнители, ссылки, ранг). Чтобы иметь доступ ко всем данным об утверждении используется полное утверждение (full statement). Полное утверждение представляет собой узел (объект) с префиксом **wds:** и идентификатором (например, wds:Q3-4cc1f2d1-490e-c9c7-4560-46c3cce05bb7). Полные утверждения связываются с сущностью и предикатом с префиксом **p:** и именем свойства (например, p:P36).

Примеры для утверждений будут рассмотрены далее.

2.4. SPARQL

SPARQL (рекурсивный акроним от англ. *SPARQL Protocol and RDF Query Language*) — это язык запросов к данным, представленным по модели RDF, а также протокол для передачи этих запросов и ответов на них.

Общая схема SPARQL-запросов имеет вид:

```
PREFIX ...
SELECT ...
WHERE {
   ...
}
```

Главным элементом языка SPARQL является шаблон тройки; это тройка, в которой на месте субъекта, предиката или объекта могут стоять переменные, состоящие из «?» и имени переменной. Если шаблоны троек объединяются в фигурные скобки, то во все переменные с одним именем подставляется одно значение.

В операторе SELECT перечисляются переменные, значения которых необходимо вернуть, а WHERE содержит ограничения на них, в основном в виде шаблона тройки.

Префиксы — это сокращения для URI. Система Викиданных автоматически распознает внутренние и наиболее распространенные внешние префиксы, например, rdf, owl, shema, поэтому нет необходимости явно указывать их. Например, утверждение о Г. Д. Торо может быть представлено с помощью трех полных URI:

```
<http://www.wikidata.org/entity/Q131149>
<http://www.wikidata.org/prop/direct/P69>
<http://www.wikidata.org/entity/Q13371>.
```

Или в более короткой форме с помощью префиксов:

```
wd:Q131149 wdt:P69 wd:Q13371.
```

«wd:» представляет элементы Викиданных (все сущности с идентификатором, начинающимся с Q).

«wdt:» — это «правдивое» свойство («truthy» property); они определяют то, что называется в Викиданных «правдивые» утверждения («truthy» statements).

В прошлой главе была затронута модель данных RDF. Для понимания, что такое правдивые и полные утверждения, разобранные в прошлой главе, напишем несколько запросов:

```
SELECT ?o
WHERE {
  wd:Q30 wdt:P36 ?o.
}
```

Результатом запроса будет wd:Q61 («Вашингтон»).

Например, мы хотим найти все столицы США. Создадим такой запрос:

```
SELECT ?o ?capital
WHERE {
  wd:Q30 p:P36 ?o.
  ?o ps:P36 ?capital.
}
```

Результат запроса приведен на рис. 3:

o	capital
wds:Q30-6f15b1c2-48be-91e1-5b2c-5d0acf61bb46	wd:Q60
wds:q30-E0257E2F-A11D-40D9-8C58-6B5A8D3AF952	wd:Q61
wds:Q30-e3cba869-48c1-dcd7-0f4d-47b7ee8b82d1	wd:Q1345

Рис. 3. Результат запроса «найти все столицы США»

Получены три полных утверждения, и с помощью префикса **ps:**, который позволяет связывать значение с утверждением, получены значения свойства «столица» для каждого утверждения. Чтобы получить значения квалификаторов утверждения используется префикс **pq:** с именем свойства. Запрос времени начала и конца для каждого утверждения представим таким образом:

```
SELECT ?o ?capital ?start ?end
WHERE {
  wd:Q30 p:P36 ?o.
  ?o ps:P36 ?capital.
  ?o pq:P580 ?start.
  ?o pq:P582 ?end.
}
```

Однако получим не совсем удовлетворяющий первоначальной идее результат:

o	capital	start	end
wds:Q30-6f15b1c2-48be-91e1-5b2c-5d0acf61bb46	wd:Q60	11 January 1785	5 December 1790
wds:Q30-e3cba869-48c1-dcd7-0f4d-47b7ee8b82d1	wd:Q1345	6 December 1790	14 May 1800

Рис. 4. Результат запроса значения квалификатора утверждения

В результат не попали сущности, у которых либо отсутствовала дата начала, либо дата конца, либо то и другое. В данном случае неопределенна дата конца для утверждения «Столица США — Вашингтон». Для того чтобы «Вашингтон» и утверждение о нем включалась в запрос необходимо использовать оператор Optional для каждой тройки, где значение может отсутствовать:

```
SELECT ?o ?capital ?start ?end
WHERE {
  wd:Q30 p:P36 ?o.
  ?o ps:P36 ?capital.
  OPTIONAL {?o pq:P580 ?start.}
  OPTIONAL {?o pq:P582 ?end.}
}
```

Получим следующий результат:

o	capital	start	end
wds:Q30-6f15b1c2-48be-91e1-5b2c-5d0acf61bb46	wd:Q60	11 January 1785	5 December 1790
wds:q30-E0257E2F-A11D-40D9-8C58-6B5A8D3AF952	wd:Q61	17 November 1800	
wds:Q30-e3cba869-48c1-dcd7-0f4d-47b7ee8b82d1	wd:Q1345	6 December 1790	14 May 1800

Рис. 5. Результат запроса с оператором Optional

Сервис запросов поддерживает некоторые расширения стандартных SPARQL возможностей. Например, сервис с URI http://wikiba.se/ontology#label позволяет легко получить метку, синонимы и описание сущности:

```
SELECT ?o ?capital ?capitalLabel ?capitalAltLabel
?capitalDescription
WHERE {
  wd:Q30 p:P36 ?o.
  ?o ps:P36 ?capital.
  SERVICE wikibase:label{
    bd:serviceParam wikibase:language "ru,en".}
}
```

При этом приоритет отдается тому языку, который указан первым:

o	capital	capitalLabel	capitalAltLabel	capitalDescription
wds:Q30-6f15b1c2-48be-91e1-5b2c-5d0acf61bb46	wd:Q60	Нью-Йорк	Большое яблоко, Нью-Йорк Сити	город в штате Нью-Йорк, США; крупнейший город США
wds:q30-E0257E2F-A11D-40D9-8C58-6B5A8D3AF952	wd:Q61	Вашингтон	Washington, DC, D.C., District of Columbia, Washington D.C., Washington DC, Washington, DC, Washington, District of Columbia	город, столица Соединённых Штатов Америки
wds:Q30-e3cba869-48c1-dcd7-0f4d-47b7ee8b82d1	wd:Q1345	Филадельфия	Philly, City of Brotherly Love, Cradle of Liberty, Philadelphia, Pennsylvania	город в штате Пенсильвания, США

Рис. 6. Получение метки, синонимов, описания сущностей с помощью службы Label Service

2.5. Расширенные шаблоны троек

В SPARQL есть синтаксические конструкции, с помощью которых можно сделать запрос короче. Напишем запрос «картины Клода Моне с изображениями»:

```
#defaultView:ImageGrid

SELECT ?item ?itemLabel ?pic
WHERE {
  ?item wdt:P31 wd:Q3305213;
        wdt:P170 wd:Q296;
        wdt:P18 ?pic.
  SERVICE wikibase:label{
    bd:serviceParam wikibase:language "ru,en".}
}
```

Комментарий указывает вид представления результата (в данном случае используется сетка изображений). На рис. 7 представлена часть результата запроса (найдено 585 картин на 05.10.2018):

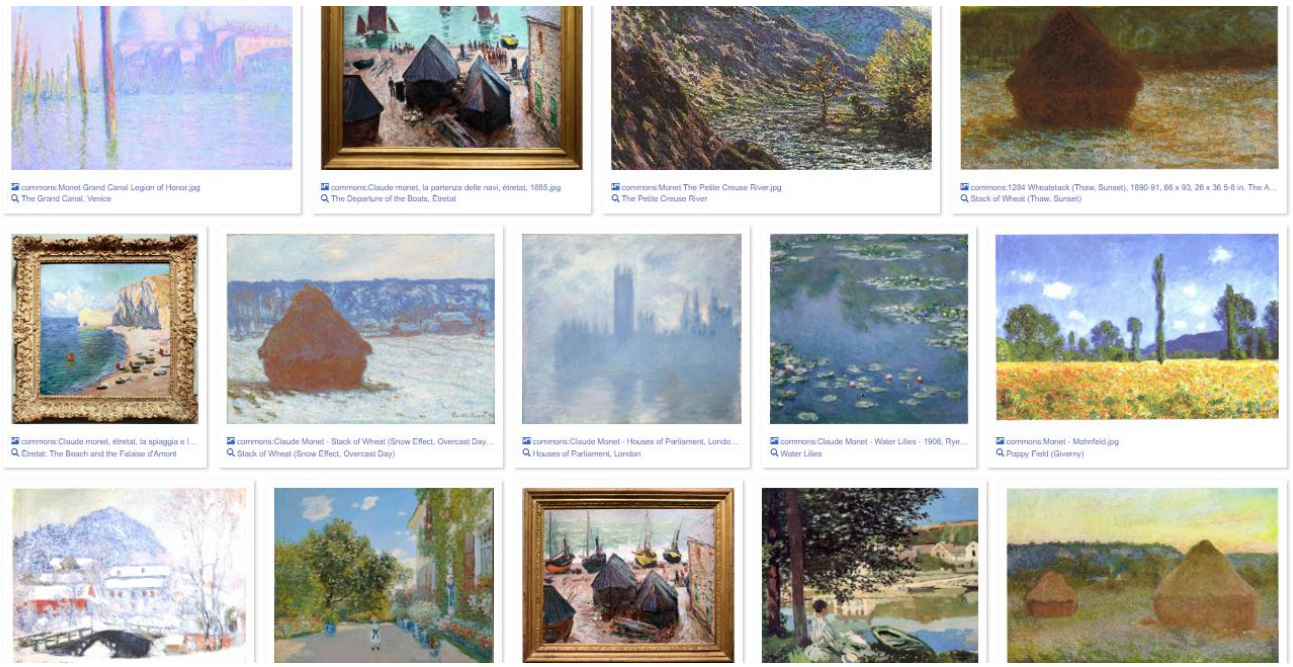

Рис. 7. Представление результатов запроса с конструкцией «;»

В запросе у нас присутствует несколько подряд идущих троек с одним субъектом. В конце тройки, если она не последняя, ставится точка с запятой, а в конце последней тройки — точка. При этом имя субъекта ставится только для первой тройки, в остальные оно подставляется автоматически из предыдущей.

Аналогично можно добавлять объекты к свойству через запятую. Например, найдем всех людей, на которых повлиял Генри Дэвид Торо и Будда:

```
SELECT ?item ?itemLabel
WHERE {
  ?item wdt:P737 wd: Q131149, Q9441.
  SERVICE wikibase:label{
    bd:serviceParam wikibase:language "ru,en".}
}
```

Получаем результат:

item	itemLabel
wd:Q1001	Махатма Ганди
wd:Q7243	Лев Николаевич Толстой

Рис. 8. Результат запроса с конструкцией «,»

Предположим нам необходимо найти все картины Моне, для которых известно, что они находятся в России. Модель запроса должна выглядеть следующим образом:

?элемент представитель Картина.

?элемент создатель Клод Моне.

?элемент располагается в ?локация.

?локация страна Россия.

Последние строчки представим так: локация, чья (которая) страна Россия. Если нас не интересуют промежуточные элементы, мы можем использовать синтаксическую конструкцию «item predicate [predicate2 item2]»:

```
SELECT ?item ?itemLabel ?pic
WHERE {
  ?item wdt:P31 wd:Q3305213;
        wdt:P170 wd:Q296;
        wdt:P276 [wdt:P17 wd:Q159].
  OPTIONAL {?item wdt:P18 ?pic.}
  SERVICE wikibase:label{
    bd:serviceParam wikibase:language "ru,en".}
}
```

Получаем 22 строчки:

item	itemLabel	pic
wd:Q19869038	Поле маков	commons:Monet, Claude - Poppy Field.jpg
wd:Q19869126	Сад в Бордигере	commons:'Garden at Bordighera, Impression of Morning' by Claude Monet, 1884, Hermitage.JPG
wd:Q681876	Дама в саду Сент-Адресс	commons:Claude Monet 022.jpg
wd:Q2939683	Carnaval Boulevard des Capucines	commons:Claude Monet 009.jpg
wd:Q3696343	Haystack at Giverny	commons:Monet, Claude - Haystack at Giverny.jpg
wd:Q3696344	Haystack near Giverny	
wd:Q3794121	Уголок сада в Монжероне	commons:Monet, Claude - Corner of the Garden at Montgeron.jpg
wd:Q3795194	Мост Ватерлоо	commons:Claude Monet - Waterloo-Brücke - 1903.jpeg
wd:Q3820703	Rouen Cathedral, End of the Day	commons:'The Rouen Cathedral at Sunset' by Claude Monet, 1894, Pushkin Museum.JPG
wd:Q3820708	Rouen Cathedral, Portal and Tower d'Albane, Midday	commons:The Rouen Cathedral at Noon by Claude Monet, 1894, Pushkin Museum.JPG
wd:Q3832426	Lilac in the Sun	commons:Lilacs in the Sun, 1872.jpg
wd:Q3877091	White Water Lilies	commons:'White Water Lilies' by Claude Monet, 1899, Pushkin Museum.JPG
wd:Q3909928	Meadows at Giverny	commons:'Meadows at Giverny' by Claude Monet, 1888, Hermitage.JPG
wd:Q3952498	Пирамиды Порт-Котон, бурное море	commons:Scogli a Belle-Île.jpg
wd:Q3952520	The Cliffs at Étretat	commons:'The Beach at Étretat' by Claude Monet, 1885-86, Pushkin Museum.jpg
wd:Q3968022	Пруд в Монжероне	commons:Monet, Pond-at-Montgeron.jpg
wd:Q10315096	Пейзаж. Сена в Аньере	commons:La Seine à Asnière - Monet.jpg
wd:Q27969146	На крутых берегах близ Дьеппа	commons:'Cliffs near Dieppe' by Claude Monet, 1897, Hermitage.JPG
wd:Q27969409	Сад	commons:Claude Monet – Garden (1876).jpg
wd:Q27970657	Женщина, сидящая в саду	commons:Monet - Femme assise dans le jardin (1876).jpg
wd:Q27975121	Сена в Руане	commons:'The Seine at Rouen' by Claude Monet, 1872, Hermitage.JPG
wd:Q27976551	Большая набережная в Гавре	commons:'The Grand Quay at Havre' by Claude Monet, 1874, Hermitage.JPG

Рис. 9. Результат запроса, с конструкцией «[]»

2.6. Пути свойств

Пути свойств — это способ коротко записать путь свойств между двумя элементами. В прошлом параграфе были приведены запросы, где использовалось свойство P31, которое можно описать как конкретный объект, представитель, экземпляр класса или категории. «Унесенные призраками» это представитель понятия «анимационный фильм». «Анимационный фильм» это подкласс для понятия «фильм», который в свою очередь подкласс для концепта «произведение искусства», которое есть подкласс …

Таким образом, невозможно напрямую получить все произведения искусства (конкретные экземпляры), сформулировав запрос так:

```
?work wdt:P31 wd:Q838948. # instance of work of art.
```

Будет получено всего лишь 2826 строк, что очень мало для всех произведений искусства. Решением будет использовать конструкцию вида:

```
?item wdt:P31/wdt:P279* class.
```

Это значит, что находятся все элементы «представители» какого-либо (или никакого) «подкласса» данного класса.

```
SELECT ?work ?workLabel
WHERE {
  ?work wdt:P31/wdt:P279* wd:Q838948.
  SERVICE wikibase:label { bd:serviceParam wikibase:language
"[AUTO_LANGUAGE]". }
}
```

Данный запрос превышает стандартное время для выполнения запроса (60 с.). Для того, чтобы избежать превышения времени, можно ограничить количество выдаваемых результатов с помощью оператора LIMIT (Например, LIMIT 555).

Вместо «*» можно использовать «+», тогда в результат не попадут элементы, которые удовлетворяют «instance of work of art». Иными словами «*» означает «0 или сколько угодно», а «+» — (1 или сколько угодно).

Если разделить элементы пути вертикальной полосой «|», то используется одно из свойств, разделенных чертой. К тому же, с помощью «()» можно комбинировать все эти знаки, например, путь (wdt:P22|wdt:P25)+ может содержать двух матерей и 1 отца или 4 отцов, или мать-отец-отец-мать и все возможные такие комбинации.

2.6. Обзор основных операторов и функций

Рассмотрим важные составляющие базовые элементы языка, не затронутые ранее.

Для создания выражений используются математические операторы (+, –, *, ÷), операторы сравнения (<, >, =, <=, >=, !=) и логические операторы (&&, | |).

Краткий обзор агрегатных функций (производят одиночное значение для всей группы таблицы):

- COUNT: количество элементов. COUNT(*): подсчет всех результатов.
- SUM, AVG: суммирование и взятие среднего значения элементов;
- MIN, MAX: минимум и максимум значений элементов;
- DISTINCT; устранение дубликатов в результате. Такая необходимость часто возникает с конструкцией ?item wdt:P31/wdt:P279* ?class, когда от субъекта к объекту существует несколько путей, поэтому получается новый результат для каждого пути, хотя значения идентичны (можно использовать SELECT DISTINCT);

Остальные функции будут рассмотрены вместе с запросами:

1. Запрос карты картин Винсента ван Гога и подсчет количества картин в каждой локации:

```
#defaultView:Map
SELECT ?locationLabel ?coord (COUNT(?painting) as ?count)
WHERE {
  ?painting wdt:P31 wd:Q3305213;
            wdt:P170 wd:Q5582;
            wdt:P276 ?location.
  ?location wdt:P625 ?coord.
  SERVICE wikibase:label { bd:serviceParam wikibase:language
"en" }
}
GROUP BY ?locationLabel ?coord
ORDER BY DESC(?count)
```

ORDER BY: сортирует результаты по выбранному столбцу. По умолчанию стоит сортировка по возрастанию ASC(). Здесь используется DESC(), для сортировки значений в столбце count по убыванию. Локацией с наибольшим количеством картин является Музей Винсента ван Гога в Амстердаме (215 картин).

GROUP BY: оператор, определяющий, как строки будут группироваться при использовании в запросе агрегатных функций, таких как sum, max, min, count и других. Стоит учитывать, что любой столбец, который используется в выражении SELECT (не считая столбцов, которые хранят результат агрегатных функций), должны быть указаны после GROUP BY. Все выходные строки запроса разбиваются на группы, характеризуемые одинаковыми комбинациями значений в этих столбцах. Результат запроса 140 локаций. Результат запроса отражен с помощью карты:

Рис. 10. Отображение результат запроса на карте

2. Напишем запрос для нахождения всех людей, родившихся в 2018 году:

```
SELECT ?person ?personLabel ?date
WHERE
{
  ?person wdt:P31 wd:Q5;
          wdt:P569 ?date.
  FILTER("2018-01-01"^^xsd:dateTime <= ?dob && ?date < "2019-
01-01"^^xsd:dateTime).
  SERVICE wikibase:label { bd:serviceParam wikibase:language
"[AUTO_LANGUAGE]". }
}
```

FILTER: фильтрация, отбор результатов по какому-либо условию. Здесь также приведен один из способов приведения к типу Дата-Время.

3. Запрос «у кого сегодня день рождения»

```
SELECT ?entityLabel (YEAR(?date) as ?year)
WHERE
{
    BIND(MONTH(NOW()) AS ?nowMonth)
    BIND(DAY(NOW()) AS ?nowDay)

    ?entity wdt:P569 ?date .
    FILTER (MONTH(?date) = ?nowMonth && DAY(?date) = ?nowDay)
    SERVICE wikibase:label {
        bd:serviceParam wikibase:language "en" .
    }
}
```

BIND в общем случае используется с конструкцией `BIND(expression AS ?variable)`. В таком случае выражение присваивается новой или переопределяется значение уже существующей переменной. В этом запросе использованы функции NOW, которая возвращает текущее время, а также

MONTH и DAY, которые берут от даты месяц и число соответственно. Можно рассмотреть такие операторы как BOUND и IF:

BOUND(?var): проверка на то, была ли переменная привязана к значению. Часто используется с OPTIONAL.

IF (condition, thenExpression, elseExpression): если условие оценивается как true, то дальше вычисляется thenExpression, если false, то переход на elseExpression.

Заключение

База Викиданные, содержание и ее основное программное обеспечение находятся в стадии постоянного развития, исход которого трудно предвидеть. Учитывая важную роль, которую играет база Викиданные для Википедии, можно быть уверенным в том, что этот проект будет продолжать расти по размеру и качеству. Помимо этого открываются возможности использования не только в обеспечении проектов Викимедии, но и проект может принести огромной вклад в развитие идей Семантического Вэба.

В данной работе был дан общий обзор системы Викиданные. В обзоре не рассматривались технические моменты, но были подробно описаны некоторые аспекты, связанные с моделью данных системы. Раздел обзора сервиса запросов, языка запросов и основных конструкций не является полным, а лишь пытается продемонстрировать возможности системы Викиданные.

Список использованных источников и литературы

1. Denny Vrandečić, Markus Krötzsch Wikidata: A Free Collaborative Knowledgebase. Communications of the ACM, October 2014, Vol. 57 No. 10, Pages 78-85 [Электронный ресурс]. URL: https://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext.
2. Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez and Denny Vrandeciˇ Introducing Wikidata to the Linked Data Web. The Semantic Web – ISWC 2014. Lecture Notes in Computer Science. Vol. 8796. P. 50–65 [Электронный ресурс]. URL: https://iccl.inf.tu-dresden.de/w/images/3/3a/Wikidata-RDF-export-2014.pdf
3. Викиданные: Введение [Электронный ресурс]. URL: https://www.wikidata.org/wiki/Wikidata:Introduction/ru.
4. Викиданные: Доступ к данным [Электронный ресурс]. URL: https://www.wikidata.org/wiki/Wikidata:Data_access/ru.
5. Wikidata: SPARQL tutorial [Электронный ресурс]. URL: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial.
6. Wikidata: SPARQL query service/queries/examples [Электронный ресурс]. URL: https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/queries/examples.
7. Wikidata Query Service/User Manual [Электронный ресурс]. URL: https://www.mediawiki.org/wiki/Wikidata_Query_Service/User_Manual.
8. Wikibase/Indexing/RDF Dump Format [Электронный ресурс]. URL: https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format.
9. Викиданные: Глоссарий [Электронный ресурс]. URL: https://www.wikidata.org/wiki/Wikidata:Glossary/ru.
10. Wikibase/DataModel [Электронный ресурс]. URL: https://www.mediawiki.org/wiki/Wikibase/DataModel.

11. Wikibase/DataModel/Primer [Электронный ресурс]. URL: https://www.mediawiki.org/wiki/Wikibase/DataModel/Primer.
12. Wikibase [Электронный ресурс]. URL: https://www.mediawiki.org/wiki/Wikibase.