ПРОГРАММИРОВАНИЕ ВИКИДАННЫХ

Учебное электронное пособие

Под редакцией А. А. Крижановского

Петрозаводск
2024

Министерство науки и высшего образования Российской Федерации
Федеральное бюджетное государственное образовательное
учреждение высшего образования
ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПРОГРАММИРОВАНИЕ ВИКИДАННЫХ

Учебное электронное пособие

Под редакцией А. А. Крижановского

Петрозаводск
Издательство ПетрГУ
2024

ISBN 978-5-8021-4154-0

УДК 004
ББК 32.973

Авторы:
А. А. Крижановский, М. С. Балакирева, Е. А. Меньшикова,
Е. О. Паренченков, А. С. Потес, Е. Д. Трубина, А. Обрегон Сьерра

Издаётся по решению редакционно-издательского совета
Петрозаводского государственного университета

Рецензенты:
А. А. Рогов, доктор технических наук, профессор;
Р. Н. Кербуш, председатель РОО «Викимедиа Санкт-Петербург»

Программирование Викиданных : учебное электронное пособие / А. А. Крижановский, М. С. Балакирева, Е. А. Меньшикова [и др.] ; под ред. А. А. Крижановского ; М-во науки и высш. образования Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. образования Петрозав. гос. ун-т. – Электрон. дан. – Петрозаводск : Издательство ПетрГУ, 2024. – 1 CD-ROM. – Систем. требования : PC, MAC с процессором Intel 1,3 ГГц и выше ; Microsoft Windows, MAC OSX ; 256 Мб (RAM); Adobe Reader ; дисковод CD-ROM. – Загл. с титул. экрана. – Текст, изображение : электронные.

П784

ISBN 978-5-8021-4154-0

Викиданные – это огромная база данных, лежащая в основе Википедии и связывающая вики-проекты воедино. Ознакомившись с пособием, вы сможете извлекать из Викиданных информацию с помощью SPARQL-скриптов, затем обрабатывать её и строить по ней таблицы, графики и карты. Пособие распространяется на правах свободной лицензии Creative Commons Attribution-ShareAlike 4.0.

Издание адресовано студентам, изучающим информатику, а также тем, кто хочет развить свои аналитические навыки, научиться программировать и глубже разбираться в принципах работы самого компьютерного вики-проекта.

УДК 004
ББК 32.973

Учебное электронное издание
Минимальные системные требования:
PC, MAC с процессором Intel 1,3 ГГц и выше;
Microsoft Windows, MAC OSX;
256 Мб (RAM); Adobe Reader; дисковод CD-ROM

Учебное электронное издание

Крижановский Андрей Анатольевич, **Балакирева** Мария Сергеевна,
Меньшикова Екатерина Андреевна, **Паренченков** Евгений Олегович,
Потес Артём Сергеевич, **Трубина** Елизавета Денисовна, **Обрегон Сьерра** Анхель

ПРОГРАММИРОВАНИЕ ВИКИДАННЫХ

Учебное электронное пособие

Под редакцией А. А. Крижановского

Редактор *И. И. Куроптева*
Художественный редактор *Е. В. Лавренова*

Подписано к использованию 20.02.2024.
1 CD-R. 30 Мб. Тираж 100 экз. Изд. № 56

Федеральное государственное бюджетное образовательное
учреждение высшего образования
ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
185910, г. Петрозаводск, пр. Ленина, 33
https://petrsu.ru
Тел.: (8142) 71-10-01

Изготовлено в Издательстве ПетрГУ
185910, г. Петрозаводск, пр. Ленина, 33
URL: press.petrsu.ru/UNIPRESS/UNIPRESS.html
Тел./факс: (8142) 78-15-40
nvpahomova@yandex.ru

Оглавление

Предисловие 11

Часть I. Основы Викиданных. 14

1. Программы и программки 15

2. Обзор Викиданных 16
Викиданные 16
Wikidata Query Service 16
Об исследовании Викиданных 17
Неоднозначность объекта Викиданных 18
Качество и техническая платформа Викиданных 19

3. Корзины и мячи 20

Часть II. Исследуем объекты Викиданных 26

4. Воздушные суда и их производители 27
Список самолётов 27
Производители воздушных судов 28
Количество выпущенных воздушных судов 29
В каких странах производят самолёты 31
Полнота Викиданных по числу производителей воздушных судов 32
Упражнения 32

5. Аниме: загадочный и поразительный мир японской анимации . . . **33**
Экземпляры объекта «Аниме» . . . 33
Список сэйю, упорядоченный по числу озвученных ими аниме . . . 35
График по числу сэйю, озвучивших одно и более аниме . . . 36
Граф, связывающий сэйю и озвученные ими аниме . . . 38
Полнота Викиданных по числу аниме и актёров . . . 40
Указана ли дата публикации у аниме? . . . 41
Анализ возраста, в котором сэйю озвучивают аниме . . . 42
Упражнения . . . 46

6. В каких населённых пунктах России больше рождаётся учёных, в сельских или городских? . . . **47**
Список «населённых пунктов» . . . 48
Список стран по суммарному количеству населения . . . 49
Полнота Викиданных по заполненности свойств населённых пунктов . . . 50
Доля населения страны, проживающего в «населённых пунктах» . . . 50
Список классов, сопутствующих «населённому пункту» в свойстве «экземпляр» . . . 54
Отечественные учёные на селе и в городе . . . 56
Список сельских и список городских типов поселений в России . . . 57
Профессии, представленные в Викиданных . . . 59
Найти учёных среди людей . . . 60
Построение диаграммы с научными направлениями и годами рождения учёных, родившихся в сельских поселениях . . . 61
Построение диаграммы числа учёных, родившихся в городских поселениях, и сравнение диаграмм . . . 66

7. От малых городов до городов-миллионеров . . . **68**
Списки разных типов городов . . . 68
Численность населения . . . 70
Города-побратимы . . . 73
Сколько городов обходятся без городов-побратимов? . . . 73
Упорядоченный список городов по числу побратимов . . . 74
Число городов с определённым числом побратимов . . . 75
У какой страны больше всего побратимов? . . . 77
Ближайшие соседи России по числу городов-побратимов . . . 78
Полнота и недостатки Викиданных: дублирование объектов . . . 80
Упражнения . . . 81

8. Анализ стран: возраст, формы правления и этнохоронимы . . . 82
Список стран и степень полноты информации ряда стран . . . 82
Возраст стран и почему Россия бывает не страной? О конструкции p:/ps: . . . 83
Полнота Викиданных по современным и историческим странам . . . 85
Этнохоронимы стран на русском языке . . . 86
Список этнохоронимов стран . . . 87
Страны с незаполненными этнохоронимами . . . 87
Количество заполненных этнохоронимов у стран . . . 88
Формы правления стран . . . 89
Граф и карта соседних стран . . . 91
Упражнения . . . 92

9. Регионы России . . . 93
Экземпляры объекта «Области России» . . . 93
Субъекты Российской Федерации . . . 94
Соседние субъекты . . . 95
Полнота Викиданных по числу субъектов, имеющих границу с кем-либо . . . 101
Численность населения по субъектам Российской Федерации . . . 102

10. На каких языках программирования пишутся операционные системы . . . 104
Список операционных систем . . . 104
Предшественники операционных систем и даты выпуска . . . 105
Количество операционных систем, написанных на языках программирования . . . 106
Полнота Викиданных по числу операционных систем . . . 109
Языки программирования, используемые для написания операционных систем . . . 109
Количество программ для каждой операционной системы . . . 109
Сколько компьютерных программ было написано для операционной системы с использованием того или иного языка программирования . . . 110
Документирование программного обеспечения . . . 112
Упражнения . . . 112

11. Где учатся и кем работают изобретатели языков программирования . . . 113
Список языков программирования . . . 114
Операции над множествами в SPARQL . . . 115
Пермиссивные лицензии . . . 116
Количество форматов файлов исходного кода . . . 117

Страны, в которых живут люди и располагаются организации, связанные с созданием языков программирования . . . 119
Университеты, в которых учились разработчики языков программирования . . . 122
Профессии создателей языков программирования . . . 123
Объектно-ориентированные языки программирования . . . 124
Степень заполненности имён разработчиков языков программирования на русском языке . . . 125
Упражнения . . . 126

12. Военные корабли и их операторы . . . 127
Список кораблей . . . 127
Полнота Викиданных по числу кораблей . . . 129
Полнота свойств объектов военных кораблей . . . 129
Корабли-музеи в странах мира . . . 132
Упражнения . . . 134

13. Космические корабли и станции: современные реалии 50-летней давности . . . 136
Список космических кораблей и станций . . . 136
Глубина проработки объектов . . . 137
Список отечественных кораблей и станций . . . 137
Полнота Викиданных по числу отечественных космических кораблей . . . 138
Временные графики освоения космоса в нашей стране и в мире . . . 138
Космонавты в международных полётах . . . 142
Упражнения . . . 142

Часть III. Специальные возможности вики-проектов и Викиданных . . . 145

14. Сервис балансировки ProWD . . . 146

15. Защита страниц . . . 147

16. Боты в Викиданных . . . 149
Требования . . . 149
Наши первые скрипты . . . 150
SPARQL-запросы внутри программы на Python . . . 154
Изменение значений, введённых в Викиданные . . . 156

Заключительная часть 160

Ответы 161

Список литературы 183

Ключевые слова и понятия 186

Что может быть проще примитивного нуль-передатчика? Только примитивный нуль-аккумулятор.

Обитаемый остров
Аркадий и Борис Стругацкие

Эта машина, подпиравшая стены железными боками, была огромна. Электроник, поглощая цифры, скоро пришёл к выводу, что ему нужно сидеть здесь несколько суток. . .
— Я всего этого не запомню, — скрипуче сказал Электроник. — Миллионы цифр! Слишком большое количество информации. . .
Профессор положил руку на его плечо.
— Отключись, — посоветовал он. — Привыкать надо постепенно.

Рэсси — неуловимый друг
Евгений Велтистов

Издание посвящается изобретателям SPARQL-скриптов — программистам будущего, а также OSINT-разведчикам, для которых Викиданные будут школьным полигоном

Предисловие

Это издание адресовано не только студентам, но и школьникам, изучающим информатику. Оно поможет тем, кто хочет развить свои аналитические навыки, научиться программировать и глубже разбираться в принципах работы самого компьютерного вики-проекта. Вы познакомитесь с работой базы Викиданных и языком запросов SPARQL. Прочитав пособие, вы сможете извлекать из Викиданных информацию с помощью SPARQL-скриптов, затем обрабатывать её и строить по ней таблицы, графики и карты.

Викиданные — это искусно сделанная база данных, которая, как огромный кит, лежит в основе громадной планеты Википедии. Впечатляет скорость роста этого кита, во многих главах мы будем обращать на это внимание. Если изначально проект Викиданные создавался для обслуживания нужд Википедии, то сейчас он применяется крайне широко и в самых разных целях.

В первой части пособия (Основы Викиданных) объясняется, что такое запросы и листинги программ, как ими пользоваться, даны базовые понятия об объектах Викиданных. Глава «Обзор Викиданных» включает историческую справку, вопрос качества данных и связи с Википедией. Описан сервис WDQS, к которому мы будем обращаться на каждой странице. Также включён небольшой обзор научных исследований, связанных с Викиданными. Глава «Корзины и мячи» с помощью образов и аналогий позволяет подступиться к Викиданным тем, кто хочет научиться программировать.

Во второй и основной части издания (Исследуем объекты Викиданных) каждая глава — это небольшое приключение, где один из объектов Викиданных является главным героем. С помощью SPARQL-запросов мы пытаемся расспросить Викиданные об этом герое, с помощью графиков, таблиц, диаграмм — проанализировать и нарисовать его образ.

В третьей части пособия (Специальные возможности вики-проектов и Викиданных) рассмотрен вопрос защиты страниц в вики-проектах, рассказано о сервисе балансировки ProWD и о том, как можно последовательно двигаться от SPARQL-запросов к языку Python при создании компьютерных программ (ботов), редактирующих Викиданные.

В заключительной части даны ответы на вопросы, рассеянные по всему пособию, приведены список литературы и индекс ключевых слов для удобного постраничного поиска.

В исследовании объектов Викиданных приняли участие студенты ПетрГУ в рамках курса «Программирование Викиданных». Материалы этого онлайн-курса доступны на сайте Викиверситета[1]. Если у вас будут вопросы, замечания или пожелания к этому пособию, то пишите их на странице обсуждения курса[2].

Курс и пособие разрабатывались и писались с 2017 года[3]. За семь лет часть сайтов, на которые мы ссылались, канула в лету. Копии таких утраченных сайтов вы можете найти в Архиве Интернета (archive.org).

Потребовалось несколько лет работы со студентами в Википедии, прежде чем мы пришли с ними к таким проектам, как Викиверситет и Викиданные. Результатом предыдущей работы в Википедии стало учебное пособие для тех, кто хочет научиться редактировать мировую энциклопедию[4]. Это пособие доступно онлайн[5].

Будет ли в этом издании рассказано, как редактировать и пополнять Викиданные новой информацией? — Нет, хотя это не сложнее, чем писать текст в SMS или статью в Википедии. Здесь мы покажем вам, как увлекательно формулировать вопросы на языке Викиданных.

Почему мы занялись и увлеклись Викиданными? — Потому что это самая большая, сложная и быстрорастущая база данных на Земле. Потому что эта база лежит в основе Википедии, которую может редактировать каждый. И Викиданные тоже может редактировать каждый. Часто редакторы родственных проектов (Википедия, Викисклад, Викисловарь) заглядывают в Викиданные, чтобы там что-то поправить, изменить. Работу серверов Википедии и Викиданных обеспечивают сотрудники организации Викимедиа.

Почему мы адресуем издание не только студентам, но и школьникам? — Потому что программирование и программирование Викиданных в частности — это увлекательная и, если увлечься, простая вещь. Если с помощью учителя или самостоятельно школьник увлечётся, то сможет программировать Викиданные, многие ребята плодотворно редактируют статьи Википедии. Надеемся, что благодаря этому пособию одним из первых компьютерных языков, изучаемых в школе, станет язык Викиданных.

[1] URL: https://ru.wikiversity.org/wiki/Программирование Викиданных.

[2] URL: https://ru.wikiversity.org/wiki/Обсуждение:Программирование Викиданных.

[3] Это пособие написано в LaTeX. Исходный код доступен по ссылке: https://github.com/componavt/wd_book. Возьмите его и напишите свою книгу.

[4] *Крижановский А. А.* Работа в вики-среде на примере Русской Википедии. Ч. 1 : учеб. пособие. Петрозаводск : ПетрГУ, 2015. URL: `https://commons.wikimedia.org/?curid=86907665`.

[5] URL: https://commons.wikimedia.org/?curid=86907665.

Покажем прямо сейчас, что Викиданные находятся от нас на расстоянии одного клика. Откройте на компьютере или на телефоне ссылку: https://w.wiki/4cXU. Вы увидите главное окно, в котором мы с вами будем писать наши небольшие программы. Если вы нажмёте большую синюю кнопку с белым треугольником (рис. 1), то запустите эту программу из семи строк. Программа обратится к базе данных и спросит, какие столицы есть в Викиданных. Результатом будет список столиц. Подробнее о городах в Викиданных читайте в разделе «От малых городов до городов-миллионеров» на с. 68. Вот так просто можно запускать SPARQL-программы: достаточно браузера, Интернета и гиперссылки с запросом к Викиданным.

Дадим читателю подсказку относительно тех изображений, которые кажутся слишком мелкими. Поскольку это электронное издание, то вы можете воспользоваться зумом и увеличить изображение. Как раз для этого мы старались включать сюда изображения с максимально большим разрешением.

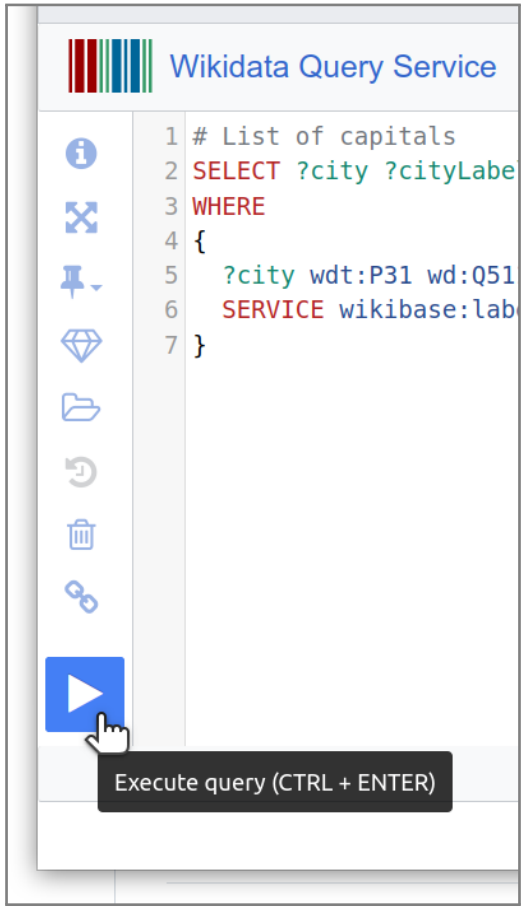


Рис. 1. Выполнение скрипта в сервисе Wikidata Query Service

Часть I

Основы Викиданных

1. Программы и программки

В этом пособии приводится *программный код*[1] на языке SPARQL. Именно на этом языке пишут запросы к Викиданным.

Вот пример SPARQL-скрипта (запрос 1.1), с помощью которого можно получить из Викиданных список городов, точнее, экземпляров объекта city (Q515):

Запрос 1.1. Список городов[2]

```
SELECT ?city ?cityLabel WHERE {
  ?city wdt:P31 wd:Q515.       # instance of city
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

Мы будем регулярно ссылаться на объекты Викиданных. Например, city (Q515) — это имя объекта Викиданных. Здесь *city* — имя метки объекта (Label), а *Q515* — уникальный идентификатор объекта в базе Викиданных. При этом идентификатор является частью названия страницы сайта Викиданных с описанием этого объекта: https://www.wikidata.org/wiki/Q515.

Объекты Викиданных нумеруются последовательно: чем раньше объект создан в Викиданных, тем меньше его номер. Так сложилось, что обычно объекты, имеющие меньший номер, являются более значимыми. Например, *Солнце* имеет идентификатор Q525, а объект *мушка дрозофилы* — Q312154.

[1] Программный код также называют *исходным кодом* или *листингом*.

Что такое *экземпляр объекта*? Какая разница между экземпляром объекта в объектно-ориентированном программировании и в Викиданных? См. ответ на с. 161.

[2] Получено: 20 800 городов в 2017 году, 9260 городов в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/jcE.

2. Обзор Викиданных

Викиданные

Викиданные — это структурированная и совместно редактируемая база данных[1]. Проект был официально запущен 30 октября 2012 года, его разработка ведётся под руководством Wikimedia Deutschland[2]. Проект создавался за счёт пожертвований Allen Institute for Artificial Intelligence, Gordon and Betty Moore Foundation и Google. Викиданные — это бесплатная и свободная база знаний, которая может использоваться и редактироваться людьми и компьютерными программами[3].

Содержимое Викиданных распространяется по лицензии Creative Commons CC0, которая позволяет повторно использовать информацию самыми разными способами: пользователи могут копировать, изменять, распространять и обрабатывать эти данные в любых целях. Ещё одна особенность Викиданных — это многоязычность. Любой человек может редактировать Викиданные более чем на 350 языках.

Викиданные постоянно обновляются, добавляются новые объекты. На 2023 год насчитывается более 100 млн страниц и около 2 млрд правок (данные взяты с официальной страницы статистики Викиданных: https://www.wikidata.org/wiki/Wikidata:Statistics). Уже в 2019 году на сайте Викиданных было совершено более 800 тыс. правок, что превзошло количество правок в Английской Википедии и сделало Викиданные наиболее редактируемым сайтом Викимедиа. Веб-сайт Викиданных, к которому мы будем регулярно обращаться, такой: https://www.wikidata.org.

Wikidata Query Service

Любой объект Викиданных имеет свой уникальный идентификатор и свойства. Эта информация может быть обработана с помощью компьютера, и при этом она наглядно представлена и понятна

[1] Викиданные (Wikidata) — это свободная, совместно наполняемая, многоязычная, вторичная база данных, в которой собрана структурированная информация для поддержки работы Википедии, Викисклада и других проектов Викимедиа.

[2] Немецкое отделение Фонда Викимедиа.

[3] *Vrandečić D., Krötzsch M.* Wikidata: A Free Collaborative Knowledgebase // Communications of the ACM. 2014. Vol. 57, no. 10. P. 78–85. ISSN 0001-0782.

Рис. 2.1. Логотип Викиданных. Wikimedia Commons / Planemad

пользователям без специальной предобработки. Сайт Викиданных содержит сервис Wikidata Query[4], включающий набор инструментов для построения SPARQL-запросов и их визуализации в виде таблиц, диаграмм, графов или географических карт.

[4] Полное название инструмента Wikidata Query Service, кратко — WDQS. URL: https://query.wikidata.org/.

Об исследовании Викиданных

В работе *A large-scale collaborative ontological medical database*[5] описываются плюсы использования Викиданных для создания крупномасштабной совместно используемой медицинской базы данных. Основные требования к создаваемой базе данных таковы: это должна быть платформа с обновлением в реальном времени, с лицензией, разрешающей дальнейшее использование полученной информации, с возможностью редактирования на любом языке и с открытым доступом. Именно это и есть основные характеристики Викиданных. Во-первых, Викиданные — это открытая, редактируемая база знаний. Любой пользователь без навыков программирования может вносить изменения более чем на 350 языках. Во-вторых, информация постоянно обновляется, добавляются новые объекты. На 2023 год Викиданные насчитывают около 25 тыс. редакторов[6]. В-третьих, лицензия Creative Commons CC0 обеспечивает широкое использование полученной информации.

[5] Wikidata: A large-scale collaborative ontological medical database / H. Turki [et al.] // Journal of Biomedical Informatics. 2019. Vol. 99. P. 1–13. ISSN 1532-0464. URL: https://www.sciencedirect.com/science/article/pii/S1532046419302114.

[6] Для сравнения: число активных редакторов в Русской Википедии составляет 11 тыс., в Английской — 130 тыс., на Викискладе — 40 тыс. Поясним термин *активный редактор* — это такой пользователь сайта, который сделал хотя бы одну правку за последние 30 дней.

Ввиду этих преимуществ у Викиданных сейчас нет конкурентов. Но принято указывать аналоги и альтернативы. Укажем и мы. Есть несколько альтернативных баз знаний:

1. Cyc — проект компании Cycorp (Остин, США) по созданию онтологической базы знаний, позволяющий решать задачи из области искусственного интеллекта. База Cyc имеет исследовательскую лицензию ResearchCyc. У этой базы есть некоторые недостатки: сложность системы (сложность добавления данных вручную), недостаток документации, неполнота системы.

2. Evi (ранее True Knowledge) — технологическая компания в Кембридже (Англия), которая специализируется на базе знаний и программном обеспечении *семантического поиска*[7]. Добавление информации в базу знаний осуществляется двумя способами: импорт из «заслуживающих доверия» внешних баз данных (например, Википедия) и добавление данных самими пользователями. Как и в Википедии, пользователь может изменять данные, «соглашаться» или «не соглашаться» с информацией, представленной системой Evi. Система может отклонить любые факты, которые семантически несовместимы с другими утверждениями, в отличие от Викиданных, где могут храниться противоречивые данные.

3. DBPedia — краудсорсинговый проект, направленный на извлечение структурированной информации из данных, созданных в рамках проекта Википедия, и публикации её в виде доступных под

[7] Семантический поиск — это способ и технология поиска информации с использованием контекстного значения запрашиваемых фраз вместо словарных значений отдельных слов или выражений, входящих в поисковый запрос.

свободной лицензией наборов данных. Проект был отмечен как один из наиболее известных примеров реализации концепции связанных данных[8]. Он был начат группой добровольцев из Свободного университета Берлина и Лейпцигского университета в сотрудничестве с фирмой OpenLink Software, первый набор данных опубликован в 2007 году. С 2012 года активным участником проекта является Университет Мангейма.

В Викиданных информация представлена в виде объектов (или элементов), связанных между собой с помощью свойств[9]. Мощь базы Викиданных в её большом объёме и в удивительно быстром росте и самоорганизации, в том, что к этой «живой» базе знаний можно обращаться с помощью SPARQL-запросов, представлять результаты их выполнения в виде таблиц, графов, диаграмм или сохранять в нужном формате (CSV, JSON, SVG).

Викиданные могут взять на себя роль централизованного хранилища данных. В статье[10] приводится пример использования Викиданных в качестве централизованной и общедоступной базы знаний для системы FALCON 2.0. Эта система идентифицирует сущности в коротком тексте или вопросе, а затем связывает их ссылками с соответствующими объектами Викиданных.

Неоднозначность объекта Викиданных

Любой объект Викиданных имеет свойства. Одно из них — это «экземпляр класса» (instance of (P31)). Оно определяет класс, к которому принадлежит объект. Мы обнаружили, что один объект Викиданных может соответствовать нескольким классам. Некоторые объекты являются экземплярами совершенно разных классов. Например, Королевская шведская академия наук (Q191583) является экземпляром сразу трёх классов: академии наук, сооружения (здания) и королевской академии Швеции. Такое определение классов верно, поскольку этот объект можно рассматривать и как организацию, целью которой является развитие науки, и как архитектурное сооружение. Мы рассмотрели пример многозначности в Викиданных, которая разрешается с помощью свойства instance of.

В Википедии принято иначе оформлять многозначность. Если слово имеет несколько значений, то есть если слово многозначное, то в Википедии есть несколько статей с одинаковым названием об этих значениях, но в конце названия в скобках пишется уточнение, например: «Коса (причёска)» и «Коса (рельеф)». Такое чёткое явное разнесение многозначности слов позволило использовать тексты Википедии в задаче разрешения лексической многозначности или WSD-задаче[11].

[8] О связанных данных и графах знаний см. в главе «Корзины и мячи» на с. 20.

[9] Например, существуют такие свойства: экземпляр (P31), подкласс (P279), часть (P361), имеет часть (P527).

[10] Falcon 2.0: An entity and relation linking tool over Wikidata / A. Sakor [et al.] // Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020. P. 3141–3148. URL: https://arxiv.org/pdf/1912.11270.pdf.

[11] *Fogarolli A.* Word Sense Disambiguation Based on Wikipedia Link Structure // Semantic Computing, 2009. ICSC '09. IEEE International Conference on. 2009. P. 77–82. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.1696&rep=rep1&type=pdf.

Качество и техническая платформа Викиданных

Викиданные существуют с 2012 года. На 2023 год в Викиданных зарегистрировано 6 млн пользователей, которые сделали около 2 млрд правок.

В диссертации Alessandro Piscopo[12] идёт речь о социально-технических процессах и качестве данных проекта «Викиданные», о том, что пользователи Викиданных имеют возможность добавлять отдельные фрагменты информации, выполнять редактирование через различные интерфейсы и работать с такими платформами, как Википедия, но при этом они в полной мере несут ответственность за поддержание схемы графа знаний[13] в рабочем состоянии. Однако эту работу должна выполнять команда обученных специалистов в соответствии с чётко продуманными методами. Эти действия осуществляются с помощью инструментов, которые составляют техническую основу системы.

Особым инструментом как в Викиданных, так и в Википедии являются *боты*[14]. Это части программного обеспечения, которые автоматически могут выполнять различные действия на платформе с большой скоростью (более тысячи правок в минуту). Их основная задача — редактирование существующих данных, добавление и импорт новых даных из других ресурсов. Боты — один из ключевых технических компонентов Викиданных.

В статье «Сетевая структура научных революций»[15] на примере Википедии рассматривается процесс формирования знаний в виде постоянно растущих сетей из статей и связывающих их гиперссылок. Эта концепция реализуется за счёт заполнения пробелов в знаниях. Цель этой работы сформулирована в одном предложении: «Авторы проверяют теории научного прогресса на растущих концептуальных сетях и раскрывают управляемые данными условия, лежащие в основе прорывов»[16]. В процессе исследования научных революций было проведено ранжирование всех статей Википедии в виде сети по определённым критериям. Каждый узел сети соответствует определённой статье, имя узла — это заголовок статьи, год рождения узла — это первый год, указанный во введении или в разделе истории как год, когда концепция была задумана. Затем на основе текущего состояния сетей были определены некоторые закономерности в эволюции этих структур на протяжении времени и периоды, когда сеть наиболее быстро менялась. Полученные результаты показали, что человеческие знания растут и, как следствие, происходит постепенное изменение сетевой структуры (заполняются некоторые пробелы в знаниях). Авторы исследования считают, что знания, обнаруженные при заполнении пробелов, будут иметь важное значения для научных инноваций. Это исследование связано с качеством Викиданных, потому что информация для пополнения Викиданных чаще всего берётся из Википедии. Если будут заполнены пробелы в Википедии, то новые данные обязательно будут добавлены в Викиданные и база знаний станет более полной и подробной.

[12] *Piscopo A.* Structuring the world's knowledge: Socio-technical processes and data quality in Wikidata : PhD thesis / Piscopo Alessandro. 2019. URL: https://figshare.com/articles/thesis/Structuring_the_world_s_knowledge_Socio-technical_processes_and_data_quality_in_Wikidata/10998791/2.

[13] Knowledge Graphs / A. Hogan [et al.]. Springer Cham, 2022. (Synthesis Lectures on Data, Semantics, and Knowledge). URL: https://kgbook.org/.

[14] Более подробно о ботах см. раздел «Боты в Викиданных», с. 149.

[15] The network structure of scientific revolutions / H. Ju [et al.] // Computing Research Repository in arXiv (CoRR). 2020. URL: https://arxiv.org/abs/2010.08381.

[16] Оригинальный текст (англ.): The authors test theories of scientific progress on growing concept networks and reveal data-driven conditions underlying breakthroughs.

3. Корзины и мячи

★ Глава основана на статье — *Velitchkov I.* Buckets and Balls // strategic structures. 07/2020. URL: https://www.strategicstructures.com/?p=1889 — с разрешения автора.

Понятие «связанные данные» (Linked Data) по-прежнему недопонято и недооценено. Возможно, эти данные кажутся слишком сложными. Попробуем разобраться. Начнём с переменных. Отметим, что Викиданные являются примером связанных данных (рис. 3.1).

Что такое переменная в SPARQL? Пусть это будет то, что нужно чем-то заполнить. Но как себе представить это «то»? Что-то абстрактное легче представить, если связать с чем-то физическим и конкретным. Трудно представить себе время, но как только мы представим себе часы со стрелками, становится легче. Мы не можем представить себе мебель вообще, но представить стул может каждый.

Другая трудность заключается в том, чтобы сформулировать запрос на языке SPARQL. Хотя работа со SPARQL помогает понять, как работает граф знаний[1], запрос SPARQL на него не похож. Это как с символами в математике: «5 не похоже на пять, в то время как ||||| равно пяти».

Итак, как решить проблему с определением переменной и формулировкой SPARQL-запроса?

Представим себе каждый SPARQL-запрос в виде графа связанных между собой корзин и мячей.

Пусть переменные являются чем-то, что нужно заполнить, но сейчас переменная — это абстрактное понятие. Нам нужен физический контейнер, чтобы наполнить его вещами. Нам нужны корзины и вещи, похожие на мячи. Давайте представим выполнение запроса как заполнение корзин мячами. Тогда процесс заполнения графа мячами будет выглядеть так, как представлено на рис. 3.2. Корзина **?A** должна быть заполнена теми мячами, которые имеют отношение **R** к мячу **B**.

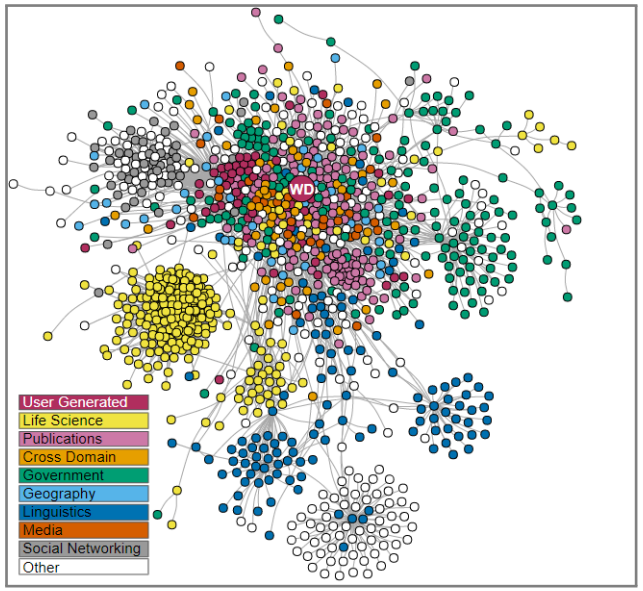


Рис. 3.1. Викиданные в связанном облаке открытых данных.
Базы данных обозначены кружками (Викиданные обозначены как *WD*) с серыми линиями, связывающими базы данных в сети, если их данные выровнены. См. статью в Английской Википедии: Linked data. Wikimedia Commons / Thomas Shafee

[1] Граф знаний — это база знаний, которая использует графовую структурированную модель для интеграции данных. Графы знаний часто используются для хранения взаимосвязанных описаний сущностей — объектов, событий, ситуаций или абстрактных понятий.

Рисунок будет понятнее, если мы его упростим и нарисуем прямую линию (рис. 3.3). Это шаблон графа в нотации «Корзины и мячи». Направление отношения R не показано, но оно всегда слева направо. Тогда процесс написания и выполнения SPARQL-запроса будет состоять из следующих шагов:

1. Выберите свои корзины (в них вы будете собирать нужные вам мячи).
2. Составьте свои условия в виде графа корзин и мячей.
3. Запустите свой запрос, чтобы наполнить корзины мячами.

Теперь напишем запрос, чтобы, например, получить список всех руководителей областей России, выполняя такие шаги:

1. Возьмём две корзины: одна для регионов, другая для руководителей.
2. Корзину для регионов привяжем к мячу «область России» отношением «экземпляр» (instance of). Тогда из множества мячей — объектов Викиданных — в эту корзину попадут только те мячи, которые являются областью России. Корзина для регионов связана с корзиной «руководитель» (head) отношением «имеет руководителя», в нашем случае это губернатор или глава области.
3. Сделаем запрос более интересным и добавим ещё одну корзину для фотографий губернаторов.

Теперь запрос в нотации «Корзины и мячи» будет выглядеть как на рис. 3.4.

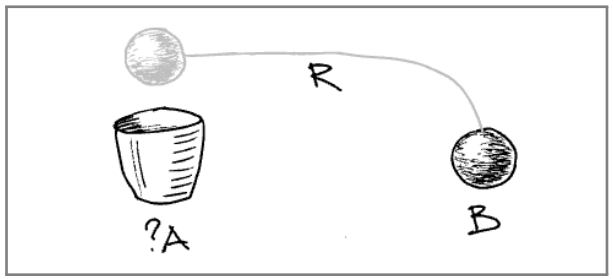


Рис. 3.2. Образец графа заполнения корзин мячами

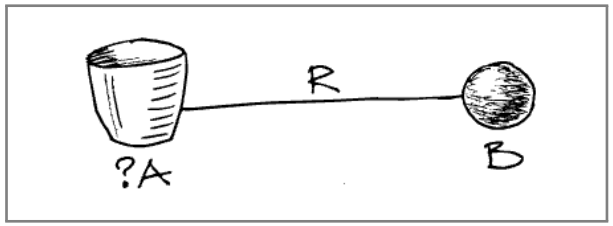


Рис. 3.3. Шаблон графа в нотации «Корзины и мячи»

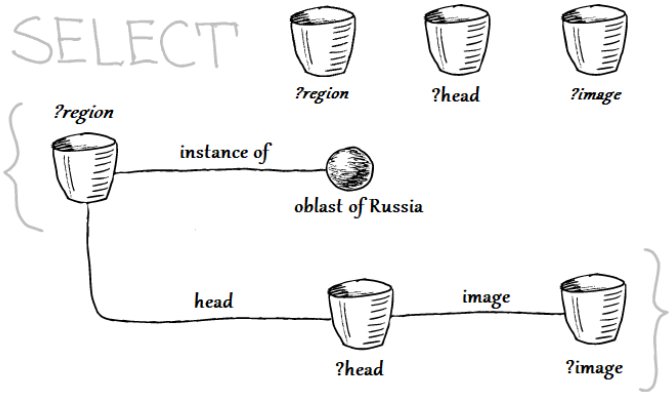


Рис. 3.4. Запрос в нотации «Корзины и мячи» для заполнения корзин «регион» мячами «область России», корзины «руководитель» — губернаторами или главами области, корзины «изображение» — их фотографиями

После выполнения запроса (рис. 3.4) наши корзины будут выглядеть как на рис. 3.5.

Теперь перейдём в сервис Wikidata Query Service (далее — WDQS[2]) и напишем этот запрос на языке SPARQL. Сначала выберем и назовём корзины для заполнения так:

```
SELECT DISTINCT ?region ?head ?image
```

Далее напишем условия соответствия мячей корзинам. Все условия в SPARQL должны заключаться в фигурные скобки (см. рис. 3.4).

Когда нам необходимо определённое отношение или мяч, нам нужно использовать их идентификаторы. Викиданные облегчают поиск идентификатора и подсказывают его, когда вы выбираете отношение (свойство) или мяч (объект) по его имени (метке). Чтобы указать отношения в графе знаний Викиданных, мы используем префикс *wdt:*, а для объектов (наших мячей) — префикс *wd:*.

Следуя нашей нотации (рис. 3.4), напишем условия для заполнения первой корзины *?region*, затем напишем первое отношение. Поскольку общей частью идентификаторов прямых отношений является wdt, мы пишем *wdt:*, а затем в сервисе WDQS нажимаем Ctrl + пробел, чтобы запустить службу подсказок или автозаполнения Викиданных (рис. 3.6).

После этого мы воспроизведём модель из рис. 3.4 в реальном запросе SPARQL.

Обычно при написании SPARQL-запроса его представляют в виде таблицы с тремя столбцами: субъект, предикат, объект; или на языке Викиданных — объект, свойство, значение.

Возможно, начинающему программисту осваивать язык SPARQL будет проще, если первые запросы будут напоминать граф. Попробуем взять граф на рис. 3.4 и написать запрос 3.1 на языке SPARQL максимально похожим образом:

Запрос 3.1. Список руководителей областей России с фотографиями[3]

```
SELECT DISTINCT ?region ?head ?image
{
    ?region wdt:P31 wd:Q835714; # oblast of Russia
            wdt:P6  ?head. # heads of government
    ?head  wdt:P18 ?image. # images of heads of government
}
```

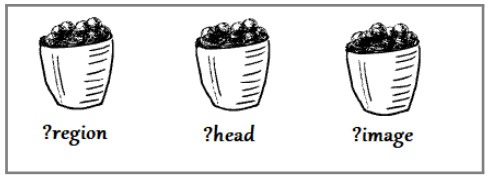


Рис. 3.5. Заполненные корзины после выполнении запроса: *?region* — области России, *?head* — руководители, *?image* — фотографии руководства

[2] См. о службе WDQS на с. 16.

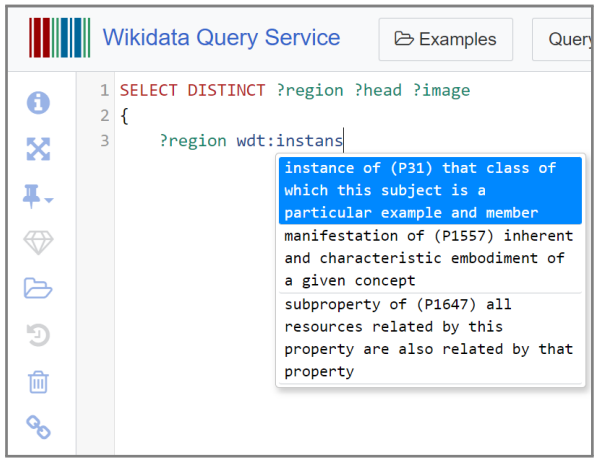


Рис. 3.6. Контекстное меню автозаполнения свойства Викиданых в сервисе WDQS, открываемое с помощью команды Ctrl + пробел

[3] Получено 44 ссылки на области России и их руководителей. Ссылка на SPARQL-запрос: https://w.wiki/4NVc.

Теперь посмотрите, как этот запрос 3.1 выглядит в сервисе WDQS, и запустите его. Затем нажмите на значок глаза слева и выберите *image grid* (рис. 3.7), чтобы просмотреть результаты в виде сетки изображений.

В этой сетке результатов запроса под каждой фотографией мы видим только идентификаторы людей и областей. И если щёлкнуть по гиперссылке идентификатора, то мы получим много информации об этом объекте. Но нагляднее было бы в этой сетке, кроме идентификаторов, указывать имена людей и названия областей. Это похоже на наклеивание этикеток на наши корзины (рис. 3.8).

Метка (*Label*) — это то, что есть у каждого мяча, то есть объекта Викиданных. Метка — это имя, позволяющее различать объекты между собой. В Викиданных есть сервис, упрощающий вывод меток, для этого достаточно добавить в запросе в конец имени переменной слово *Label*. Чтобы активировать этот сервис, на новой строке внутри фигурных скобок наберите Ctrl + пробел; когда вы начнёте писать слово *Label*, тогда будет добавлена строка с этим сервисом[4]. По умолчанию вы получаете подсказку с языком интерфейса и английским языком в качестве альтернативы, если метка недоступна на языке выбранного интерфейса Викиданных.

Викиданные полны таких замечательных сервисов, и для окончательного запроса мы воспользуемся ещё одним. Чтобы получить результат сразу в виде набора фотографий глав областей, без дополнительного нажатия на значок глаза, поместите где-нибудь в своём запросе следующую конструкцию для сервиса WDQS:

```
#defaultView:ImageGrid
```

На самом деле не всё нужно писать вручную. Когда вы начнёте набирать текст, сервис автозаполнения предложит варианты.

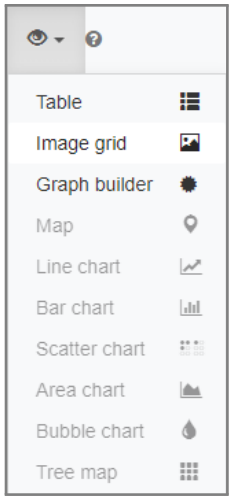


Рис. 3.7. Выбор отображения результатов запроса в виде *image grid* (сетки изображений)

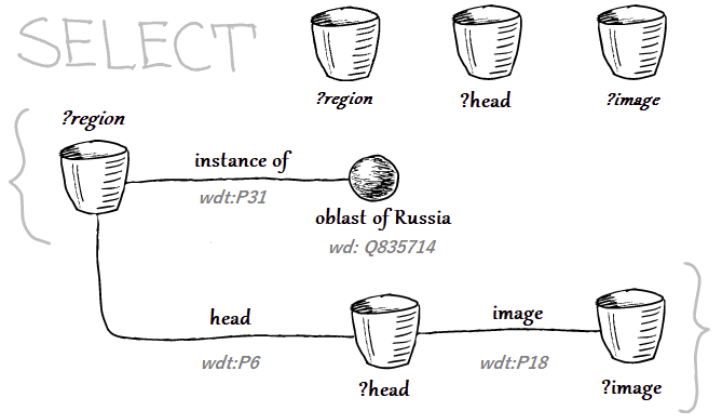


Рис. 3.8. Запрос в нотации «Корзины и мячи» с номерами свойств и объектов Викиданных

[4] Пример вызова этого сервиса представлен в запросе 3.2, в строке 8. В этой строке языком для меток указан русский язык (код ru).

Итоговый запрос 3.2 позволил нам получить требуемую сетку фотографий с подписью в виде названия файла иллюстрации, имени руководителя и названия области (рис. 3.9).

Запрос 3.2. Список руководителей областей России с фотографиями[5]

```
# List of regions of the Russia and images of heads of government
#defaultView:ImageGrid
SELECT DISTINCT ?region ?regionLabel ?head ?headLabel ?image
{
  ?region wdt:P31 wd:Q835714; # ?region is Oblast of Russia
          wdt:P6  ?head.      # has head of government
  ?head  wdt:P18 ?image.      # head has image
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
```

Такое представление о запросах SPARQL, как о связанных корзинах и мячах, может быть полезным, по крайней мере, в начале освоения Викиданных. И, конечно, у каждой метафоры есть свои ограничения. Например, нельзя поместить один и тот же мяч в две разные настоящие корзины, но в эти виртуальные — можно. «Корзины и мячи» могут быть полезны, чтобы вскарабкаться на высоту абстракции Викиданных.

[5] Получено: 44 области России и список их руководителей. Ссылка на SPARQL-запрос: https://w.wiki/4ENR.

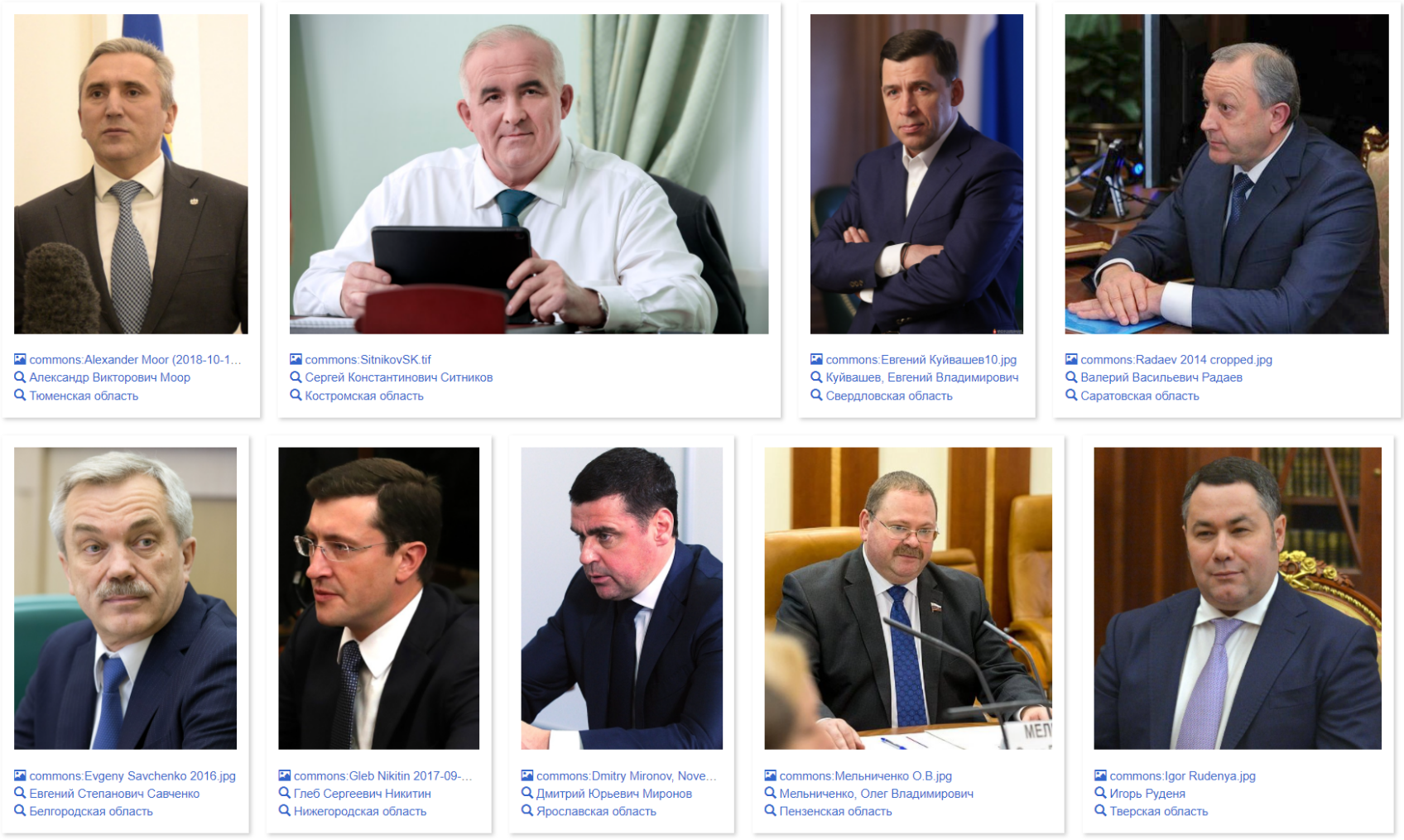


Рис. 3.9. Сетка фотографий руководителей областей с названиями файлов фотографий, с именами руководителей и названиями областей

Часть II

Исследуем объекты Викиданных

4. Воздушные суда и их производители

Эта глава посвящена исследованию различных свойств воздушных судов на основе базы Викиданные. В ходе исследования с помощью SPARQL-запросов, вычисляемых на объектах типа «Воздушные суда», получен список воздушных судов и их производителей, а также число выпущенных самолётов для разных моделей. Для этого числа самолётов по моделям проверено выполнение закона Парето. Также получена диаграмма, отражающая соотношение общего количества производителей воздушных судов по странам. В заключение получена оценка полноты данных, представленных в Википедии и Викиданных. Согласно ей, в Викиданных представлено всего 595 записей о производителях воздушных судов из 1700 на 2020 год. Если считать, что ежегодно будет появляться фиксированное количество новых авиапроизводителей и количество ежегодно заносимых записей в Викиданные останется неизменным, то можно предположить, что примерно через 75 лет (то есть в 2095 году) Викиданные будут содержать записи обо всех авиапроизводителях.

Список самолётов

Воздушное судно — летательный аппарат, поддерживаемый в атмосфере за счёт взаимодействия с воздухом, отличного от взаимодействия с воздухом, отражённым от поверхности земли или воды. К воздушным судам относятся следующие виды летательной техники: автожир, аэростат, вертолёт, винтокрыл, дирижабль, махолёт, планёр и самолёт. К воздушным судам не относятся космические корабли, ракеты, экранопланы (но не экранолёты) и суда на воздушной подушке.

Построим список воздушных судов с помощью запроса 4.1. Здесь мы работаем с объектом «Воздушные суда» Q11436.

Запрос 4.1. Список воздушных судов[1]

```
#List of 'instances of' "aircraft"
SELECT ?plane ?planeLabel
WHERE
{
    ?plane wdt:P31 wd:Q11436. # instances of aircraft
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

По данным сервиса ProWD, наиболее полными по числу свойств на 2017 год были объекты следующих воздушных судов: МиГ-3, Як-36, Mitsubishi A5M. На 2020 год — Sopwith Triplane (18 свойств), Ил-103 (14 свойств) и Martin 2-0-2 (14 свойств). На 2020 год малоинформативными воздушными судами оказались: Бе-1 (3 свойства), Литуаника (4 свойства) и Ла-168 (3 свойства).

[1] Получено: 153 воздушных судна в 2017 году, 299 воздушных судов в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/t3j.

Производители воздушных судов

Построим список производителей воздушных судов, выполнив запрос 4.2.

Запрос 4.2. Производители воздушных судов[2]

```
# Count aircraft having property manufacture, group by manufacture
SELECT ?manufacture ?manufactureLabel (COUNT(?plane) AS ?count)
WHERE {
  ?plane wdt:P31 wd:Q11436.     # instance of aircraft
  ?plane wdt:P176 ?manufacture. # manufacture
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
GROUP BY ?manufacture ?manufactureLabel
```

Результатом запроса 4.2 является список всех производителей воздушных судов с указанием количества различных моделей, производимых данным заводом.

У каких из представленных ниже российских производителей самолётов есть веб-сайты?

- «МиГ»
- Саратовский авиационный завод
- «Туполев»
- «Сухой»

См. ответ на с. 162.

[2] Получено: 300 производителей воздушных судов в 2017 году, 590 производителей воздушных судов в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/t3n.

Количество выпущенных воздушных судов

Авиационная промышленность — это одна из самых крупных отраслей машиностроения в мире. В её задачи входит как разработка, так и производство различной воздушной техники. Для того чтобы оценить, какие модели воздушных судов являются самыми массовыми, мы построим диаграмму произведённых судов различных моделей с помощью запроса 4.3.

Запрос 4.3. Список моделей, упорядоченный по количеству выпущенных самолётов[3]

```
# List of aircraft models, sorted by number of aircraft built
SELECT ?plane ?planeLabel ?planes_produced WHERE {
  ?plane wdt:P31 wd:Q11436. # instance of aircraft
  ?plane wdt:P1092 ?planes_produced.  # total aircraft manufactured
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
ORDER BY DESC(?planes_produced)
```

Результаты запроса 4.4 были загружены в программу Microsoft Excel, где и была построена диаграмма количества построенных самолётов для каждой из моделей (рис. 4.1). На этой диаграмме видно, что на 2020 год больше всего было выпущено воздушных судов следующих моделей: Piper PA-32 (7842 штуки), Piper PA-24 Comanche (4857), Junkers W 34 (3000), Piper J-4 (1251).

Запрос 4.4. Список, включающий только те самолёты, которые были выпущены в количестве более 10 штук[4]

```
1 # List of aircraft models, sorted by number of aircraft built
2 #defaultView:BarChart
3 SELECT ?plane ?planeLabel ?planes_produced WHERE {
4   ?plane wdt:P31 wd:Q11436. # instance of aircraft
5   ?plane wdt:P1092 ?planes_produced.  # total aircraft manufactured
6   FILTER (?planes_produced > 10)
7   SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
8 }
9 ORDER BY (?planes_produced)
```

Найдите соответствие между датой основания и компанией в следующей таблице:

Компания	Дата основания
«МиГ»	1939
«Вымпел»	18 ноября 1949
«Туполев»	8 декабря 1939
«Сухой»	22 октября 1922

См. ответ на с. 162.

[3] Получено: 288 моделей, для которых известно число выпущенных самолётов, 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/v4J.

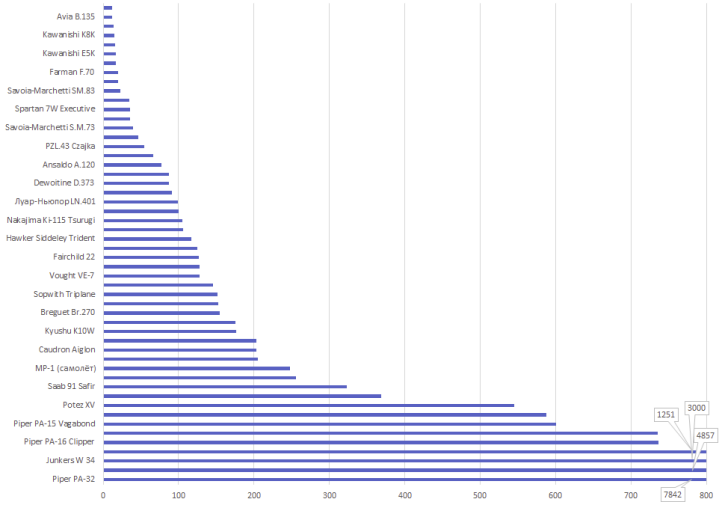


Рис. 4.1. Количество выпущенных воздушных судов с разбивкой по моделям, 2020 год

[4] Получен отфильтрованный по количеству выпущенных самолётов список, состоящий из 124 моделей. Ссылка на SPARQL-запрос: https://w.wiki/v4N.

Некоторые модели воздушных судов были выпущены в единичных экземплярах, поэтому для повышения читабельности диаграммы (рис. 4.1) их можно исключить. Для получения нового списка добавим в строке 6 запроса 4.4 фильтр по количеству выпущенных самолётов.

Теперь попытаемся ответить на вопрос: выполняется ли закон Парето относительно числа моделей самолётов? Чтобы построить график процентного соотношения количества выпущенных моделей самолётов к общему числу произведённых самолётов, необходимо выполнить следующие шаги:

1. Подсчитать общее число самолётов по всем моделям с помощью запроса 4.5.

Запрос 4.5. Общее число произведённых самолётов[5]

```
SELECT (SUM(?count) as ?sum) WHERE {
  SELECT ?count WHERE {
    ?plane wdt:P31 wd:Q11436; # instance of aircraft
       wdt:P1092 ?count. # total aircraft production
  }
}
```

2. По оси X отложить число рассматриваемых моделей самолётов (то есть при $x = 1$ мы рассматриваем число выпущенных самолётов первой модели, при $x = 2$ — число выпущенных самолётов первой и второй модели и так далее). По оси Y взять значение $F(n) = \sum_{i=1}^{n} f(i)$, где $f(i)$ — число выпущенных самолётов модели i. При этом выполняется условие $f(i) > f(j)$, при $i < j$, где i, j — номер модели самолёта (то есть модели самолётов заранее упорядочены так, что первых моделей произведено больше, чем последующих, это хорошо видно на рис. 4.2). Также по оси X следует отложить вторую шкалу от 0 до 1, чтобы легче было определить параметры для проверки выполнения закона Парето.

По графику, представленному на рис. 4.2, видно, что 80 % всех выпущенных самолётов приходится на 16 различных моделей самолётов, что составляет 9,2 % от общего числа моделей. Закон Парето утверждает, что «20 % усилий дают 80 % результата, а остальные 80 % усилий — лишь 20 % результата». Можно сделать вывод, что выполняется более сильный закон, чем принцип Парето, относительно числа моделей самолётов.

Найдите соответствие между расположением штаб-квартиры компании и названием компании производителя самолётов в следующей таблице:

Компания	Штаб-квартира
«Камов»	Улан-Удэ
«Авиадвигатель»	Пермь
Улан-Удэнский завод	Москва
«Сухой»	Люберцы

См. ответ на с. 162.

[5] Общее число выпущенных самолётов на 2020 год составляет 44 151. Ссылка на SPARQL-запрос: https://w.wiki/vE9.

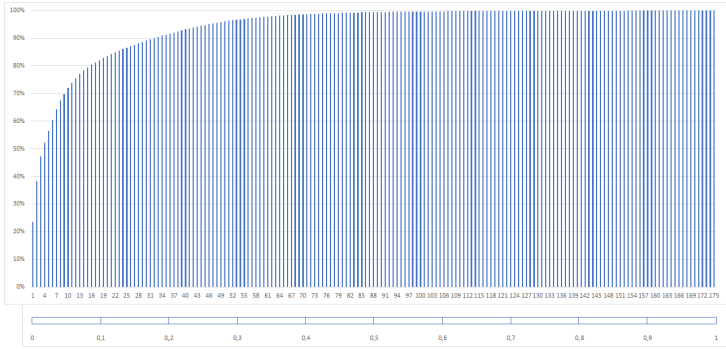


Рис. 4.2. Процентное соотношение количества выпущенных моделей самолётов по n моделям к общему числу выпущенных самолётов за всё время, 2020 год

В каких странах производят самолёты

Построим список количества производителей воздушных судов по странам. Для выполнения запроса 4.6 используем группировку по странам (`GROUP BY`) и при помощи функции `Count()` для каждой страны подсчитаем общее количество авиастроительных заводов.

Получив список стран по количеству заводов-производителей авиационной техники, построим пузырьковую диаграмму соотношения количества производителей воздушных судов по странам с помощью запроса 4.6.

Запрос 4.6. Список стран с указанием количества производителей воздушных судов[6]

```
#defaultView:BubbleChart
SELECT ?country ?countryLabel (count(?manufacturer) as ?count) WHERE
{
    ?manufacturer wdt:P31 wd:Q936518. # instance of aerospace manufacture
    ?manufacturer wdt:P17 ?country. # belong to country
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
GROUP BY ?country ?countryLabel
```

В результате выполнения запроса 4.6 будут построены пузырьковые диаграммы (рис. 4.3 и 4.4), в которых круги означают страны, а их размеры соответствуют количеству авиапроизводителей в указанной стране. Такие диаграммы помогают более наглядно увидеть разницу в количестве авиационных заводов в разных странах.

Сравнивая две пузырьковые диаграммы за 2017 (рис. 4.3) и 2020 (рис. 4.4) годы, можно сделать вывод, что основными производителями воздушных судов в мире в 2017 и 2020 годах были: США (115 заводов в 2017-м и 135 заводов в 2020 году), Великобритания (30 и 43 завода), Германия (17 и 26 заводов) и Россия (17 и 21 завод). Лидером по-прежнему являются США, а вот Франция за три года сумела опередить Германию, увеличив количество производств до 29 (Германия — 26), тем самым заняв третье место. В целом соотношение по странам остаётся прежнем.

Ответ на запрос 4.2 показывает, что Викиданные содержат неполный список производителей воздушных судов по сравнению с данными сайта aviationfanatic.com. Исследуем вопрос полноты Викиданных ниже.

Как называется воздушное судно, удерживаемое в воздухе огромным баллоном с горючим смертельно опасным газом, расположенным прямо над головами пассажиров?
См. ответ на с. 163.

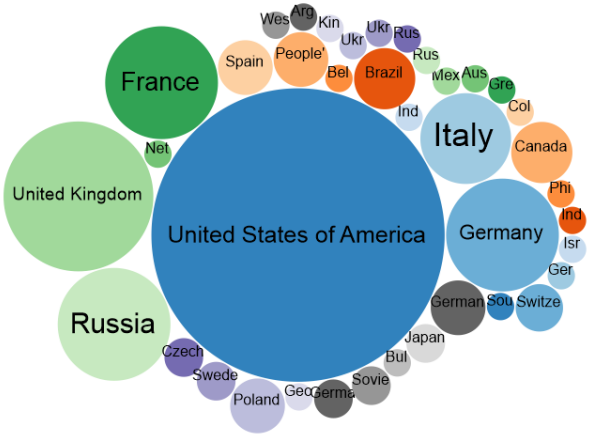


Рис. 4.3. Соотношение количества производителей воздушных судов по странам, 2017 год

[6] Получено: 39 стран, выпускающих самолёты, по данным на 2017 год, 46 стран — на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/t3v.

Рис. 4.4. Соотношение количества производителей воздушных судов по странам, 2020 год

Полнота Викиданных по числу производителей воздушных судов

Согласно сайту aviationfanatic.com существовало около 1700 производителей воздушных судов на 2017 год и 1939 судов на 2020 год[7], но SPARQL-запрос 4.2 вернул всего 300 авиазаводов в Викиданных в 2017 году и 595 авиазаводов в 2020 году. Из этого можно сделать вывод о неполноте Викиданных.

Попробуем спрогнозировать, когда Викиданные будут описывать не меньше самолётов, чем сайт aviationfanatic.com. За три года количество производителей воздушных судов увеличилось на 239, что составляет ежегодный прирост примерно на 80 авиапроизводителей. Также за это время в Викиданные была занесена информация о 295 авиапроизводителях, то есть ежегодно добавляется около сотни авиазаводов. На 2020 год в Викиданных не было информации о 1344 авиапроизводителях, представленных на сайте aviationfanatic.com. Если считать, что ежегодно будет добавляться фиксированное количество новых авиапроизводителей и количество ежегодно заносимых записей в Викиданные останется неизменным, то можно предположить, что примерно через 75 лет (то есть в 2095 году) Викиданные будут содержать записи обо всех авиапроизводителях, представленных на сайте aviationfanatic.com.

В категории Википедии «Авиастроительные компании России»[8] указано наличие в России 58 авиастроительных компаний в 2017 году и 62 заводов, институтов и корпораций, связанных с самолётостроением, в 2020 году, но в то же время на сайте aviationfanatic.com указано наличие 61 завода в 2017 году и 71 предприятия в 2020 году[9]. Среди авиастроительных компаний России представлены такие компании, как Иркут, МиГ, Туполев.

Упражнения

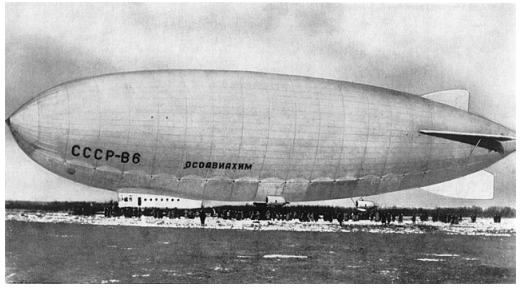
1. Найти самолет с максимальным радиусом полета.
2. Отметить на политической карте мира местоположение главных офисов компаний авиапроизводителей.
3. Найти производителя с максимальным числом изготовленных самолетов, используя свойство *manufacturer (P176)* у воздушных судов.
4. Когда был построен первый самолёт?
5. Какие фирмы первыми выпустили 10, 100 и 1000 самолётов?
6. Нарисуйте диаграмму количества выпускаемых самолётов в мире и в России по годам.

[7] Aviation Fanatic — Free online aviation encyclopedia and pilot logbook. List of all Manufacturers. URL: https://www.aviationfanatic.com/ent_list.php?ent=3.

[8] См.: https://w.wiki/vF4.

[9] Aviation Fanatic — Free online aviation encyclopedia and pilot logbook. List of all Manufacturers. URL: https://www.aviationfanatic.com/ent_list.php?ent=3&MAN_-Country=RU.

Какое воздушное судно здесь изображено?



См. ответ на с. 163.

5. Аниме: загадочный и поразительный мир японской анимации

В рамках этой главы исследуется объект Викиданных «аниме» (Q1107). С помощью SPARQL-запросов, вычисляемых на объектах типа «аниме» в Викиданных, получен список сэйю[1], упорядоченный по числу озвученных ими аниме, построен график числа аниме, озвученных одним сэйю, построен граф, связывающий сэйю и озвученные ими аниме, получена оценка трудоспособного возраста сэйю.

Экземпляры объекта «Аниме»

Аниме — это японская мультипликация. Она стоит особняком и выделяется своим визуальным стилем, однако есть и менее очевидные отличия: так, по сравнению с американской и японской анимацией у аниме значительно шире список жанров — от детских и семейных комедий до драматических историй, которые на Западе обычно представляются только в фильмах с живыми актёрами[2].

У каждого аниме есть свои актёры озвучивания. В дальнейшем под «сэйю» будем понимать японского актёра озвучивания. В японской анимации слова «актёры озвучивания» и «сэйю» являются синонимами[3]. Под словом «тайтл» (от англ. *title*, *название*) обычно понимают конкретное аниме[4]. В общем же смысле слово «тайтл» означает понятие, объединяющее различные виды продукции (от кинофильма до романа), созданные на основе конкретного произведения, за которым закреплено строго определённое название[5].

Чтобы работать со списком аниме в Викиданных, нам понадобятся объект «аниме» (Q1107) и свойство «экземпляр» (P31). Получим список всех аниме без учёта подклассов (запрос 5.1).

[1] Сэйю — это японские актёры озвучивания. Сэйю обычно озвучивают роли персонажей в аниме, видеоиграх, фильмах, а также на радио и телевидении или выступают в роли рассказчика в радиопостановках. Кроме того, голоса сэйю используются в рекламе, голосовых объявлениях, аудиозаписях книг и учебных материалов, а также для переозвучивания. Сэйю могут быть как мужчины, так и женщины, как взрослые, так и дети.

Сэйю Кэндзи Акабанэ озвучивал роль персонажа Sasuke Sarutobi в видеоигре Ikemen Sengoku, 2018 год. Wikimedia Commons / numan (CC BY-SA)

[2] *Slowpoke T.* Аниме vs мультипликация. 2020. URL: https://cadelta.ru/anime/id6119.

[3] Шикимори — энциклопедия аниме и манги. URL: https://shikimori.one/.

[4] *Кузьмина Д. П.* Аниме: от социализации к межкультурной коммуникации // Традиции и инновации в пространстве современной культуры. Липецк, 2021. С. 235—238. URL: https://www.elibrary.ru/item.asp?id=46107074.

[5] *Магера Ю. А.* История появления первых японских комиксов на русском языке // Японские исследования. 2018. № 4. С. 6—23. ISSN 2500-2872. URL: https://publications.hse.ru/pubs/share/direct/229921306.pdf.

Запрос 5.1. Список аниме без учёта подклассов[6]

```
# List of instances of anime
SELECT ?anime ?animeLabel
WHERE
{
    ?anime wdt:P31 wd:Q1107. # instance of anime
    SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en,ja"}
}
```

В действительности в Викиданных объектов аниме гораздо больше, но они являются экземплярами не объекта «аниме», а его подклассов, таких как, например, «аниме-сериал» (Q63952888). Чтобы получить список жанров аниме и количество аниме, относящихся к этим жанрам, выполним запрос 5.2.

Запрос 5.2. Список жанров аниме и количество аниме, относящихся к этим жанрам[7]

```
# Select anime and its subclasses with number of titles
# corresponding to these subclasses
SELECT ?subAnime ?subAnimeLabel (COUNT(?subAnimeInst) AS ?count)
WHERE {
  ?subAnime wdt:P279* wd:Q1107.   # select anime subclass list
  ?subAnimeInst wdt:P31 ?subAnime # link titles and subclasses
    SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en,ja"}
}
GROUP BY ?subAnime ?subAnimeLabel
ORDER BY DESC(?count)
```

Визуализировать распределение аниме по жанрам можно в виде круговой диаграммы (рис. 5.1), построенной с помощью сервиса Rawgraphs (https://app.rawgraphs.io). По-английски такой вид диаграм, имеющих радиально расходящиеся лучи, называется *sunburst diagram*. Полученная классификация аниме по жанрам не идеальна, так как есть большое смещение в сторону аниме-сериалов: из 4875 тайтлов 2984 отнесены к жанру «аниме-сериал» (62.7 %); вероятно, классификация жанров аниме в Викиданных требует дальнейшего уточнения. Также в подклассы попали понятия, относящиеся не к общей классификации, а к отдельным аниме, например «Евангелион».

[6] Получено: 683 результата в 2017 году и 216 результатов в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4ABw.

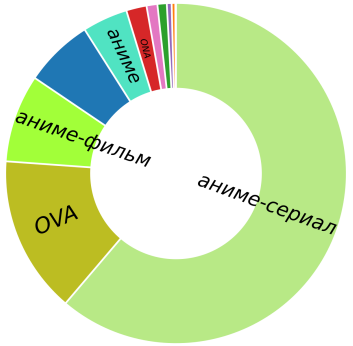


Рис. 5.1. Жанры аниме на круговой диаграмме, 2021 год

[7] Получено: 11 результатов в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4ABj.

Получим список всех аниме, включая тайтлы, относящиеся к жанрам аниме (запрос 5.3).

Запрос 5.3. Список всех аниме в Викиданных[8]

```
# List of instances of anime and subclasses of anime
SELECT ?anime ?animeLabel
WHERE
{
    ?anime wdt:P31/wdt:P279* wd:Q1107. # instance of anime/subclass
    SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en,ja"}
}
```

Аниме, о которых есть наиболее полная информация на Викиданных, — это Гуррен-Лаганн (Q4277), Space Battleship Yamato (Q4292), Project A-ko (Q4316). Аниме с малоинформативными записями на Викиданных оказались Doraemon (Q711311), The Animal Conference on the Environment (Q97195557), Assassins Pride (Q96737300).

Среди всех аниме-тайтлов в Викиданных больше всего свойств, по данным сервиса ProWD, у фильма Fullmetal Alchemist: The Sacred Star of Milos (Q1004318)[9] (24 свойства).

Список сэйю, упорядоченный по числу озвученных ими аниме

Разумеется, в большинстве аниме участвует не один, а множество персонажей. Соответственно, разных персонажей озвучивают разные сэйю. Большинство сэйю озвучили за свою карьеру несколько тайтлов, а многие — десятки тайтлов. Талантливых сэйю приглашают озвучивать сразу нескольких персонажей в одном аниме. Одним из самых популярных сэйю является Хироси Камия, имеющий множество наград и озвучивший более 180 аниме. Самым известным аниме с его участием является «Атака титанов», где он озвучил одного из главных персонажей — капитана Леви.

Построим упорядоченный список сэйю по числу озвученных ими аниме (запрос 5.4).

[8] Получено: 683 результата в 2017 году и 4875 результатов в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/49zY.

[9] «Стальной алхимик: Священная звезда Милоса» — полнометражный аниме-фильм, являющийся продолжением аниме-сериала «Стальной алхимик». Его главные герои — братья-алхимики, использующие свои магические способности для борьбы с силами зла и противостояния преступникам.

Запрос 5.4. Упорядоченный список сэйю по числу озвученных ими аниме[10]

```
# Ordered list of actors-seiyu according to the number of anime where they took part in
SELECT ?seiyu ?seiyuLabel (COUNT(?anime) AS ?count) WHERE
{
  ?anime wdt:P31/wdt:P279* wd:Q1107;  # instance of anime/subclass
         wdt:P725 ?seiyu.  # instance of seiyu (voice actor)
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en,ja" }
}
GROUP BY ?seiyu ?seiyuLabel  # group by seiyu
ORDER BY DESC(?count)  # order by count of voiced anime
```

[10] Получено: 148 результатов в 2017 году и 2910 результатов в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4Xph.

График по числу сэйю, озвучивших одно и более аниме

Построим линейную диаграмму сэйю, озвучивших аниме, с помощью запроса 5.5. При этом чем больше аниме озвучил сэйю, тем правее на диаграмме он будет находиться (рис. 5.2).

Запрос 5.5. Построение графика по числу сэйю, озвучивших одно и более аниме[11]

```
1  # Graph of the number of voice actings of different seiyu
2  #defaultView:LineChart
3  SELECT ?seiyuRoles (COUNT(?seiyuRoles) AS ?quantity) WHERE {
4    FILTER(?seiyuRoles < 71) # limit a numbef of seiyu in graph
5    {       # count quantity of voice acting by one seiyu
6      SELECT (COUNT(?seiyu) AS ?seiyuRoles) WHERE {
7        ?anime wdt:P31/wdt:P279* wd:Q1107; # instance of anime and its subclasses
8                  wdt:P725 ?seiyu.        # instance of seiyu
9        SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en,ja"}
10     }
11     GROUP BY ?anime            # group list by number of voiced anime
12     ORDER BY DESC(?seiyuRoles) # order by voice acting quantity (descending)
13   }
14 }
15 GROUP BY ?seiyuRoles       # grouping and
16 ORDER BY DESC(?seiyuRoles) # sorting seiyu by number of voice actings
```

[11] Получено: 13 результатов в 2017 году и 58 результатов в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4JvT.

На рис. 5.2 хорошо видно, что чем выше планка количества озвученных аниме, тем меньше сэйю достигают этой планки. В строке 4 запроса 5.5 установлено ограничение в 71 аниме, поскольку сэйю, которые отметились в большем количестве аниме, — единицы и расширение графика вправо было бы не слишком информативным.

Многие сэйю, как показано на рис. 5.2, озвучили только одно аниме — на графике их оказалось 254. Однако сэйю — это профессия, которой люди зачастую посвящают всю свою жизнь. То, что, согласно Викиданным, человек за многие годы принял участие в озвучке только одного аниме, может быть связано с отсутствием информации о других его ролях в Викиданных.

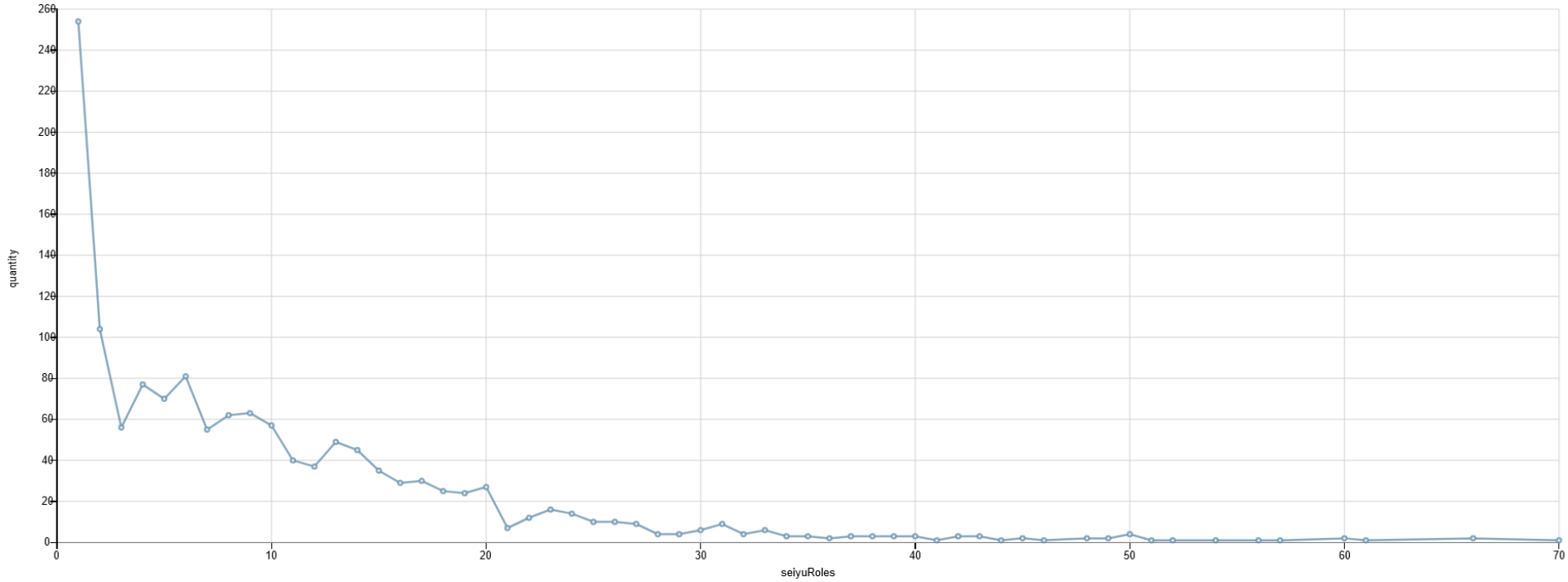


Рис. 5.2. График числа ролей, озвученных различными сэйю, 2021 год.
График построен на основе данных, полученных с помощью запроса 5.5

Граф, связывающий сэйю и озвученные ими аниме

Итак, многие сэйю за свою карьеру успевают озвучить несколько аниме. Чтобы нагляднее показать эту взаимосвязь, построим граф, связывающий сэйю и озвученные ими аниме с помощью запроса 5.6. Фрагмент итогового графа представлен на рис. 5.3.

Запрос 5.6. Построение графа, связывающего сэйю и озвученные ими аниме[12]

```
1  # Graph of seiyu with more than one anime
2  #defaultView:Graph
3  SELECT DISTINCT ?item ?itemLabel ?rgb ?link
4  WHERE
5  { # voice actors (seiyu) with more than one anime
6    VALUES ?toggle { true false }
7    VALUES ?seiyu { wd:Q6381410 wd:Q1347031 wd:Q1207010
8                    wd:Q233902  wd:Q1323728 wd:Q2440809
9                    wd:Q355173  wd:Q957795  wd:Q50033}
10   ?anime  wdt:P31/wdt:P279* wd:Q1107; # instance of anime/subclass
11           wdt:P725 ?seiyu;           # seiyu who voiced this anime
12   SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en,ja"}
13   BIND(IF(?toggle,?anime,?seiyu) AS ?item).
14   BIND(IF(?toggle,?animeLabel,?seiyuLabel) AS ?itemLabel).
15   BIND(IF(?toggle,"FFFFFF","7FFF00") AS ?rgb).
16   BIND(IF(?toggle,"",?anime) AS ?link).
17 }
```

В переменную `?seiyu` (строки 7–9 запроса 5.6) с помощью оператора `VALUES` записан массив объектов Викиданных, соответствующих некоторым известным сэйю — Кадзуэ Комия (Q6381410) и другим. Мы выбрали только девять сэйю в иллюстративных целях, поскольку для большего числа сэйю граф стал бы неудобным для восприятия.

Конструкция `BIND(IF(?toggle, ?anime, ?seiyu)...)` в строке 13 позволяет определить тип вершины графа: если `?toggle` принимает значение `true`, то вершина графа соответствует аниме, иначе — сэйю. В строках 14 и 15 определяются тип подписи для вершины и цвет вершины. Строка 16 позволяет отобразить связи между сэйю и аниме.

[12] Получено: 826 результатов в 2017 году и 494 результата в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4HFt.

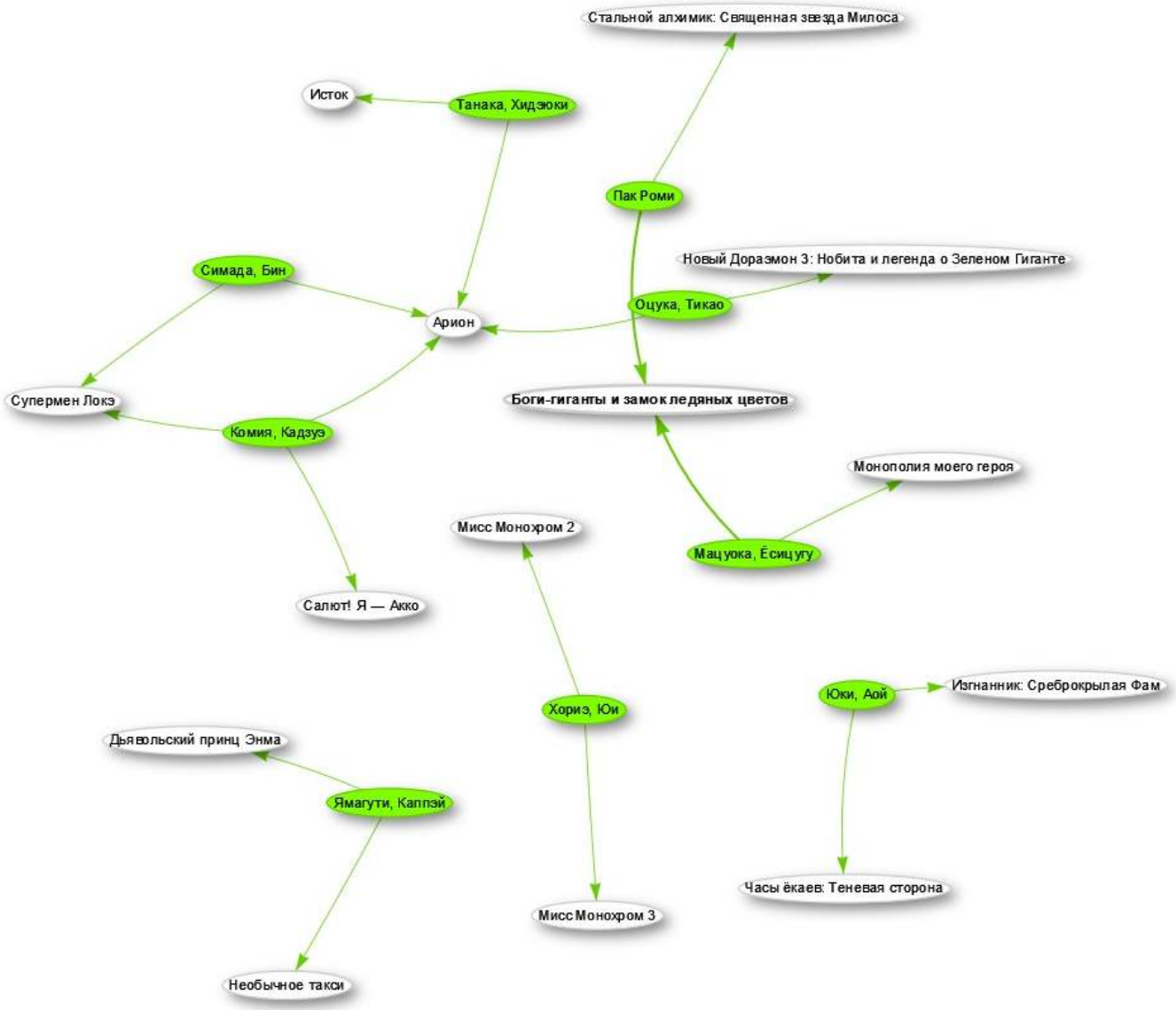


Рис. 5.3. Фрагмент графа, связывающего сэйю и озвученные ими аниме, 2021. Граф построен на основе данных, полученных с помощью запроса 5.6

Полнота Викиданных по числу аниме и актёров

Список тайтлов в Русской Википедии[13] содержит 1638 аниме. Также можно посмотреть телевизионные показы аниме в России по годам[14]. На сайте любительской энциклопедии аниме Shikimori[15] список аниме включает 801 страницу по 20 наименований. Нетрудно подсчитать, что на сайте есть информация о 16 020 тайтлах, в то время как в Викиданных объектов, описывающих аниме, всего 4875 (см. запрос 5.3). К тому же стоит учитывать, что скорость выхода новых аниме довольно велика. Из этого можно сделать вывод, что Викиданные крайне неполно отражают данные (есть информация только о 29.6 % аниме).

Возможно, приведённые ниже статьи и сайты не будут являться АИ[16], но с их помощью можно проанализировать информацию об имеющихся аниме и сделать дополнительные выводы о неполноте Викиданных.

- На сайте любительской озвучки AniDub[17] приведён список из 5756 аниме.
- На сайте онлайн-кинотеатра AnimeSpirit[18] приведён список из 1968 аниме.
- На новостном форуме по тематике аниме AnimeLand[19] приведён список из 4795 аниме.
- На сайте онлайн-кинотеатра Anivost[20] приведён список из 420 аниме.

Какие-то сайты появились позже, какие-то раньше, поэтому количество аниме на них может разниться, причём довольно серьёзно. Если упорядочить все приведённые сайты, данные Русской Википедии, Английской Википедии и Викиданные по количеству аниме, то Викиданные окажутся не на последнем месте, но, например, вышеупомянутой энциклопедии Shikimori они уступают почти в 4 раза.

[13] Проект:Аниме и манга/Списки/Список аниме. URL: https://w.wiki/4JVE.

[14] Проект:Аниме и манга/Телевизионные показы аниме в России по годам. URL: https://w.wiki/4JVH.

[15] Шикимори — энциклопедия аниме и манги. URL: https://shikimori.one/.

С помощью SPARQL подсчитайте, сколько аниме вышло за минувший год.

[16] Авторитетный источник (АИ) — это ресурс, который владеет информацией, в компетентности и актуальности которой не может быть никаких сомнений. См. https://w.wiki/3u9v.

[17] АниДаб. URL: http://online.anidub.best/.

[18] AnimeSpirit. URL: http://animespirit.tv/.

[19] AnimeLand. URL: http://animeland.su/.

[20] Anivost. URL: https://amedia.cc/.

Вспомним запрос 5.4, в котором говорилось о 2910 сэйю на Викиданных. Дело в том, что поиск производился только по актёрам озвучивания, связанным с аниме, поэтому результат оказался таким скромным. Если запросить информацию о всех актёрах озвучивания (то есть убрать ограничение на категорию аниме), то количество результатов может увеличиться в 5 раз (запрос 5.7). Значительный прирост числа результатов относительно запроса 5.4 напоминает нам о том, что в индустрии озвучивания гораздо больше направлений, чем только аниме, например озвучивание фильмов и видеоигр. Сэйю могут участвовать в работе и над такими проектами, что нужно учитывать при формировании запросов.

Запрос 5.7. Получение списка актёров озвучки и числа озвученных ими проектов[21]

```
# Ordered list of actors according to the quantity of projects
# voiced by them
SELECT ?actor ?actorLabel (COUNT(?project) AS ?count)
WHERE
{
  ?project wdt:P725 ?actor. # instance of voice actor
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en,ja"}
}
GROUP BY ?actor ?actorLabel
ORDER BY DESC(?count)       # order by number of voiced projects
```

Круговая диаграмма на рис. 5.4 — один из вариантов визуализации данных, полученных с помощью скрипта 5.7 и затем обработанных сервисом Rawgraphs. С помощью подобных диаграмм можно, например, оценить, какой актёр внёс наибольший вклад в развитие индустрии озвучивания.

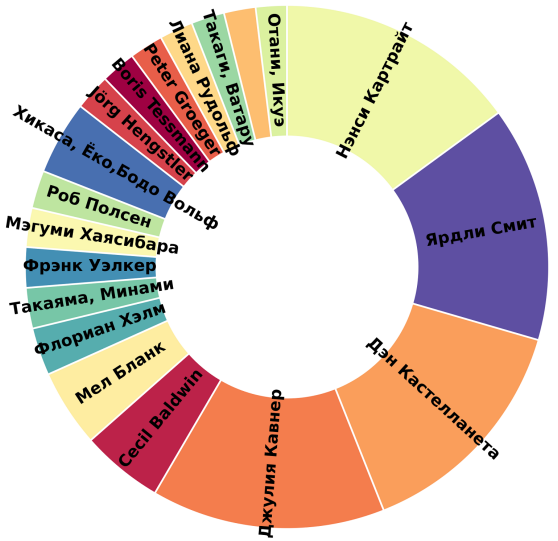


Рис. 5.4. Круговая диаграмма числа ролей, озвученных различными актёрами, по данным на 2021 год

[21] Получено: 3965 результатов в 2017 году и 14 744 результата в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4aQt.

Указана ли дата публикации у аниме?

Каждый ценитель японской анимации желает знать, в каком году вышло его любимое аниме. Викиданные располагают этой информацией не в полной мере. В следующем запросе 5.8 подсчитывается количество аниме с незаполненным полем publication date (дата публикации). То, что поле должно быть пустым, указано в строке 6 запроса 5.8 с помощью пустых квадратных скобок.

Запрос 5.8. Получение списка аниме, у которых на Викиданных не указана дата выхода[22]

```
# List of anime the release date of which is empty
SELECT ?anime ?animeLabel
WHERE
{
    ?anime wdt:P31/wdt:P279* wd:Q1107;  # instance of anime
    FILTER NOT EXISTS { ?anime wdt:P577 [] }
    SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en,ja"}
}
```

На 2021 год из 4875 аниме на Викиданных (см. запрос 5.3) у 2940, а это 62 %, не указана дата выхода. В 2017 году из 683 аниме на Викиданных только 237 (то есть 35 %) не имели указанной даты выхода. Похоже, что, к сожалению, увеличение количества информации не всегда сопровождается сохранением её качества.

Напишите скрипт для вычисления доли аниме, у которых не указана дата публикации, относительно всех аниме на Викиданных. Сравните эту долю с долей за 2021 год (62 %) и сделайте вывод об изменении качества Викиданных.

[22] Получено: 237 результатов в 2017 году и 2940 результатов в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4Hcz.

Анализ возраста, в котором сэйю озвучивают аниме

Как и в любой другой профессии, у актёра озвучки есть возраст, когда он находится в «расцвете сил» и может озвучить множество аниме. Использование SPARQL и внешних инструментов для анализа данных, подобных языку программирования Python[23], может позволить оценить такой активный возраст на основе информации из Викиданных.

Чтобы получить исходные данные для исследования, необходимо выполнить три SPARQL-скрипта и экспортировать результаты их выполнения в формате .csv[24]. CSV-файлы затем используются в скрипте на языке Python, который генерирует выходной график. Запускать программы на Python можно, например, на платформе Google Colaboratory[25].

[23] Язык Python — интерпретируемый язык программирования, благодаря своей гибкости используемый для решения разнообразных задач. Его можно применять в том числе и для работы с Викиданными: например, в разделе 16 (с. 149) описан процесс создания программных ботов для Викиданных.

[24] CSV (comma-separated values) — формат представления табличных данных, в котором таблица хранится в виде последовательности строк текста. Эти строки содержат значения полей таблицы, разделённые запятыми.

[25] Google Colaboratory (Colab) — облачная среда разработки от компании Google, в которой можно создавать и запускать скрипты на языке программирования Python, а также делиться результатами своей работы с другими людьми. Также этот сервис удобен тем, что предоставляет вычислительные мощности — как обычные процессоры, так и видеокарты для, например, работы с нейронными сетями. Сервис доступен по ссылке: https://colab.research.google.com.

Получить список всех зарегистрированных в Викиданных сэйю и их дат рождения можно двумя способами (запросы 5.9 и 5.10): с помощью команды `SERVICE` и с помощью конструкции `rdfs:label`. Различия между запросами 5.9 и 5.10 заключаются в том, что:

- метка (имя) сэйю в первом случае получается с помощью переменной `?seiyuLabel` (в таком случае нужно указать команду `SERVICE` для установки языков, на котором будут возвращены имена), а во втором — с помощью конструкции `rdfs:label`;
- в первом варианте скрипта необходимо указывать `?seiyuLabel` как параметр `GROUP BY`, чтобы связать объекты сэйю и их метки.

Запрос 5.9. Получение списка дат рождения сэйю с помощью `SERVICE`[26]

```
SELECT ?seiyu ?seiyuLabel ?bDate
WHERE {
  ?anime (wdt:P31/(wdt:P279*)) wd:Q1107;
    wdt:P725 ?seiyu.
  ?seiyu wdt:P569 ?bDate.
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en,ja"}
}
GROUP BY ?seiyu ?seiyuLabel ?bDate
```

[30] Получено: 2515 результатов в 2021 году. SPARQL-запрос: https://w.wiki/4FPq.

Запрос 5.10. Получение списка дат рождения сэйю с помощью `rdfs:label`[27]

```
SELECT ?seiyu (SAMPLE(?seiyu) AS ?seiyuLabel) ?bDate
WHERE {
  ?anime (wdt:P31/(wdt:P279*)) wd:Q1107;
    wdt:P725 ?seiyu.      # seiyu is anime voice actor
  ?seiyu wdt:P569 ?bDate. # has a birthday
  ?seiyu rdfs:label ?label.
}
GROUP BY ?seiyu ?bDate
```

[31] Получено: 2515 результатов в 2021 году. SPARQL-запрос: https://w.wiki/4FPn.

Получим список всех зарегистрированных в Викиданных аниме и дат их выхода (запрос 5.11).

Запрос 5.11. Получение дат выхода аниме[28]

```
# Get all anime objects, their names and release dates
SELECT ?anime ?animeLabel ?animePubDate ?animeSeriesStartDate
WHERE {
  ?anime (wdt:P31/(wdt:P279*)) wd:Q1107.          # object of anime/subclass
  OPTIONAL { ?anime wdt:P577 ?animePubDate. }       # release date of a movie
  OPTIONAL { ?anime wdt:P580 ?animeSeriesStartDate. } # start date of a series
  SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en,ja"}
}
```

Визуализируйте результаты работы скрипта 5.11.
Усложните задачу — добавьте на график даты окончания показа сериалов.

Обратите внимание, что запрос 5.11 получает не только даты выхода полнометражных аниме (P577), но и даты начала показа сериалов (P580).

Получим ссылки между объектами сэйю и аниме, которые они озвучивали (запрос 5.12).

Запрос 5.12. Получение ссылок между сэйю и аниме[29]

```
# List of links between seiyu and anime where they are involved in
SELECT DISTINCT ?item ?itemLabel ?link ?itemType
WHERE
{
  VALUES ?toggle { true false }
  ?anime  wdt:P31/wdt:P279* wd:Q1107;   # instance of anime/subclass
          wdt:P725 ?seiyu.              # list seiyu who acted in this anime

  BIND(IF(?toggle,?anime,?seiyu) AS ?item).                # anime/seiyu object
  BIND(IF(?toggle,?animeLabel,?seiyuLabel) AS ?itemLabel). # anime/seiyu labels link
  BIND(IF(?toggle,?seiyu,?anime) AS ?link).                # seiyu/anime link
  BIND(IF(?toggle,?seiyu,"seiyu") AS ?itemType).
  SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en,ja"}
}
```

[28] Получено: 5264 результата в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4ENc.

[29] Получено: 27 092 результата в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4ELh.

Результат анализа удобно представить в виде гистограммы. Для её построения воспользуемся средствами таких библиотек для Python, как pandas[30] и Matplotlib[31]. Код скрипта, создающего гистограмму, опубликован на сервисе GitHub[32].

В результате получим гистограмму, по оси абсцисс которой отложен возраст в годах, а по оси ординат — суммарное количество ролей, озвученных всеми сэйю такого возраста. Получившаяся гистограмма представлена на рис. 5.5.

[30] Библиотека pandas предоставляет различные функции для обработки табличных данных для тех, кто пишет программы на языке Python.

[31] Библиотека Matplotlib, также написанная на языка Python, позволяет программистам рисовать графики и диаграммы.

[32] Ссылка на скрипт в проекте wd_book на GitHub: https://git.io/J1UGA.

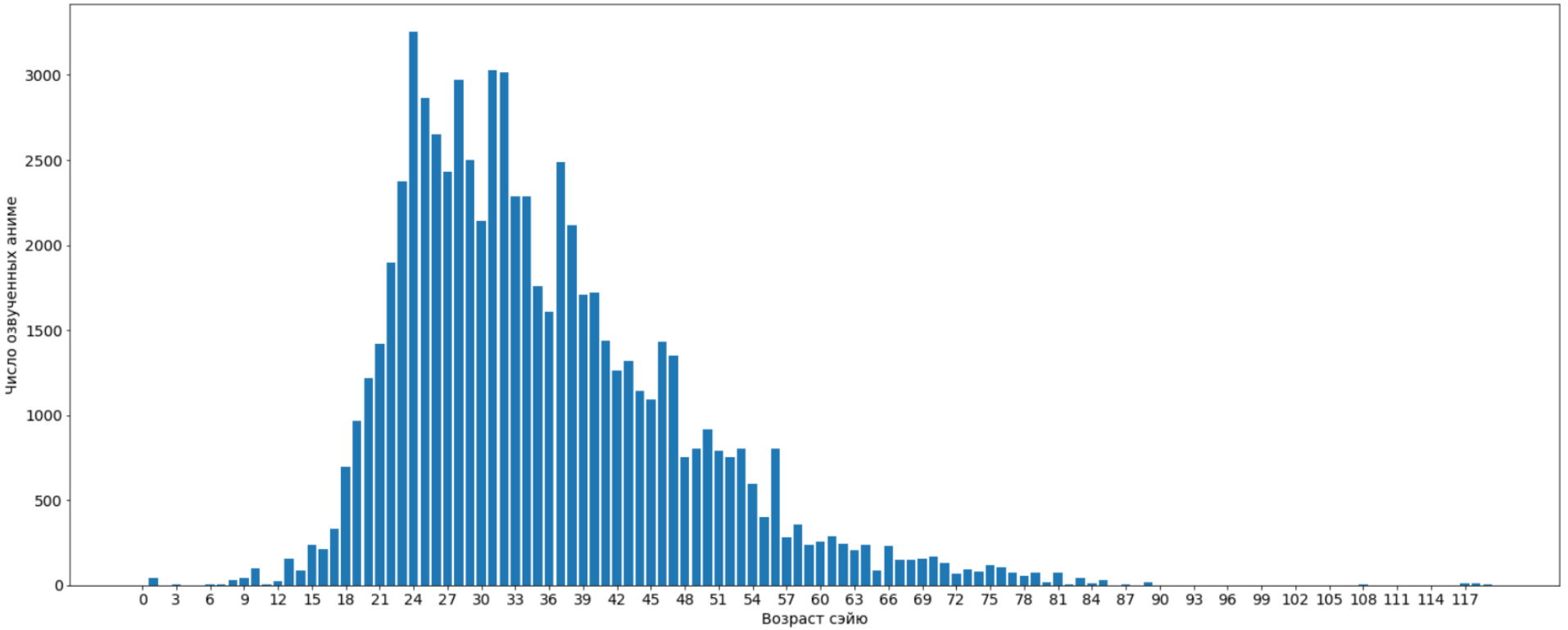


Рис. 5.5. Гистограмма с числом аниме, озвученных сэйю разных возрастов, 2021 год. Гистограмма построена на основе данных, полученных с помощью запросов 5.9 (или 5.10), 5.11 и 5.12.

Отметим следующий забавный факт: в Викиданных нашлись случаи, когда сэйю родился позже, чем вышло аниме с его участием. Вероятно, это связано с отсутствием информации в Викиданных о втором сезоне/перезапуске аниме. Например, в 2021 году такая ситуация была с аниме Sazae-san (Q11304591) и сэйю Нобунага Симадзаки (Q5968283): сэйю родился в 1988 году, а дата выхода аниме с его участием — 1969 год.

Упражнения

1. Вывести 10 самых популярных аниме, вышедших на экраны в текущем году. Популярность оценить по числу статей в разных языковых разделах. Для подсчёта числа статей об объекте Викиданных используйте SPARQL-конструкцию `wikibase:sitelinks`. Например, если статья про аниме есть в трёх Википедиях на русском, английском и испанском языках, то его популярность равна трём.
2. Вывести пять аниме, в которых задействовано самое большое число сэйю-женщин.
3. Построить пузырьковую диаграмму (BubbleChart) распределения аниме по жанрам (сколько аниме в каждом жанре), воспользовавшись свойством «подкласс» (P279).
4. Отметить на карте места рождения сэйю.
5. Построить гистограмму или пузырьковую диаграмму национальностей сэйю.
6. Построить гистограмму количества вышедших аниме по годам или количества сэйю по годам рождения.
7. Построить гистограммы, аналогичные рис. 5.5, но с учётом пола сэйю (одну для мужчин, другую для женщин).

6. В каких населённых пунктах России больше рождаётся учёных, в сельских или городских?

В главе исследуется объект Викиданных «населённый пункт (Q486972)» и его свойства. В каждом из разделов представлены задачи, решённые с помощью SPARQL-запросов.

Был получен список населённых пунктов, построены пузырьковые диаграммы с количеством населения в «населённых пунктах» по странам. Построена диаграмма, показывающая долю населения, проживающего в населённых пунктах относительно всего населения страны. Диаграмма показала, что высокий процент населения, проживающего в населённых пунктах, приходится на сельскохозяйственные страны, в то время как в более индустриальных странах меньшая доля населения проживает в населённых пунктах.

На 2017 год Википедия описывала примерно половину населённых пунктов (75 тыс.), Викиданные содержали менее 3 % таких поселений (4 тыс.) относительно данных переписи за 2010 год (155,5 тыс.). На 2021 год Викиданные содержат менее 12 % таких поселений (17 тыс.) относительно данных той же переписи за 2010 год.

Для сравнения сельских и городских поселений построены диаграммы количества учёных, сгруппированных по родам деятельности и разделённых по месту рождения — сельское или городское.

Для поиска более полных ответов на поставленные выше задачи были найдены более общие классы для объекта «населённый пункт (Q486972)» с помощью свойства «частный случай понятия (P31)». Трудность исследования вызвана отсутствием чёткой типологии населённых пунктов (например, от численности населения) в законодательстве России и в Викиданных.

Подсчитайте, сколько человек на 1 км^2 живёт в Барабинске и в Алейске? В каком из этих *населённых пунктов* плотность населения выше?

См. ответ на с. 164.

Герб какого отечественного или зарубежного населённого пункта изображён на рисунке?

См. ответ на с. 164.

Список «населённых пунктов»

Построим список всех населённых пунктов с помощью запроса 6.1.

Запрос 6.1. Список всех населённых пунктов[1]

```
# List of all human settlements
SELECT ?hum ?humLabel WHERE{
  ?hum wdt:P31 wd:Q486972. # instance of human settlement
  SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en"}
}
```

В 2021 году оказалось невозможным получить список населённых пунктов из-за большого числа объектов и поэтому слишком долгой работы запроса 6.1. Для подсчёта числа всех населённых пунктов обратимся к функции `COUNT()` в запросе 6.2.

Запрос 6.2. Количество всех населённых пунктов[2]

```
# Number of human settlements
SELECT (COUNT(?hum) AS ?count) WHERE {
  ?hum wdt:P31 wd:Q486972. # instance of human settlement
}
```

Среди отечественных населённых пунктов на Викиданных, которым соответствуют статьи Русской Википедии, почти пустыми являются, например, бывшая деревня Борисово (Q4093951) (3 свойства) и Бригадирское лесничество (Q21668554) (4 свойства).

По данным сервиса ProWD, среди отечественных населённых пунктов больше всего свойств (36) у Ялты (Q128499). Лидером по всему миру является Токио (Q1490) (73 свойства).

Это герб населённого пункта России или другой страны?

См. ответ на с. 164.

[1] Получено: 411 393 пункта в 2017 году. SPARQL-запрос: https://w.wiki/4d7x.

[2] Получено: 563 126 населённых пунктов в 2021 году. SPARQL-запрос: https://w.wiki/4d7s.

Список стран по суммарному количеству населения

С помощью запроса 6.3 построим упорядоченный список стран по суммарному количеству населения, проживающего в «населённых пунктах».

Запрос 6.3. Список стран по суммарному количеству населения, проживающего в «населённых пунктах»[3]

```
1  # List of countries by population in settlements
2  SELECT ?country ?countryLabel (SUM(?population) as ?sumPopulation)
3  WHERE {
4    ?hum wdt:P31 wd:Q486972;       # instance of human settlement
5         wdt:P17 ?country;         # in the ?country
6         wdt:P1082 ?population.    # has ?population
7    SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en"}
8  }
9  GROUP BY ?country ?countryLabel
10 ORDER BY DESC (?sumPopulation)
```

Для подсчёта количества населения по странам используем команду `SUM()` в строке 2 запроса 6.3. Для группировки населённых пунктов по странам используем команду `GROUP BY` в строке 9 того же запроса.

Пузырьковая диаграмма на рис. 6.1 и 6.2 показывает соотношение стран по количеству населения в «населённых пунктах» в 2017 и 2021 годах соответственно. Размер пузырька соответствует количеству населения, проживающего в «населённых пунктах» одной страны. SPARQL-запрос для построения такой диаграммы доступен по ссылке: https://w.wiki/4dAv.

В 2017 году больше всего населения проживало в «населённых пунктах» Бразилии (Q155) (12 млн), Пакистана (Q843) (10 млн), Мексики (Q96) (8 млн), Йемена (Q805) (8 млн), Индии (Q668) и Бангладеш (Q902) (по 7 млн).

На рис. 6.2 можно увидеть список стран на 2021 год: Индия (Q668) (30 млн), Китай (Q148) (28 млн), Мексика (Q96) (17 млн), Индонезия (Q252) (13 млн), Канада (Q16) (9 млн) и Саудовская Аравия (Q851) (9 млн).

Итак, результаты запроса 6.3 в 2017 и 2021 годах существенно разнятся. По этим результатам получается, что за четыре года в населённых пунктах Индии стало больше на 23 млн человек.

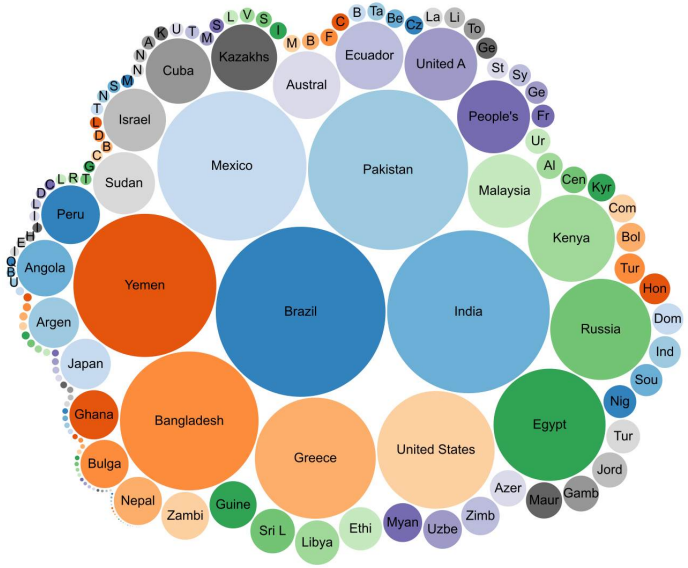


Рис. 6.1. Пузырьковая диаграмма с суммарным количеством населения, проживающего в «населённых пунктах» на 2017 год

[3] Получено: 161 страна в 2017 году и 213 стран в 2021 году. SPARQL-запрос: https://w.wiki/4d9M.

Рис. 6.2. Пузырьковая диаграмма с суммарным количеством населения, проживающего в «населённых пунктах» на 2021 год

Полнота Викиданных по заполненности свойств населённых пунктов

Населённый пункт — это общее название мест с постоянными жителями. По версии редакторов Викиданных, в понятие «насёленный пункт» входят города, сёла, деревни и другие виды поселений[4]. Точной информации о количестве населённых пунктов в мире мы не нашли. Поэтому проверим полноту тех населённых пунктов, которые есть в Викиданных и которые использовались для решения задачи. В задачах выше мы использовали свойства «численность населения (P1082)» и «государство (P17)» (привязка к стране). Исходя из этого, проверку полноты разделим на подзадачи:

1. Проверка заполненности свойства «численность населения».
2. Проверка принадлежности к государству.

Проверка заполненности свойства «численность населения»

Для такой проверки напишем SPARQL-запрос[5], который выведет населённые пункты с незаполненным свойством «численность населения». Произведя расчёты, получили, что только у 9,3 % населённых пунктов мира было указано свойство «численность населения» на 2017 год. В 2021 году получили 11,2 % населённых пунктов мира с заполненным свойство «численность населения». Итак, одновременно с ростом числа населённых пунктов в Викиданных растёт доля пунктов с заполненным свойством «численность населения».

Проверка принадлежности к государству

А теперь посмотрим населённые пункты, у которых не указана принадлежность к какой-либо стране, с помощью SPARQL-запроса[6].

Результаты запроса показывают, что число пунктов без привязки к стране растёт с годами. Поэтому вычисления, связанные с населёнными пунктами и странами, по определению будут неполными даже относительно тех пунктов, которые уже есть в Викиданных.

Доля населения страны, проживающего в «населённых пунктах»

Построим список стран, упорядоченный по доли населения (в процентах), проживающей в населённых пунктах, относительно числа всех жителей страны (листинг 6.4).

[4] Несколько наиболее частотных классов таких поселений можно посмотреть в табл. 6.1 на с. 55.

Герб населённого пункта какой страны изображён?

См. ответ на с. 164.

[5] В 2017 году запрос выдал 372 997 населённых пунктов с незаполненным свойством «численность населения». Тот же запрос в 2021 году выдал 507 078 таких населённых пунктов. Ссылка на SPARQL-запрос: https://w.wiki/4FUz.

[6] В 2017 году нашлось 8427 объектов, у которых не указана принадлежность к какой-либо стране. В 2021 году таких объектов уже больше — 27 824. SPARQL-запрос: https://w.wiki/4FV8.

Запрос 6.4. Соотношение количества людей, проживающих в населённых пунктах, к количеству всех людей в стране[7]

```
# An ordered list of the ratio of the number of people living in
# "human_settlement"to the number of inhabitants in the country.
SELECT ?country ?countryLabel ?proportionPopulation WHERE {
 SELECT ?country ?countryLabel (SUM(?population / ?pop)
       as ?proportionPopulation) WHERE {
  ?hum wdt:P31 wd:Q486972;    # instances of human settlement
       wdt:P17 ?country;          # has ?country
       wdt:P1082 ?population.     # has ?population
  ?country wdt:P1082 ?pop.    # population in the country
  SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en"}
 }
 GROUP BY ?country ?countryLabel
}
ORDER BY ?proportionPopulation
```

Столбчатая диаграмма на рис. 6.3 позволяет увидеть для каждой страны отношение количества людей, проживающих в населённых пунктах, к числу жителей в стране на 2017 год. Наиболее высокий процент приходился на следующие страны: Кирибати (78 %), Ниуэ (70 %), Греция (53 %), Тувалу (48 %), Коморы (43 %), Маврикий (42 %). В 2022 году картина изменилась: Нигерия (93 %), Папуа — Новая Гвинея (71 %), Израиль (50 %), Греция (47 %), Азербайджан (47 %), Казахстан (37 %).

Для построения аналогичной диаграммы за 2022 год (рис. 6.4) мы немного изменили запрос 6.4, добавив ограничение — выводить страны с населением более 5 млн человек[8]. Таким образом, рис. 6.4 построен с помощью запроса https://w.wiki/5Ajn.

На 2017 год в странах большой восьмёрки доля жителей в населённых пунктах составила: Россия (2.98 %), США (1.76 %), Япония (0.80 %), Канада (0.26 %), Франция (0.20 %), Германия (0.24 %), Великобритания (0.18 %), Италия (0.07 %). В 2022 году значения доли населения снизились (рис. 6.4): Россия (0.045 %), США (0.014 %), Япония (0.008 %), Канада (0.23 %), Франция (0.005 %), Германия (0.005 %), Великобритания (0.014 %), Италия (0.0005 %). Отметим, что это страны промышленно развитые.

Отметим некоторую закономерность, которую показывают эти диаграммы, а именно: чем более развита страна, тем меньшая доля людей проживает в объектах типа «населённый пункт (Q486972)». По-видимому, рис. 6.3 и 6.4 подсказывают нам, что в таких странах население в основном проживает в поселениях другого типа. Осталось найти, что это за поселения.

Герб населённого пункта какой страны изображён?

См. ответ на с. 164.

[7] Получено: 158 результатов в 2017 году и 206 результатов в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4dE3.

[8] Строка с ограничением:
`FILTER (?pop > 5000000).`

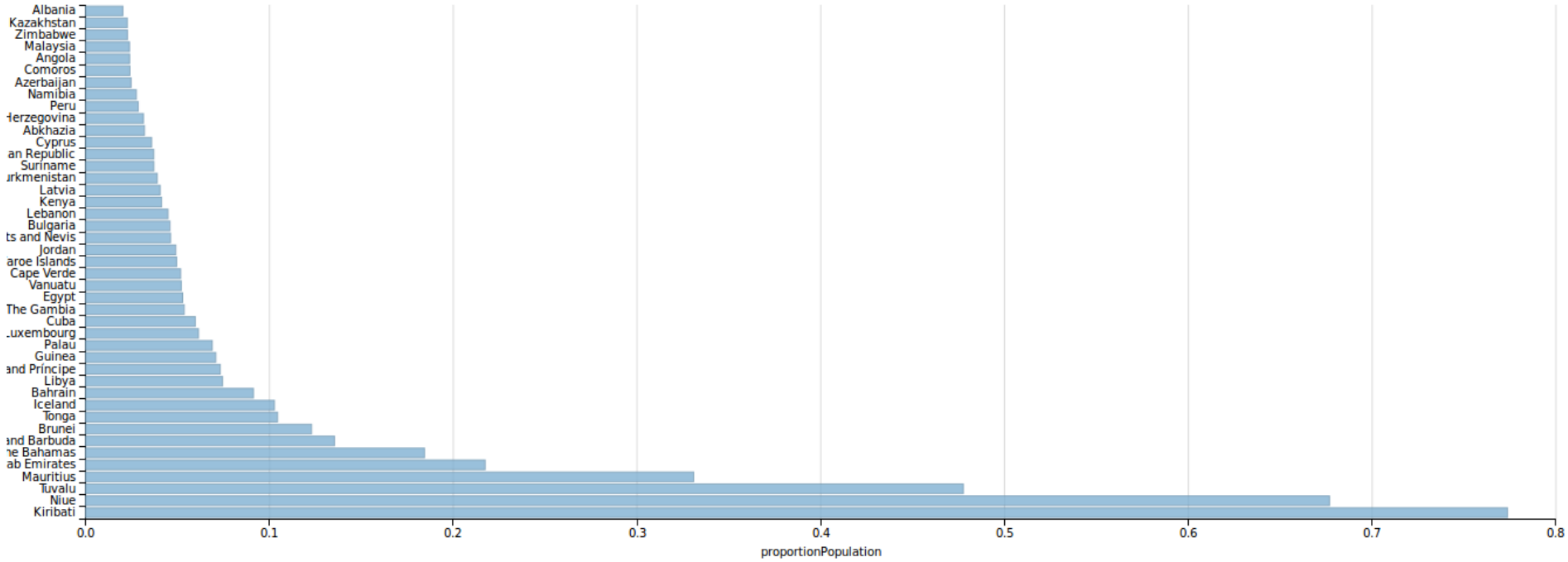


Рис. 6.3. Доля населения страны, проживающего в «населённых пунктах», 2017 год.
Ссылка на SPARQL-запрос: https://w.wiki/4dE3

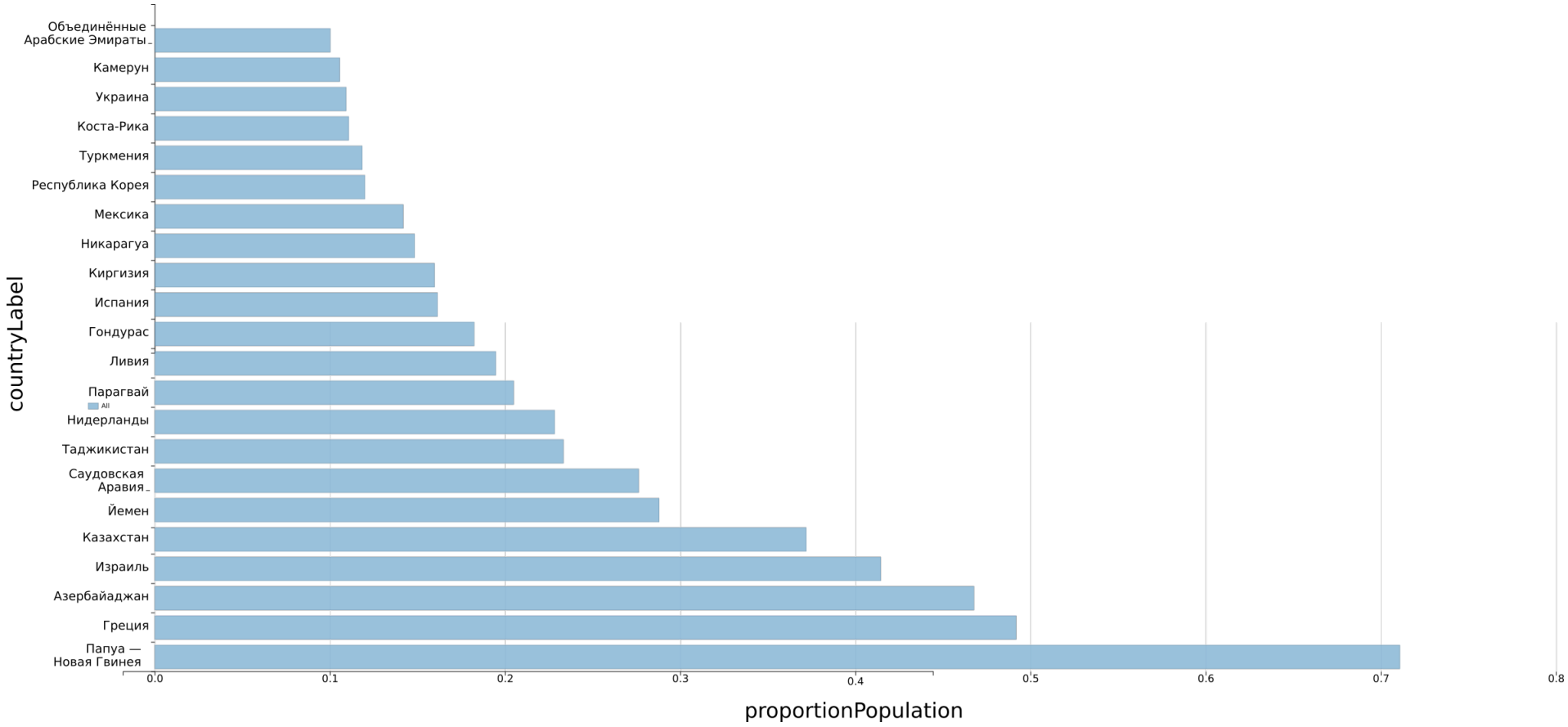


Рис. 6.4. Доля населения страны, проживающего в «населённых пунктах», 2022 год. Выбраны страны с населением более 5 млн чел. SPARQL-запрос: https://w.wiki/92jq

Список классов, сопутствующих «населённому пункту» в свойстве «экземпляр»

Будем называть «классом» такие объекты Викиданных, которые связаны с каким-либо объектом Викиданных посредством свойства экземпляр (P31). Цель этого раздела — найти объекты X, для которых объект «населённый пункт (Q486972)» является *классом*, и получить другие классы объекта X, кроме «населённого пункта». Эти «другие» классы будем называть сопутствующими «населённому пункту». С помощью запроса 6.5 найдём объекты, сопутствующие «населённому пункту».

Запрос 6.5. Список объектов, сопутствующих «населённому пункту»[9]

```
1 # List of classes accompanying the human_settlement (instance of)
2 SELECT ?inst (COUNT(?hum) as ?sumHum)
3 WHERE{
4   ?hum wdt:P31 wd:Q486972; # instance of human settlement
5        wdt:P31 ?inst.      # other objects in instance
6   SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en"}
7 }
8 GROUP BY ?inst
```

Для ускорения выполнения запроса 6.5 выполним два шага.

Во-первых, выключим из рассмотрения поселения, имеющие в списке экземпляров только «населённый пункт», поскольку тогда других сопутствующих классов нет. Для этого добавим в запрос 6.6 строку 9 с фильтром, требующим, чтобы у объекта `?hum`, кроме экземпляра (P31) «населённый пункт (Q486972)», был ещё какой-либо класс.

Во-вторых, в строке 8 запроса 6.6 уберём такие объекты переменной `?inst`, которые имеют свойство «государство (Q17)». Это позволит отсечь сотни типов населённых пунктов, специфичных для отдельных стран, например «административно-территориальная единица России».

Эти преобразования позволили выполнить запрос 6.6 по всем странам мира за 13 мс, что является вполне приемлемым временем.

Герб населённого пункта какой страны изображён?

См. ответ на с. 164.

[9] Получено: 610 результатов в 2017 году и 1245 результатов в 2021 году. SPARQL-запрос: https://w.wiki/4dEW.

Запрос 6.6. Список классов, сопутствующих «населённому пункту» в свойстве «экземпляр» без специализированных классов разных стран[10]

```
1  # List of objects with the class of human settlement, without
2  # country and single human settlement
3  SELECT ?inst ?instLabel (COUNT(?hum) as ?sumHum)
4  WHERE{
5    ?hum wdt:P31 wd:Q486972;  # instance of human settlement
6         wdt:P31 ?inst.       # other objects in instance
7
8    MINUS {?inst wdt:P17 []}. # without country
9    FILTER(?inst != wd:Q486972 ). # without human settlement
10   SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en"}
11 }
12 GROUP BY ?inst ?instLabel
13 ORDER BY DESC (?sumHum)
```

В табл. 6.1 представлены классы, сопутствующие «населённому пункту» в свойстве «экземпляр», и сравнение количества этих классов за 2017 и 2021 годы. Получены данные по всему миру, представлены первые строки ответа на запрос 6.7.

Запрос 6.6 показал, что на 2021 год на первых местах среди сопутствующих «населённому пункту» классов были поселения «латенского периода» (Q106505016), «бронзового века» (Q106491277) и поселения «доисторического времени, где есть письменность» (Q106505070).

Попробуем сформулировать запрос так, чтобы отсечь всё множество доисторических поселений. Что есть общего у этих трёх объектов на Викиданных? Они являются экземплярами объектов, которые, в свою очередь, являются экземплярами объектов «археологической культуры» (Q465299), «исторического периода» (Q11514315), «археологического века» (Q15401699), «всемирной истории» (Q200325) и «геологического периода» (Q392928). Применим такой фильтр в запросе 6.7 в строках 11–12, чтобы отсечь именно эти типы объектов.

В итоге 707 классов запроса 6.6 мы сократили в запросе 6.7 до 89 различных классов, сопутствующих «населённому пункту» в свойстве «экземпляр». В табл. 6.1 были представлены первые 10 наиболее частотных классов из 89.

[10] Получено: 355 записей в 2017 году и 707 записей в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4ePf.

Таблица 6.1. Сопутствующие «населённому пункту» классы, 2017 и 2021 годы

№	Класс населённого пункта	2017	2021	Δ
1	Село (Q532)	2844	4853	+2009
2	Муниципалитеты (Q15284)	1181	3376	+2195
3	Деревни (Q5084)	662	1761	+1099
4	Археологические памятники (Q839954)	425	887	+462
5	Местные поселения (Q3257686)	425	158	-257
6	Разрушенные города (Q14616455)	423	388	-40
7	Города (Q515)	322	545	+223
8	Малые города (Q3957)	277	446	+169
9	Заброшенные деревни (Q350895)	254	474	+220
10	Внутренние районы (Q2983893)	207	503	+296

Запрос 6.7. Список классов, сопутствующих «населённому пункту» в свойстве «экземпляр», без исторических объектов[11]

```
# List of classes accompanying the human_settlement in the property 'instance of'
# without historical objects
SELECT ?inst ?instLabel (COUNT(?hum) as ?sumHum)
WHERE{
  ?hum wdt:P31 wd:Q486972;    # instance of human settlement
       wdt:P31 ?inst.         # other objects in instance of human settlement
  ?inst wdt:P31 [wdt:P31 ?typ]. # instance of instance
  MINUS {?inst wdt:P17 []}.   # without country

  # without human settlement and prehistoric settlements
  FILTER(?inst != wd:Q486972 && ?typ != wd:Q465299 && ?typ != wd:Q11514315
         && ?typ != wd:Q15401699 && ?typ != wd:Q200325 && ?typ != wd:Q392928).

  SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en"}
}
GROUP BY ?inst ?instLabel
ORDER BY DESC (?sumHum)
```

[11] Получено: 89 результатов. Ссылка на SPARQL-запрос: https://w.wiki/5AdW.

Отечественные учёные на селе и в городе

Подсчитаем и сравним число отечественных учёных, родившихся в сельских и городских типах населённых пунктов. Решим эту задачу в пять шагов:

1. Выявим список сельских и список городских типов поселений именно в России.
2. Определим основные научные направления, представленные в Викиданных.
3. Выявим способ определения отечественных учёных.
4. Сделаем такую диаграмму, на которой разным цветом будут указаны разные научные направления (математики, физики, химики и так далее) для учёных, родившихся в *сельских поселениях*.
5. Сделаем аналогичную диаграмму по *городским поселениям* и сравним результаты.

Список сельских и список городских типов поселений в России

С помощью запроса 6.8 выведем список типов поселений и их количество для объектов, имеющих свойство «численность населения» (P1082) и принадлежащих России (Q159).

Запрос 6.8. Список классов поселений и их количество для объектов, имеющих свойство «численность населения» в России[12]

```
1  # List of instances of settlement with population in Russia
2  SELECT ?class ?classLabel (COUNT(?class) AS ?count) WHERE {
3    SELECT ?class ?classLabel WHERE {
4      [] wdt:P31 ?class; # noname instance of class
5           wdt:P17 wd:Q159;  # in Russia
6           wdt:P1082 [].     # has population
7      FILTER(?class != wd:Q486972).# skip settlement itself
8      SERVICE wikibase:label{bd:serviceParam wikibase:language "ru"}
9    }
10 }
11 GROUP BY ?class ?classLabel
12 ORDER BY DESC (?count)
```

Из 229 классов поселений, полученных с помощью запроса 6.8, были выбраны те, которых много, и такие, которые можно уверенно отнести к сельским или городским населённым пунктам. 11 таких первых поселений представлены в табл. 6.2. Число в столбце «№» указывает номер строки с классом поселения в списке из 229 ответов на запрос 6.8, упорядоченных по их количству.

Обратите внимание на безымянные переменные (`[]`) в строках 4 и 6 запроса 6.8. Эта безымянность связана с тем, что важен факт указания численности населения, но не количество людей в них (строка 6), нам не нужны названия сёл и деревень (строка 4).

Наличие строки 6 в запросе 6.8 (`wdt:P1082 []`) существенно уменьшает числа в табл. 6.2 в столбце «Количество». Это связано с тем, что у многих населённых пунктов не заполнено поле «численность населения».

Объект Викиданных «населённый пункт (Q486972)» может быть указан в качестве класса (свойство «экземпляр (P31)») как у сельских, так и у городских поселений. Поэтому он нам не поможет в различении этих типов поселений и мы его пропускаем с помощью строки 7 запроса 6.8.

Используя команду `FILTER NOT EXISTS`, измените запрос 6.8 так, чтобы получить число классов с пустым полем «численность населения».
См. пример использования этой команды в запросе 5.8 на с. 42.

[12] Получено: 229 разных классов поселений в 2022 году. Ссылка на SPARQL-запрос: https://w.wiki/4zPi.

Таблица 6.2. Сельские и городские населённые пункты России, 2022 год

№	Название класса	Количество	Население
1	Сельское поселение в России (Q634099)	18 103	34 млн
2	Деревня (Q5084)	15 517	1.8 млн
3	Село (Q532)	10 204	10.9 млн
4	Посёлок (Q2514025)	4536	3.4 млн
7	Хутор (Q2023000)	1782	0.52 млн
8	Городское поселение в России (Q2661988)	1502	20.0 млн
9	Город (Q7930989)	1170	103.6 млн
11	Рабочий посёлок (Q20019082)	583	3.7 млн
13	Станица (Q748331)	255	1.5 млн
20	Город с населением более 100 000 человек (Q1549591)	107	58 млн
55	Город-миллионер (Q1637706)	14	32.1 млн

Выберем из табл. 6.2 строки без подсветки, то есть классы под номерами 1, 2, 3, 4, 7, 11 и 13. В дальнейшем комбинацию этих классов будем называть *сельскими поселениями*. Подсчёты показывают, что в таких поселениях в России живёт 55.6 млн человек[13].

Суммарное число жителей по каждому из классов *сельских поселений* вычислим с помощью запроса 6.9. Полученные числа представлены в табл. 6.2 в столбце «Население». Если запрос 6.9 выполняется слишком долго, вы можете закомментировать или удалить часть классов в строках 3–4 и тем самым упростить и ускорить обработку запроса.

Запрос 6.9. Список классов сельских поселений в России и суммарное число жителей по каждому классу[14]

```
1  # Number of population living in villages
2  SELECT ?village ?villageLabel (SUM(?population) AS ?count) WHERE {
3    VALUES ?village {wd:Q634099 wd:Q5084 wd:Q532 wd:Q2514025
4                     wd:Q2023000 wd:Q20019082 wd:Q748331}
5    [] wdt:P17 wd:Q159;  # settlement in the Russia
6       wdt:P1082 ?population; # has ?population
7       wdt:P31 ?village. # instance of ?village (see the list)
8   SERVICE wikibase:label{bd:serviceParam wikibase:language "ru"}
9  }
10 GROUP BY ?village ?villageLabel
```

Набор классов под номерами 8, 9, 20 и 55 в табл. 6.2 назовём *городскими поселениями*. Подсчитаем общее число жителей в городских поселениях России с помощью запроса 6.10.

Запрос 6.10. Суммарное число жителей в городских поселениях России[15]

```
1  # Number of population living in cities
2  SELECT (SUM(?population) AS ?count) WHERE {
3    SELECT DISTINCT ?city ?population WHERE {
4      VALUES ?city_list {wd:Q7930989 wd:Q2661988 wd:Q1549591 wd:Q1637706}
5      ?city wdt:P17 wd:Q159;       # settlement in Russia
6            wdt:P1082 ?population; # has ?population
7            wdt:P31 ?city_list.    # instance of ?city
8    }
9  }
```

[13] Ссылка на SPARQL-запрос: `https://w.wiki/4$D6`.

[14] Число жителей в сельских поселениях России в 2022 году представлено в табл. 6.2 в столбце «Население». Ссылка на SPARQL-запрос: `https://w.wiki/4$8w`.

[15] Число жителей в городских поселениях России в 2022 году составило 116.1 млн человек. Ссылка на SPARQL-запрос: https://w.wiki/4$D9.

Суммарное число жителей по каждому из классов *городских поселений* вычисляется аналогично запросу 6.9. Полученные числа представлены в табл. 6.2 в столбце «Население».

Таким образом, получили, что 55.6 млн человек живёт в сельских поселениях и 116.1 млн человек — в городских. Общее число составило 171.7 млн человек, при этом, по оценке Росстата, в России было 145 млн человек на 2022 год. Получили на 27 млн человек больше, и это с учётом того, что Викиданные достаточно неполные.

На наш взгляд, проблема подсчётов заключается в том, что большие города включают в себя множество небольших городков и жители этих городков считаются дважды. Команда `DISTINCT` [16] в запросе 6.10 эту проблему не решает, поскольку большой город и входящий в него маленький город — это разные объекты. Один из вариантов решения заключается в том, чтобы дополнительно проверять населённые пункты — не входят ли они в более крупные населённые пункты, а если входят, то не считать их население повторно.

[16] Команда `DISTINCT` в строке 3 запроса 6.10 нужна для учёта случаев, когда одному населённому пункту `?city` соответствует несколько значений списка `?city_list` в строке 7. То есть населённый пункт может одновременно быть экземпляром (P31) и города (Q7930989), и города-миллионера (Q1637706).

Команда `DISTINCT` во внутреннем цикле `SELECT` (строки 3–8) позволяет получить эти города (`?city`) без дублирования. Если убрать слово `DISTINCT`, то численность населения вырастет со 116.1 млн до 187.2 млн человек.

Отметим, что если сложить численность разных типов городов в табл. 6.2, то получим $20 + 103.6 + 58 + 32.1 = 213.7$. Очевидно, что списки разных типов городских поселений пересекаются, поэтому получилось такое большое число.

Профессии, представленные в Викиданных

Получим список профессий и количество работников, имеющих российское гражданство. Этим запросом 6.11 мы ищем такие объекты Викиданных, у которых есть свойство «гражданство (P27)» и значением этого свойства является объект «Россия (Q159)». Очевидно, что объект, имеющий свойство «гражданство», будет человеком, а в терминологии вики-проектов — персоной.

Запрос 6.11. Список профессий или должностей граждан России[17]

```
1 # List of occupation or job citizens of Russia
2 SELECT DISTINCT ?job ?jobLabel (COUNT(?hum) AS ?count) WHERE {
3   ?hum wdt:P27 wd:Q159; # citizen of Russia
4        wdt:P106 ?job. # has occupation or job
5   SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en"}
6 }
7 GROUP BY ?job ?jobLabel
8 ORDER BY ?count
```

[17] Получено: 2023 профессии на 2022 год. Ссылка на SPARQL-запрос: https://w.wiki/4daC.

В табл. 6.3 представлены такие профессии из результата запроса 6.11, которые могут соответствовать каким-либо научными направлениям.

Таблица 6.3. Научные направления и число отечественных специалистов, 2022 год

№	Профессия (объект)	Число людей
1	Физик (Q169470)	991
2	Историк (Q201788)	913
3	Экономист (Q188094)	880
4	Математик (Q170790)	857
5	Инженер (Q81096)	558
7	Химик (Q593644)	439
8	Врач (Q39631)	342
9	Юрист (Q185351)	330
10	Биолог (Q864503)	222

Найти учёных среди людей

Есть два способа получения списка учёных. Первый: у большинства учёных в Викиданных заполнено свойство «научная степень (P512)». Запросим в Викиданных россиян, имеющих такое свойство (запрос 6.12), и получим список отечественных учёных.

Отметим, что если учёный жил и в СССР, и в современной России, то у персоны в поле «гражданство (P27)» будут указаны обе страны. Благодаря команде `DISTINCT` (строка 2 запроса 6.12) такой учёный посчитается ровно один раз. Поэтому удаление команды `DISTINCT` почти удваивает число отечественных учёных.

Запрос 6.12. Количество людей из России с учёной степенью[18]

```
1 SELECT (COUNT(DISTINCT ?hum) AS ?human_count) WHERE {
2         # Russian Empire, Soviet Union and Russia
3   VALUES ?ruCountries {wd:Q34266 wd:Q15180 wd:Q159}
4   ?hum wdt:P512 ?academic_degree;  # holds academic degree
5        wdt:P27 ?ruCountries.} # lives (lived) in Russia
```

Второй способ получения списка учёных — это проверка свойства «участник организации (P463)». У человека в этом свойстве может быть указана одна из научных или образовательных организаций[19]. С помощью запроса 6.13 получим список людей, имеющих отношение к этим организациям, их мы и назовём учёными.

Первым способом мы нашли почти в 6 раз больше учёных. Этим способом и воспользуемся далее.

Запрос 6.13. Количество людей из академий в России[20]

```
1 SELECT (COUNT(DISTINCT ?hum) AS ?human_count) WHERE {
2   VALUES ?ruCountries {wd:Q34266 wd:Q15180 wd:Q159}
3   VALUES ?class_academy {wd:Q414147 wd:Q955824 wd:Q162633 wd:Q31855 wd:Q2385804 wd:Q83172}
4   ?hum wdt:P27 ?ruCountries;  # citizenship of one of countries
5        wdt:P463 [ wdt:P31 ?class_academy].} # member of ?academy
```

[18] Получено: 24 797 учёных на 2022 год. Ссылка на SPARQL-запрос: `https://w.wiki/537r`

[19] Мы проверяли наличие следующих научных и образовательных организаций в запросе 6.13: академия наук (Q414147), научная ассоциация (Q955824), научное общество (Q748019), академия (Q162633), научно-исследовательский институт (Q31855), образовательное учреждение (Q2385804).

[20] Получено: 4215 учёных на 2022 год. Ссылка на SPARQL-запрос: `https://w.wiki/539x`

Построение диаграммы с научными направлениями и годами рождения учёных, родившихся в сельских поселениях

Сначала с помощью запроса 6.14 построим список российских учёных, имеющих научную степень.

Запрос 6.14. Список российских учёных с научной степенью[21]

```
# List of peoples from Russia with academic degree
SELECT DISTINCT ?hum ?humLabel WHERE {
      # Russian Empire, Soviet Union and Russia
    VALUES ?state { wd:Q34266 wd:Q15180 wd:Q159}
    ?hum wdt:P27 ?state; # citizenship
         wdt:P512 []; # has academic degree
    SERVICE wikibase:label{bd:serviceParam wikibase:language "ru"}
}
```

Теперь с помощью запроса 6.15 построим диаграмму числа рождений учёных в сёлах с разбивкой по десятилетиям и научным направлениям (рис. 6.5). Этот запрос интересен тем, что результаты одного запроса `SELECT` мы передаём на вход следующему запросу с помощью конструкции `WITH { } AS ... INCLUDE`.

Запрос 6.14 как раз и является первой частью запроса запроса 6.15 (строки 5–12). Полученные в этой части учёные далее в строке 21 отфильтровываются по месту рождения, и те, кто родился в сельских поселениях (строки 17 и 21), участвуют в построении диаграммы (рис. 6.5).

Рис. 6.5 показывает количество учёных по родам деятельности, родившихся в сельских поселениях. В строках 18–19 запроса 6.15 в списке `?jobs` перечислены 10 профессий этих учёных. Каждой из этих профессий на рис. 6.5 соответствует полоска определённого цвета, см. легенду внизу диаграммы.

При наведении курсора на цветную полоску на рис. 6.5 всплывает окошко с информацией о названии профессии, десятилетии и числе родившихся в сёлах будущих учёных. Мы видим, что в 1940-е годы в советских сёлах родились 24 известных Викиданным экономиста.

[21] Получено: 24 797 учёных на 2022 год. Ссылка на SPARQL-запрос: https://w.wiki/53wx

Попробуйте изменить запрос 6.15 так, чтобы не задавать список профессий явно, а получить список *всех* профессий учёных автоматически. Вероятно, это возможно с помощью дополнительного предварительного запроса `SELECT`.

Запрос 6.15. Построение диаграммы отечественных учёных, родившихся в сёлах, с разбивкой по десятилетиям и занятиям[22]

```
#defaultView:BarChart
SELECT DISTINCT (SAMPLE(?year_lapse) AS ?year) (COUNT(?hum) AS ?count)
       (GROUP_CONCAT(DISTINCT ?jobsLabel; SEPARATOR=", ") AS ?job)
WITH {
  SELECT DISTINCT ?hum
  WHERE {
          # Russian Empire, Soviet Union and Russia
          VALUES ?state { wd:Q34266 wd:Q15180 wd:Q159}
          ?hum wdt:P27 ?state; # citizenship
               wdt:P512 []; # has academic degree
  }
} AS %result
WHERE {
  INCLUDE %result

  # rural settlement of Russia, hamlet, village...
  VALUES ?placeb {wd:Q634099 wd:Q5084 wd:Q532 wd:Q2514025 wd:Q2023000 wd:20019082 wd:748331}
  VALUES ?jobs {wd:Q169470 wd:Q201788 wd:Q188094 wd:Q170790 wd:Q81096 wd:Q1650915 wd:Q593644
                wd:Q39631 wd:Q185351 wd:Q864503}
  ?hum wdt:P106 ?jobs; # occupation
       wdt:P19 [wdt:P31 ?placeb]; # place of birth
       wdt:P569 ?birthday.
                                # count for each 10 years
  BIND(str(FLOOR(YEAR(?birthday)/10)*10) AS ?year_lapse).
  FILTER(YEAR(?birthday) > 1850 && YEAR(?birthday) < 2000).
  SERVICE wikibase:label{bd:serviceParam wikibase:language"ru".
                         ?jobs rdfs:label ?jobsLabel.}
}
GROUP BY ?year_lapse ?jobsLabel
ORDER BY ?year_lapse
```

[22] Построена диаграмма (рис. 6.5) по данным на 2022 год. Ссылка на SPARQL-запрос: https://w.wiki/55VK.

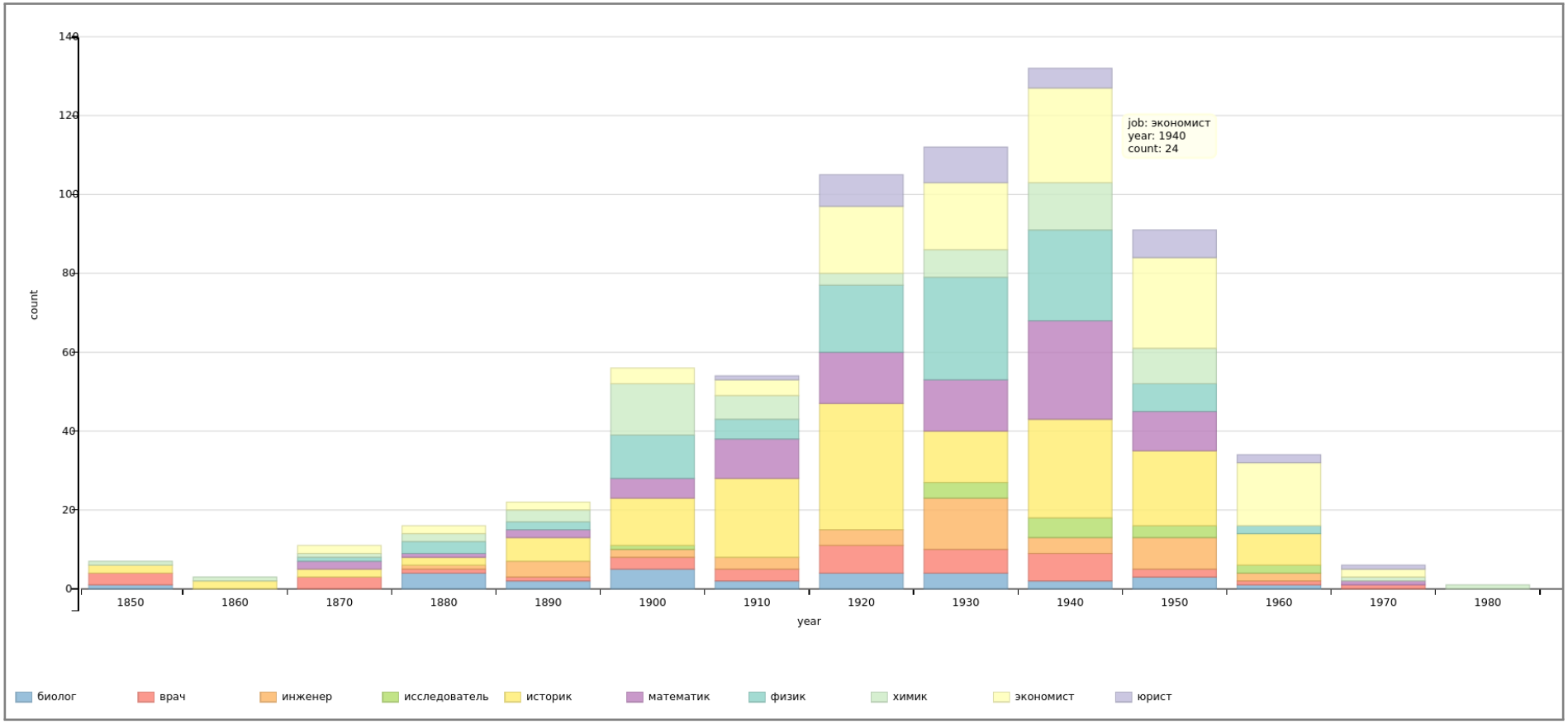


Рис. 6.5. Количество отечественных учёных 10 научных направлений, родившихся в сельских поселениях с 1850-х по 1980-е годы, 2022 год

Изменим запрос 6.15 так, чтобы были представлены не 10 профессий, а учёные всех направлений. Получили запрос 6.16, который строит диаграмму количества российских и советских учёных, родившихся в сельских поселениях в 1850–1980 годах (рис. 6.6).

Обратим внимание на разницу в пиках на этих диаграммах. При фильтрации по 10 направлениям максимум рождений учёных в сёлах (132 человека) пришёлся на 1940-е годы (рис. 6.5). Без фильтра максимум сместился на 1930-е годы, а именно: в сёлах родились 685 малышей — будущих известных учёных (рис. 6.6).

Запрос 6.16. Построение диаграммы отечественных учёных *всех направлений*, родившихся в сёлах с разбивкой по десятилетиям[23]

```
#defaultView:BarChart
SELECT DISTINCT (SAMPLE(?year_lapse) AS ?year) (COUNT(?hum) AS ?count)
WITH {
  SELECT DISTINCT ?hum WHERE {
          # Russian Empire, Soviet Union and Russia
          VALUES ?state { wd:Q34266 wd:Q15180 wd:Q159}
          ?hum wdt:P27 ?state; # citizenship
               wdt:P512 []; # has academic degree
  }
} AS %result
WHERE {
  INCLUDE %result

  # rural settlement of Russia, hamlet, village...
  VALUES ?placeb {wd:Q634099 wd:Q5084 wd:Q532 wd:Q2514025 wd:Q2023000 wd:20019082 wd:748331}
      ?hum wdt:P106 ?job; # occupation
           wdt:P19 [wdt:P31 ?placeb]; # place of birth
           wdt:P569 ?birthday.
                              # count for each 10 years
  BIND(str(FLOOR(YEAR(?birthday)/10)*10) AS ?year_lapse).
  FILTER(YEAR(?birthday) > 1850 && YEAR(?birthday) < 2000).
  SERVICE wikibase:label{bd:serviceParam wikibase:language "ru"}
}
GROUP BY ?year_lapse ?jobsLabel
ORDER BY ?year_lapse
```

[23] Построена диаграмма (рис. 6.6) по данным на 2022 год. Ссылка на SPARQL-запрос: https://w.wiki/55WY.

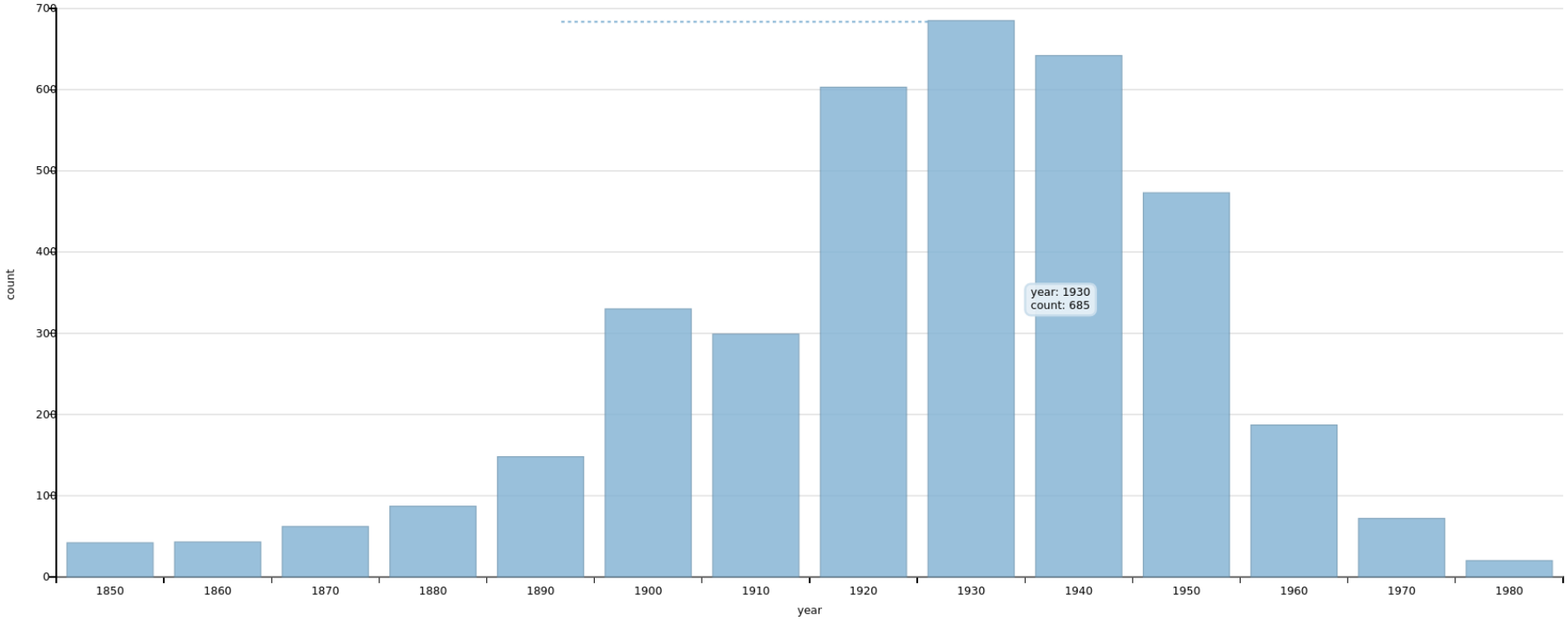


Рис. 6.6. Количество отечественных учёных *всех направлений*, родившихся в сельских поселениях с 1850-х по 1980-е годы, 2022 год

Построение диаграммы числа учёных, родившихся в городских поселениях, и сравнение диаграмм

Возьмём из табл. 6.2 объекты, соответствующие городским поселениям, и заменим в строке 16 запроса 6.16 сельские поселения на городские. Получим запрос 6.17 для построения диаграммы с числом отечественных учёных всех направлений, родившихся в городских поселениях, с разбивкой по десятилетиям (рис. 6.7).

Запрос 6.17. Фрагмент запроса построения диаграммы числа отечественных учёных *всех направлений*, родившихся в городских поселениях, с разбивкой по десятилетиям[24]

```
14  ...
15  # city/town, urban settlement in Russia, big and million city
16  VALUES ?placeb {wd:Q7930989 wd:Q2661988 wd:Q1549591 wd:Q1637706}
17  ...
```

Диаграмма на рис. 6.7 показывает количество учёных по родам деятельности, родившихся в городских поселениях. На ней чётко виден провал рождаемости в десятилетие Второй мировой войны, а именно: 5700 человек в 1940-е годы, что на 28 % меньше, чем в 1930-е годы (7900 человек), и на 16 % меньше, чем в 1950-е годы (6800 будущих учёных).

Сравним пик рождаемости по сёлам (рис. 6.6) и городам (рис. 6.7), который на обеих диаграммах приходится на 1930-е годы. В городах СССР в это десятилетие родилось 7900 учёных, а в сёлах и деревнях — 700 учёных.

Если последовательно по десятилетиям с 1930-х по 1980-е годы сравнить, во сколько раз в городах рождалось больше учёных, чем в сёлах, то получим, что в городах по сравнению с сёлами учёных рождалось больше[25]:

- в 11 раз (1930-е),
- в 9 раз (1940-е),
- в 14 раз (1950-е),
- в 22 раза (1960-е),
- в 35 раз (1970-е и 1980-е годы).

Таким образом, в городах рождалось в десятки раз больше учёных, чем в сёлах. И этот разрыв во второй половине XX века в нашей стране только увеличивается.

[24] Построена диаграмма (рис. 6.6) по данным на 2022 год. Ссылка на SPARQL-запрос: https://w.wiki/58tL.

[25] Эти численные результаты получены по Викиданным на 2022 год.

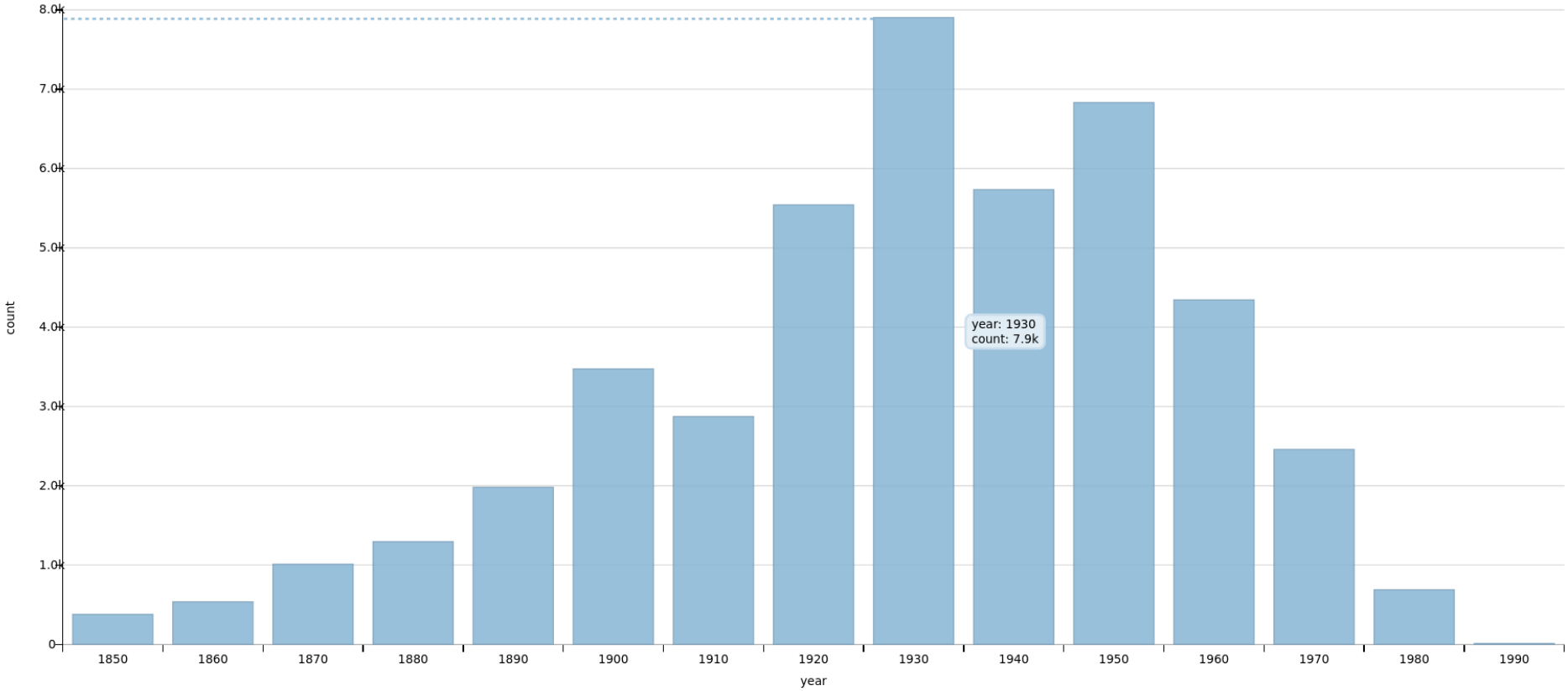


Рис. 6.7. Количество отечественных учёных, родившихся в городских поселениях с 1850-х по 1990-е годы, 2022 год

7. От малых городов до городов-миллионеров

Глава посвящена исследованию различных типов городов, соответствующих четырём объектам Викиданных, — «малый город», «город», «большой город» и «город-миллионер». В ходе исследования с использованием SPARQL-запросов получены данные о количестве экземпляров исследуемых объектов, а также рассмотрены вопросы, связанные со свойствами population (численность населения) и sister city (город-побратим) этих объектов Викиданных. Подсчитана и проанализирована численность населения разных типов городов; найдено число городов без побратимов; построен список городов, упорядоченных по числу побратимов; найдено число городов с определённым числом побратимов; определены страны с наибольшим числом побратимов; найдены ближайшие соседи России по числу городов-побратимов. В заключение дана оценка полноты данных, представленных в Википедии и Викиданных, и перечислены проблемы и сложности, возникшие при изучении разных типов городов в этих проектах.

Списки разных типов городов

Ниже представлены исследуемые объекты и SPARQL-запросы для получения списков экземпляров исследуемых объектов:

- «малый город» или town (Q3957), запрос 7.1;
- «город» или city (Q515), запрос 7.2;
- «большой город» или big city (Q1549591), запрос 7.3;
- «город-миллионер» или city with millions of inhabitants (Q1637706), запрос 7.4.

Дополнительно составлен запрос для нахождения списка экземпляров объектов разных типов городов (запрос 7.5).

Запрос 7.1. Экземпляры объекта «малый город»[1]

```
SELECT ?city ?cityLabel WHERE {
  ?city wdt:P31 wd:Q3957. # instances of "town"
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

Среди отечественных «городов» в Викиданных больше всего свойств у Новороссийска (Q15760) — 31. Лидером по «городам» всего мира является Сингапур (Q334) — 104 свойства.

Запрос 7.2. Экземпляры объекта «город»[2]

```
SELECT ?city ?cityLabel WHERE {
  ?city wdt:P31 wd:Q515. # instances of "city"
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

Среди отечественных «больших городов» в Викиданных больше всего свойств у Москвы (Q649) — 76. Лидером по «большим городам» всего мира снова является Сингапур (Q334) — 104 свойства.

Запрос 7.3. Экземпляры объекта «большой город»[3]

```
SELECT ?city ?cityLabel WHERE {
  ?city wdt:P31 wd:Q1549591. # instances of "big city"
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

[1] Получено: 13 800 «малых городов» в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/t4o.

[2] Получено: 20 800 «городов» в 2017 году, 9260 «городов» в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/t4q.

[3] Получено: 198 «больших городов» в 2017 году, 3075 «больших городов» в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/t4r.

Запрос 7.4. Экземпляры объекта «город-миллионер»[4]

```
SELECT ?city ?cityLabel WHERE {
  ?city wdt:P31 wd:Q1637706. # instances of "city 1000000+"
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

[4] Получено: 616 «городов-миллионеров» в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/t4t.

Запрос 7.5. Экземпляры объектов разных типов городов[5]

```
SELECT ?city ?cityLabel WHERE { # Selecting items which are ...
  { ?city wdt:P31 wd:Q3957 } UNION # instances of "town"
  { ?city wdt:P31 wd:Q515 } UNION # OR instances of "city"
  { ?city wdt:P31 wd:Q1549591 } UNION # OR instances of "big city"
  { ?city wdt:P31 wd:Q1637706 } # OR instances of "city 1000000+"
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru". }
}
```

[5] Получено: 26 751 экземпляр объектов разных типов городов в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/t4v.

Численность населения

Ниже представлены SPARQL-запросы для нахождения суммарной численности населения по всем типам городов:

- «малый город» или town (Q3957), запрос 7.6;
- «город» или city (Q515), запрос 7.7;
- «большой город» или big city (Q1549591), запрос 7.8;
- «город-миллионер» или city with millions of inhabitants (Q1637706), запрос 7.9.

Какие из городов названы в честь географических объектов?

- Тольятти
- Тула
- Черняховск
- Курильск
- Вологда
- Обнинск

См. ответ на с. 165.

Запрос 7.6. Численность населения «малых городов»[6]

```
# Selecting total population of items which are ...
SELECT (SUM(?population_city) as ?sum) WHERE {
  SELECT (MAX(xsd:integer(REPLACE(STR(?population),"\\.","")))
            as ?population_city) ?city WHERE {
    ?city wdt:P31 wd:Q3957. # instances of "town"
    ?city wdt:P1082 ?population # with filled property "population"
  }
  GROUP BY ?city
}
```

[6] Получено: sum = 53.3 млн чел. в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/jgX.

Запрос 7.7. Численность населения «городов»[7]

```
1 # Selecting total population of items which are ...
2 SELECT (SUM(?population_city) as ?sum) WHERE {
3   SELECT (MAX(xsd:integer(REPLACE(STR(?population),"\\.","")))
4             as ?population_city) ?city WHERE {
5     ?city wdt:P31 wd:Q515. # instances of "city"
6     ?city wdt:P1082 ?population # with filled property "population"
7   }
8   GROUP BY ?city
9 }
```

Для приведения значений свойства population к единообразному виду можно использовать различные функции обработки строк, например REPLACE. Функция REPLACE здесь в строке 3 удаляет в тексте точки, например строка «1.000.000» преобразуется в «1000000».

[7] Получено: sum = 1133.56 млн чел. в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/jgY.

Запрос 7.8. Численность населения «больших городов»[8]

```
# Selecting total population of items which are ...
SELECT (SUM(?population_city) as ?sum) WHERE {
  SELECT (MAX(xsd:integer(REPLACE(STR(?population),"\\.","")))
            as ?population_city) ?city WHERE {
    ?city wdt:P31 wd:Q1549591. # instances of "big city"
    ?city wdt:P1082 ?population # with filled property "population"
  }
  GROUP BY ?city
}
```

[8] Получено: sum = 2538.49 млн чел. в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/jga.

Поясним в последних трёх скриптах (запросы 7.6–7.8) использование функции MAX. Численность населения постоянно изменяется, поэтому город может иметь несколько значений свойства population, соответствующих различным годам. Для выбора одного из значений свойства можно использовать агрегирующие функции, например MIN, MAX, AVG.

Отметим, что последнее по времени значение свойства, например «численность населения», обычно имеет пометку *preferred value*. Это означает, что при обращении к свойству будет возвращено именно такое «предпочтительное» значение, например число жителей по данным последней переписи. В этом случае агрегирующие функции не нужны.

Запрос 7.9. Численность населения «городов-миллионеров»[9]

```
# Selecting total population of items which are ...
SELECT (SUM(?population_city) as ?sum) WHERE {
  SELECT (MAX(xsd:integer(REPLACE(STR(?population),"\\.","")))
            as ?population_city) ?city WHERE {
    ?city wdt:P31 wd:Q1637706. # instances of "city 1000000+"
    ?city wdt:P1082 ?population # with filled property "population"
  }
  GROUP BY ?city
}
```

В разных странах используются различные символы в качестве разделителей разрядов, например: точка, запятая или пробел. Как результат, варианты представления значения свойства population также могут быть различны. Проблемы возникают при использовании точки, поскольку в Викиданных этот символ является разделителем целой и десятичной частей числа. Для устранения неоднозначности необходимо использовать функцию REPLACE, позволяющую удалить указанный символ (запросы 7.6, 7.7, 7.8, 7.9). Данное преобразование не влияет на само значение, так как численность населения априори является целым числом, а разделители разрядов используются исключительно для упрощения чтения.

В табл. 7.1 приведена сводная информация о численности населения разных типов городов, а также рассчитана доля населения, приходящегося на каждый тип города, от общего населения Земли, достигшего примерно 7.8 млрд человек на 2020 год[10]. Согласно Викиданным, почти три четверти населения планеты проживает в городах.

[9] Получено: sum = 2118.39 млн чел. в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/jgb.

Таблица 7.1. Численность населения разных типов городов, 2020 год

Тип города	Численность населения (млн чел.)	% от общего населения Земли
Малый город	53.30	0.7
Город	1133.56	14.5
Большой город	2538.49	32.5
Город-миллионер	2118.39	27.1
Всего	5843.74	74.8

[10] *Салькова А.* Рост не бесконечен: что ждет население планеты к концу века. 2020. URL: https://www.gazeta.ru/science/2020/07/16_a_13154515.shtml.

Города-побратимы

Города-побратимы (породнённые города) — это города различных государств, установившие между собой постоянные дружественные связи с целью укрепления международных отношений в областях культуры, экономики, создания и управления городской инфраструктурой, функционирования гражданского общества и т. д.[11]

[11] Международная ассоциация «Породнённые города». URL: https://bit.ly/3qs2JUt.

Сколько городов обходятся без городов-побратимов?

Запрос 7.10 подсчитывает число городов, не имеющих побратимов.

Запрос 7.10. Количество городов, не имеющих побратимов[12]

```
1 # Counting items which are ...
2 SELECT (COUNT(?city) as ?count) WHERE {
3   { ?city wdt:P31 wd:Q3957 } UNION # instances of "town"
4   { ?city wdt:P31 wd:Q515 } UNION # OR instances of "city"
5   { ?city wdt:P31 wd:Q1549591 } UNION # OR instances of "big city"
6   { ?city wdt:P31 wd:Q1637706 }   # OR instances of "city 1000000+"
7   FILTER NOT EXISTS { ?city wdt:P190 [] } # with unfilled "sister city"
8 }
```

[12] Получено: 21 479 городов в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/jgv.

Всего городов четырёх типов на 2020 год Викиданным известно 26 751 (запрос 7.5). Таким образом, только для 20 % городов известны города-побратимы.

Обратите внимание на то, что в строках 3–6 запроса 7.10 части конструкции UNION заключены в фигурные скобки. В противном случае SPARQL-запрос не будет скомпилирован.

Упорядоченный список городов по числу побратимов

Далее представлен запрос 7.11 для получения списков всех городов и запрос 7.12 для получения городов России, при этом все они упорядочены по числу побратимов.

Запрос 7.11. Упорядоченный список городов по числу побратимов[13]

```
# Counting sister cities of cities which are ...
SELECT ?city ?cityLabel (COUNT(?sister) AS ?sisterCount) WHERE {
  { ?city wdt:P31 wd:Q3957 } UNION # instances of "town"
  { ?city wdt:P31 wd:Q515 } UNION # OR instances of "city"
  { ?city wdt:P31 wd:Q1549591 } UNION # OR instances of "big city"
  { ?city wdt:P31 wd:Q1637706 } # OR instances of "city 1000000+"
  ?city wdt:P190 ?sister. # with filled property "sister city"
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru". }
}
GROUP BY ?city ?cityLabel # Grouping by city
ORDER BY DESC(?sisterCount) # Sorting by number of sister cities
```

[13] Получено: 4046 городов мира с побратимами в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/sY4.

Запрос 7.12. Упорядоченный список городов России по числу побратимов[14]

```
# Counting sister cities of cities which are ...
SELECT ?city ?cityLabel (COUNT(?sister) AS ?sisterCount) WHERE {
  VALUES ?cityTypes {wd:Q3957 wd:Q515 wd:Q1549591 wd:Q1637706}
  ?city wdt:P31 ?cityTypes. # instances of different types of cities
  ?city wdt:P17 wd:Q159. # belonging to Russia
  ?city wdt:P190 ?sister. # with filled property "sister city"
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru". }
}
GROUP BY ?city ?cityLabel # Grouping by city
ORDER BY DESC(?sisterCount) # Sorting by number of sister cities
```

[14] Получено: 82 города России с побратимами в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/t4A.

С культурной столицей России желающих дружить нашлось больше (230 побратимов), чем с официальной столицей (134 побратима) на 2020 год. Почти одинаковое число побратимов делят Омск (58), Волгоград (56) и Калининград (54). У Петрозаводска, Перми, Владимира и Белгорода по 14 побратимов.

Число городов с определённым числом побратимов

Ниже представлены SPARQL-запросы для получения числа городов (N) с определённым числом побратимов (S) для всех городов, известных Викиданным (запрос 7.13), и для городов России (запрос 7.14).

Строка `#defaultView:LineChart` в начале скрипта 7.13 не является комментарием. Это специальная команда для построения диаграммы (рис. 7.1) на платформе Wikidata Query Service, в которой мы пишем скрипты для обработки Викиданных.

Запрос 7.13. Число городов с определённым числом побратимов[15]

```
# Count number of cities having sister cities and number of sister cities themselves
#defaultView:LineChart
SELECT ?sisterCount (COUNT(?sisterCount) AS ?FreqNSister) WHERE {
  { # Count sister cities of cities which are ...
    SELECT (COUNT(?sister) AS ?sisterCount) WHERE {
      VALUES ?cityTypes {wd:Q3957 wd:Q515 wd:Q1549591 wd:Q1637706}
      ?city wdt:P31 ?cityTypes. # instances of different types of cities
      ?city wdt:P190 ?sister. # with filled property "sister city"
    }
    GROUP BY ?city
  }
}
GROUP BY ?sisterCount # Group by number of sister cities
ORDER BY DESC(?sisterCount) # Order by number of sister cities
```

Результаты SPARQL-запроса для всех городов отображены на рис. 7.1. Максимальное значение N (1224 города) наблюдается при S, равном единице, затем следует резкое уменьшение числа городов с увеличением числа имеющихся у них побратимов. На рис. 7.2 представлены те же данные, но в логарифмической шкале. Так, чуть более 4 тыс. городов (4046) побратались хотя бы с одним городом, из них:

- 32 % (1314 городов) связаны братскими отношениями более чем с пятью городами;
- 18 % (728 городов) имеют как минимум 11 городов-побратимов;
- 9 % (345 городов) подружились более чем с 20 городами;
- 2 % (94 города) имеют от 50 побратимов.

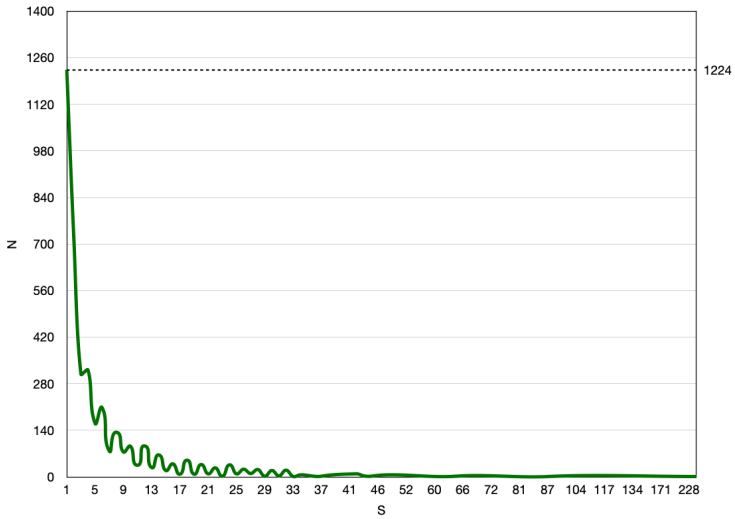


Рис. 7.1. Зависимость числа городов всего мира (N) от числа имеющихся у этих городов побратимов (S), 2020 год

[15] Получено: 90 вариантов числа братских городов в мире в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/pM9.

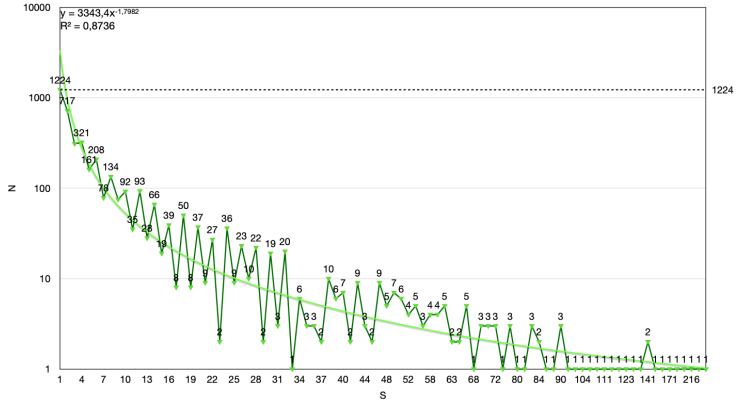


Рис. 7.2. Зависимость числа городов мира (N) от числа имеющихся побратимов (S) на логарифмической шкале, 2020 год

На основании построенного тренда можно сделать предположение, что зависимость числа городов от числа имеющихся у этих городов побратимов имеет распределение, близкое к степенному.

Запрос 7.14. Число городов России с определённым числом побратимов[16]

```
#defaultView:LineChart
# Count number of cities having sisterCount sister cities
# and number of sister cities themselves
SELECT ?sisterCount (COUNT(?sisterCount) AS ?FreqNSister) WHERE {
  { # Count sister cities of cities which are ...
        SELECT (COUNT(?sister) AS ?sisterCount) WHERE {
            VALUES ?cityTypes {wd:Q3957 wd:Q515 wd:Q1549591 wd:Q1637706}
            ?city wdt:P31 ?cityTypes. # instances of different types of cities
            ?city wdt:P17 wd:Q159.    # belonging to Russia
            ?city wdt:P190 ?sister.   # with filled property "sister city"
    }
    GROUP BY ?city # Group list by city
  }
}
GROUP BY ?sisterCount # Group by number of sister cities
ORDER BY DESC(?sisterCount) # Order by number of sister cities
```

Ситуация с отечественными городами представлена на рис. 7.3. Чуть менее ста городов России (82 города) побратались хотя бы с одним городом, из них только 48 % (39 городов) связаны братскими отношениями более чем с пятью городами.

Какие из городов были основаны более 400 лет назад: Москва, Саров, Казань, Астрахань, Самара, Воронеж?

См. ответ на с. 165.

[16] Получено: 24 варианта числа братских городов в России в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/pMA.

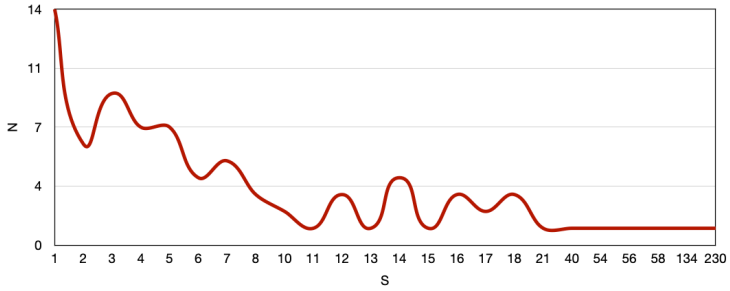


Рис. 7.3. Зависимость числа городов России (N) от числа побратимов (S), 2020 год

У какой страны больше всего побратимов?

Запрос 7.15 позволяет построить пузырьковую диаграмму (рис. 7.4), показывающую соотношение количества городов-побратимов между странами.

Запрос 7.15. Пузырьковая диаграмма стран мира, упорядоченных по количеству городов-побратимов[17]

```
#defaultView:BubbleChart
# Ordered list of countries by number of sister cities
SELECT DISTINCT ?country ?countryLabel (COUNT(?sister) as ?sisterCount) WHERE
{ VALUES ?cityTypes {wd:Q3957 wd:Q515 wd:Q1549591 wd:Q1637706}
  ?city wdt:P31 ?cityTypes;     # instance of some type of city
        wdt:P17 ?country;       # belongs to country
        wdt:P190 ?sister.       # has a sister city
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
GROUP BY ?country ?countryLabel
ORDER BY DESC(?sisterCount)
```

Запрос 7.16. Список стран, имеющих города-побратимы с городами Германии[18]

```
SELECT ?country ?countryLabel (COUNT(DISTINCT ?sister) as ?sisterCount) WHERE
{ VALUES ?cityTypes {wd:Q3957 wd:Q515 wd:Q1549591 wd:Q1637706}
  ?city wdt:P31 ?cityTypes; # instance of one of the city types
        wdt:P17 wd:Q183;    # belongs to Germany
        wdt:P190 ?sister.   # has a sister city
  ?sister wdt:P17 ?country. # sister city belongs to some country
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }}
GROUP BY ?country ?countryLabel
ORDER BY DESC(?sisterCount)
```

На пузырьковой диаграмме (рис. 7.4) размер шарика соответствует числу побратимов у городов страны. На 2020 год больше всего побратимов имела Германия (1375 городов). В 2022 году у Германии было уже 1703 побратима, а на первое место вышел Китай с 2542 побратимами. Полный список стран, с которыми Германия имеет города-побратимы, можно получить с помощью запроса 7.16.

[17] Получено: 208 стран в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/5At7.

Рис. 7.4. Пузырьковая диаграмма стран мира по числу побратимов у городов страны, 2020 год

[18] Получено: 93 страны в 2020 году и 100 стран в 2022 году. На первом месте находится Франция, у неё 268 городов, имеющих побратимов в Германии в 2022 году. SPARQL-запрос: https://w.wiki/5As4.

В табл. 7.2 приведен список из первых 10 стран, имеющих наибольшее количество побратимов с городами Германии на 2020 и 2022 годы.

Ближайшие соседи России по числу городов-побратимов

Запрос 7.17 позволяет получить список стран, с которыми у России есть города-побратимы. Результаты запроса представлены на рис. 7.5.

У России нашлось больше двадцати городов-побратимов с такими странами, как США (46), Китай (46), Германия (44), Украина (28), Болгария (25), Польша (24), Франция (23) и Италия (22).

Запрос 7.17. Ближайшие соседи России по числу побратимов[19]

```
#defaultView:Map
SELECT ?country ?countryLabel ?sisterCount ?shape ?layer WHERE {
  {
    SELECT ?country ?countryLabel (COUNT(DISTINCT ?sister) as ?sisterCount)
    WHERE {
        VALUES ?cityTypes {wd:Q3957 wd:Q515 wd:Q1549591 wd:Q1637706}
        ?city wdt:P31 ?cityTypes.   # instances of different types of cities
        ?city wdt:P17 wd:Q159.      # city belongs to Russia
        ?city wdt:P190 ?sister.     # city has "sister city"
        ?sister wdt:P17 ?country.   # which belongs to "country"
        FILTER(?country NOT IN(wd:Q159)) # except the Russia
        SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
    }
    GROUP BY ?country ?countryLabel
    ORDER BY DESC(?sisterCount)
  }
  OPTIONAL {?country wdt:P3896 ?shape.} # country has "geoshape"
  BIND(
      IF(?sisterCount < 5, "<5",
    IF(?sisterCount <= 10, "5-10",
    IF(?sisterCount <= 20, "11-20",
    IF(?sisterCount <= 30, "21-30",
    IF(?sisterCount <= 40, "31-40",
    ">40"))))) AS ?layer). }
```

Таблица 7.2. Страны с наибольшим числом побратимов из Германии, 2020 и 2022 годы

№	Страна	2020	2022
1	Франция (Q142)	247	268
2	Германия (Q183)	195	204
3	Великобритания (Q145)	120	125
4	Италия (Q38)	86	91
5	Польша (Q36)	81	83
6	США (Q30)	60	63
7	Австрия (Q40)	41	42
8	Россия (Q159)	39	42
9	Венгрия (Q28)	39	41
10	Бельгия (Q31)	33	32

Какому городу России принадлежит этот флаг?

См. ответ на с. 166.

[19] Получено: 102 страны в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/q5Z.

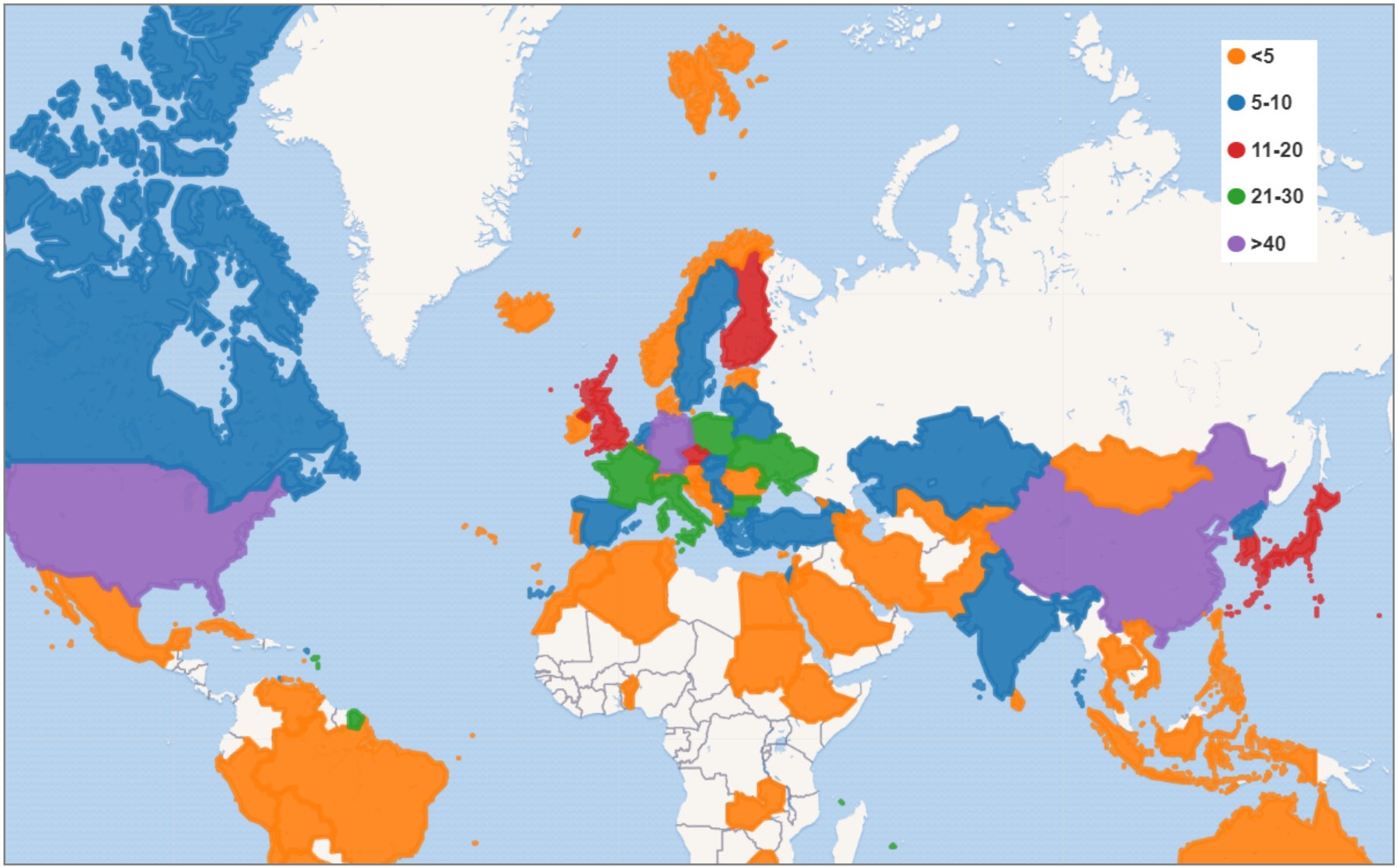


Рис. 7.5. Карта ближайших соседей России по числу городов-побратимов, 2020 год

Полнота и недостатки Викиданных: дублирование объектов

Городом принято называть крупный населённый пункт, жители которого, как правило, не заняты сельским хозяйством. При этом разные страны используют различные критерии при наделении поселений статусом города, основным из которых является численность населения. Некоторые страны и вовсе не применяют понятие города. Так, во Франции используется только одна географическая единица подобного рода — коммуна, вне зависимости от количества проживающих в ней людей и рода их деятельности. Поэтому чётко определить, какой населённый пункт относить к городам, а какой нет, может быть затруднительно.

На практике некоторые объекты Викиданных могут одновременно являться экземплярами городов разных типов. Например, Шанхай (Q8686) отнесен к трём исследуемым объектам: city (Q515), big city (Q1549591), city with millions of inhabitants (Q1637706). Такое множественное присваивание влияет на результаты SPARQL-запросов. Например, при использовании конструкции UNION[20] в результатах запроса 7.5 объект `Шанхай` встречается три раза вместо одного.

Викиданные имеют механизм наследования, выражающийся в свойстве subclass of. Заключается этот механизм в том, что если объект является экземпляром `big city`, то он является и экземпляром объекта `city`, так как `big city` — это подкласс `city`. Таким образом, описанную выше ситуацию с `Шанхаем` можно разрешить, оставив только один класс, например `city with millions of inhabitants`. Стоит отметить, что замена конструкции с использованием `UNION` (запрос 7.18) на конструкцию с учётом подклассов (запрос 7.19) неэквивалентна. Рассмотренный ранее `Шанхай` встречается в новой выборке даже четыре раза. Дело в том, что, помимо части исследуемых классов, есть и другие классы, наследуемые от `city`. Например, lost city (Q2974842), free imperial city (Q57318), autonomous city (Q1094397) и даже ideal city (Q1656724).

Запрос 7.18. Пример использования конструкции UNION[21]

```
SELECT ?city ?cityLabel WHERE {
    { ?city wdt:P31 wd:Q515 } UNION     # instances of "city"
    { ?city wdt:P31 wd:Q1549591 } UNION # OR instances of "big city"
    { ?city wdt:P31 wd:Q1637706 }       # OR instances of "city 1000000+"
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

Вопрос отнесения населённого пункта к сельскому или городскому типу также решается в подразделе «Список сельских и список городских типов поселений в России» на с. 57.

[20] Более подробно о команде `UNION` см. в Английском Викиучебнике SPARQL по ссылке: https://en.wikibooks.org/wiki/SPARQL/UNION.

[21] Ссылка на SPARQL-запрос: https://w.wiki/k5T.

Запрос 7.19. Пример использования конструкции с подклассами[22]

```
SELECT ?city ?cityLabel WHERE {
    ?city wdt:P31/wdt:P279* wd:Q515 # instances of city subclasses
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

Вероятно, в связи с неоднозначностью критериев присвоения статуса города были созданы подклассы для конкретных стран — это city in Chile (Q25412763), city in Cyprus (Q29556224), city of Japan (Q494721) и т. д. Не обошла стороной эта тенденция и города России: существует объект Викиданных «Городское поселение в России»[23].

По данным переписи населения России 2010 года[24] и переписи населения Крыма 2014 года[25], число городов составило соответственно 1100 и 17. Все города России перечислены на страницах как Русской Википедии (см. «Список городов России»), так и Английской (см. “List of cities and towns in Russia”). С другой стороны, Викиданные содержат 1108 российских городов на 2024 год[26]. То есть Викиданные почти полностью покрывают российские города.

Упражнения

1. Постройте граф братских городов России.
2. Получите список городов России, находящихся за полярным кругом.
3. На какой реке в России стоит наибольшее число городов?
4. У какой страны самая большая доля городов-побратимов внутри страны относительно числа побратимов, которые связывают эту страну с другими странами?

[22] Ссылка на SPARQL-запрос: https://w.wiki/jyB.

[23] По табл. 6.2 на с. 57 видно, что urban settlement in Russia (Q2661988) и city or town (Q7930989) являются наиболее многочисленными объектами-городами в России.

[24] Число районов, городских и сельских населенных пунктов по субъектам Российской Федерации. Всероссийская перепись населения 2010 года. URL: https://bit.ly/2JPL34b.

[25] Итоги переписи населения в Крымском федеральном округе. Федеральная служба государственной статистики. 2015. URL: https://bit.ly/2Lflc6F.

[26] Ссылка на SPARQL-запрос для подсчёта числа российских городов: https://w.wiki/jyP.

8. Анализ стран: возраст, формы правления и этнохоронимы

Глава посвящена исследованию стран на основе Викиданных. С помощью SPARQL-запросов, вычисляемых на объектах «страна», получены: список современных стран, перечень всех стран, упорядоченных по дате создания, список этнохоронимов стран. Построены пузырьковая диаграмма с формами правления стран, граф соседних стран и карта соседних стран России. Выполнена оценка Викиданных по количеству исторических и современных стран, по количеству заполненных этнохоронимов у стран.

Список стран и степень полноты информации ряда стран

Анализ экземпляров объекта страна (Q6256) с помощью сервиса ProWD в 2020 году показал, что индекс Джини равен 0.091 (рис. 8.1). Это значит, что страны хорошо проработаны в Викиданных и имеют примерно одинаковое количество заполненных свойств (рис. 8.1). Легко убедиться, что это достаточно большое количество свойств.

Построим список всех стран на английском и русском языках (запрос 8.1).

Запрос 8.1. Экземпляры объекта «страна»[1]

```
#List of countries in English and Russian
SELECT ?country ?label_en ?label_ru WHERE
{
    ?country wdt:P31 wd:Q6256. # instance of country
    ?country rdfs:label ?label_en filter (lang(?label_en) = "en").
    ?country rdfs:label ?label_ru filter (lang(?label_ru) = "ru").
}
```

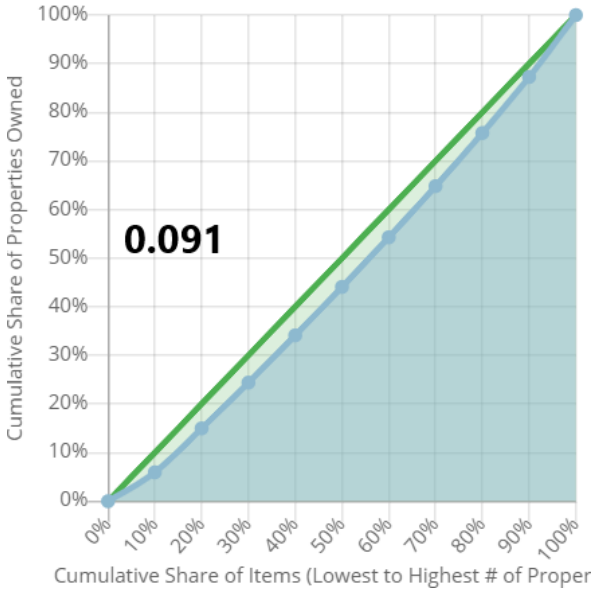


Рис. 8.1. Индекс Джини — равномерность заполнения свойств «стран», 2020 год

[1] Получено: 205 стран на 2017 год и 175 стран на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/k6L.

Возраст стран и почему Россия бывает не страной? О конструкции p:/ps:

Построим список стран, отсортированных по дате основания, то есть по первому упоминанию о ней (запрос 8.2).

Запрос 8.2. Даты основания стран[2]

```
# List of countries sorted by inception
SELECT ?country ?countryLabel ?inception
WHERE
{
  ?country wdt:P31 wd:Q6256.    # instance of country
  ?country wdt:P571 ?inception. # the first mention
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
ORDER BY (?inception)
```

В результате выполнения запроса 8.2 получен скромный список, включающий на 2020 год всего 199 стран. На примере России разберёмся, в чём здесь дело. Объект Россия (Q159) в поле `instance of` содержит не одно, а восемь значений, в том числе country (Q6256).

Решение и ответ на этот вопрос были найдены на странице Wikidata:Request a query[3]. Дело в том, что конструкция wdt позволяет находить только истинные значения. Для России предпочтительным ответом (англ. preferred value) в поле `instance of` указано «суверенное государство», а не «страна». Чтобы проверить все варианты, представленные в поле `instance of` России, нужно использовать конструкцию `p:/ps:`.

Таким образом, запрос 8.3 позволяет получить больше стран, отсортированных по дате создания. В 2020 году было получено 235 стран, в 2022 — 246 стран.

У Латвии их 119, у Таиланда 77, у Дании 5, а у России 81.
О количестве чего идет речь?

- Количество городов с населением более миллиона человек
- Количество высших учебных заведений
- Количество административных единиц
- Количество официальных языков

См. ответ на с. 167.

[2] Получено: 112 стран на 2017 год и 199 стран на 2020. Ссылка на SPARQL-запрос: https://w.wiki/suP.

[3] На странице сайта Викиданных *Request a query* одни редакторы задают вопросы, как написать тот или иной скрипт, а другие редакторы им отвечают. Пользуйтесь этим форумом. Вот ссылка с ответом на наш вопрос на этом форуме: https://w.wiki/tLm, см. раздел *List of countries*.

Запрос 8.3. Список стран, отсортированных по дате основания[4]

```
# List of countries sorted by inception date
SELECT ?country ?countryLabel (MIN(?year) AS ?min_year) WHERE
{
    ?country p:P31  [ps:P31 wd:Q6256];      # instance of a country
             p:P571 [ps:P571 ?inception].   # all inception dates
    BIND(YEAR(?inception) AS ?year)
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
GROUP BY ?country ?countryLabel
ORDER BY ?min_year
```

[4] Получено: 235 стран в 2020 году и 246 стран в 2022 году. Ссылка на SPARQL-запрос: https://w.wiki/5BCN.

Чтобы убрать из этого списка уже не существующие страны, то есть экземпляры объекта historical country (Q3024240), используем оператор `MINUS`. С помощью запроса 8.4 получим список действующих, то есть неисторических, стран с известной датой основания.

Запрос 8.4. Список стран, отсортированных по дате основания, не включающий исторические страны[5]

```
# List of countries sorted by inception date without historical countries
SELECT ?country ?countryLabel (MIN(?year) AS ?min_year) WHERE
{
  ?country p:P31 [ps:P31 wd:Q6256].               # instance of a country
  MINUS {?country p:P31 [ps:P31 wd:Q3024240]}.    # except historical countries
  ?country p:P571 [ps:P571 ?inception].           # all inception dates
  BIND(YEAR(?inception) AS ?year)
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
GROUP BY ?country ?countryLabel
ORDER BY ?min_year
```

[5] Получено: 211 стран в 2020 году и 212 стран в 2022 году. Ссылка на SPARQL-запрос: https://w.wiki/5BCP.

Например, первое упоминание о Франции (Q142) было в 463 году, о России (Q159) — в 862-м, о Республике Косово (Q1246) — в 2008-м, о Южном Судане (Q958) — в 2011 году. Наибольшее количество стран появилось в 1960 году (16 стран), в 1991-м (15 стран), в 1962-м (6 стран) и в 1821 году (6 стран).

Выведем список стран с пустым свойством «дата основания» (листинг 8.5). Такие некомплектные объекты Викиданных могли быть созданы по ошибке или могут содержать какие-то дефекты. Можно посмотреть историю правок объекта и обратиться с вопросом к редакторам этой статьи. Подробнее о том, что такое «история правок» в вики-проектах и как с ней работать см. в учебнике[6].

Запрос 8.5. Страны с незаполненной датой основания[7]

```
#List of 'instances of' "countries without a inception"
SELECT ?country ?countryLabel WHERE
{
    ?country wdt:P31 wd:Q6256.      # instance of country
    MINUS { ?country wdt:P571 [] }. # inception of country is empty
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

Полнота Викиданных по современным и историческим странам

Проанализируем полноту Викиданных на примере исторических и современных стран. По данным «Общероссийского классификатора стран мира» на Земле существует 253 страны[8]. Авторы статьи Википедии «Список государств» не смогли указать какое-то одно число. Статья Английской Википедии *List of sovereign states* содержит список из 205 стран.

Отдельного анализа заслуживают древние, уже не существующие государства, например страна Ассирия (Q41137). Объекты таких стран в Викиданных являются экземплярами не объекта country, а объекта historical country (историческая страна). С помощью запроса 8.6 построим список исторических стран, таких на 2020 год оказалось 3 тыс., что на порядок больше числа современных стран.

Запрос 8.6. Список исторических стран[9]

```
# List of historical countries
SELECT ?country ?countryLabel WHERE
{   ?country p:P31 [ps:P31 wd:Q3024240].    # instance of a historical country
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en"}
}
```

[6] *Крижановский А. А.* Работа в вики-среде на примере Русской Википедии. Ч. 1 : учеб. пособие. Петрозаводск : ПетрГУ, 2015. URL: https://commons.wikimedia.org/?curid=86907665.

[7] Получено: 100 стран на 2017 год и 7 стран на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/k6q.

[8] URL: https://classifikators.ru/oksm.

Напишите запрос, чтобы найти государства, существовавшие дольше всех. См. ответ на с. 167.

[9] Получено: 3026 стран на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/tQN.

У экземпляров объекта «страна» обычно заполнено свойство inception (P571), то есть дата основания. Однако не всегда точно можно указать дату основания страны по разным причинам: отсутствие, недостаток или противоречие письменных источников. Например, основание Древнерусского государства связывают с призванием варяжского князя Рюрика в 862 году, но точной даты нет (объект Россия (Q159)). Некоторым современным странам предшествовал ряд исторических событий, и дату какого из них считать за дату создания современной страны — это вопрос открытый. Например, датой основания Монголии (Q711) принято считать 29 декабря 1911 года, когда произошло провозглашение независимости от Китая. Хотя в истории Монголия появляется во времена деятельности Чингисхана, который кратковременно в начале XIII века объединил под своей властью большую часть Евразии.

Этнохоронимы стран на русском языке

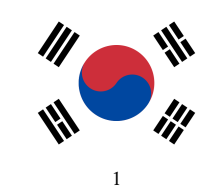
Этнохороним — это название жителей определённой местности, соотнесённое с топонимом. Например, этнохоронимами для России будут россияне, россиянин, россиянка, для Чехии — чехи, чех, чешка. Помимо географического фактора, новые лексемы, используемые для определения происхождения либо принадлежности, происходят также от этнических, политических, религиозных характеристик людей[10].

Этнохоронимы могут определяться названиями разных объектов земной поверхности: гор, островов, континентов. Также обозначение места происхождения людей может зависеть от политико-административного деления. Например, для обозначения гражданства: Таиланд — таиландцы, Канада — канадцы. Внутригосударственное деление также может породить новые наименования: Крым — крымчане. Построим список стран, у которых есть этнохоронимы на русском языке (запрос 8.7).

Запрос 8.7. Список стран с этнохоронимами на русском языке[11]

```
# List of countries with demonyms in Russian
SELECT ?country ?countryLabel WHERE
{ ?country p:P31 [ps:P31 wd:Q6256]. # instance of a country
  ?country wdt:P1549 ?demonym .     # has demonym
  FILTER((LANG(?demonym)) = "ru")
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
GROUP BY ?country ?countryLabel
```

Определите по флагам страны Азии и перечислите их в порядке возрастания плотности населения.



1

2

3

4

См. ответ на с. 168.

[10] *Журавлева Н. Ю.* Особенности перевода катойконимов (на материале испанского и русского языков) // Вестник Российского университета дружбы народов. Серия: Лингвистика. 2012. № 3. С. 41—48. ISSN 2686-8024. URL: http://journals.rudn.ru/linguistics/article/view/9530.

[11] Получено: 28 стран на 2017 год и 131 страна на 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/tec.

Список этнохоронимов стран

Построим список этнохоронимов стран на русском языке (запрос 8.8).

Запрос 8.8. Список этнохоронимов[12]

```
# List of demonyms of countries in Russian
SELECT ?country ?countryLabel ?demonym
WHERE
{
  ?country p:P31 [ps:P31 wd:Q6256]. # instance of a country
  ?country wdt:P1549 ?demonym .     # has demonym
  FILTER((LANG(?demonym)) = "ru")
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

Какие из языков — абазинский, мокшанский, эрзянский, белорусский — являются официальными в России?
См. ответ на с. 169.

[12] Получено: 83 этнохоронима на 2017 год и 296 этнохоронимов на 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/teg.

Страны с незаполненными этнохоронимами

Построим список стран, у которых нет этнохоронимов на русском языке (запрос 8.9). Этот список может быть полезен редактору для добавления этнохоронимов в Викиданные.

Благодаря конструкции MINUS в строках 5–7 запроса 8.9 в итоговый список не попали страны, имеющие этнохоронимы на русском языке.

Запрос 8.9. Страны с незаполненными этнохоронимами на русском языке[13]

```
1  # List of countries without demonyms in Russian
2  SELECT ?country ?countryLabel WHERE
3  {
4    ?country p:P31 [ps:P31 wd:Q6256].       # instance of a country
5    MINUS { ?country wdt:P1549 ?demonym.    # without demonyms
6            FILTER((LANG(?demonym)) = "ru") # in Russian
7          }
8    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
9  }
10 GROUP BY ?country ?countryLabel
```

[13] Получено: 170 стран на 2017 год и 105 стран на 2021 год.
Ссылка на SPARQL-запрос: https://w.wiki/teo.

Количество заполненных этнохоронимов у стран

У одной страны может быть от нуля (если данные не заполнены) до трёх-четырёх этнохоронимов. Например, у Турции есть три названия жителей: турок, турчанка, турки, у Эфиопии — четыре: эфиоп, эфиопка, эфиопы, эфиопки.

Получим список стран, упорядоченный по количеству заполненных в Викиданных этнохоронимов на разных языках (запрос 8.10).

Запрос 8.10. Список стран, упорядоченный по количеству заполненных этнохоронимов[14]

```
SELECT ?country ?countryLabel (COUNT(*) AS ?demonyms)
WHERE
{
  ?country p:P31 [ps:P31 wd:Q6256]; # instance of a country
           p:P1549 [ps:P1549 []].   # has demonym

  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
GROUP BY ?country ?countryLabel
ORDER BY DESC(?demonyms)
```

По данным на 2017 год, больше всего этнохоронимов у Соединённых Штатов Америки (41 этнохороним), затем идут Великобритания (40), Германия (40) и Канада (36).

В 2021 году лидером стала Германия с 64 этнохоронимами, затем идут Россия (61 этнохороним), Канада (60) и США (60). Таким образом, с 2017 по 2021 год добавилось примерно по 20 этнохоронимов на одну страну.

[14] Получено: 199 стран на 2017 год и 215 стран на 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/5BX5.

Формы правления стран

Построим пузырьковую диаграмму форм правления стран (запрос 8.11), где размер пузырька будет соответствовать числу стран с той или иной формой правления.

Запрос 8.11. Число стран с разными формами правления[15]

```
# Forms of government ordered by number of countries
#defaultView:BubbleChart
SELECT ?bfog ?form (COUNT(*) AS ?countries) WHERE
{ ?country p:P31 [ps:P31 wd:Q6256]; # instance of a country
           p:P122 [ps:P122 ?bfog].  # basic form of government
  OPTIONAL
  { ?bfog rdfs:label ?form
    FILTER (LANG(?form) = "ru")
  }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru"}
}
GROUP BY ?bfog ?form
ORDER BY DESC(?countries) ASC(?form)
```

Переменная `bfog` (сокращение от basic form of government) содержит форму правления, например «республику». Последняя строка в запросе 8.11 содержит команды упорядочения сначала по убыванию (DESC), затем по возрастанию (ASC). Таким образом, формы правления сначала сортируются по числу стран `?countries`. Затем, если стран оказалось поровну, то формы правления сортируются лексикографически[16].

Результаты запроса 8.11 показали, что если на 2017 год основными формами правления были республика (20 стран), конституционная монархия (18 стран), федеративная республика (18), парламентская республика (17), президентская республика (12 стран), то на 2020 год показатели таковы: республика (41 страна), конституционная монархия (32), федеративная республика (19), парламентская республика (22), президентская республика (14 стран). Отметим, что с 2017 по 2020 год форма правления *республика* стала более популярной (20 и 41 страна).

Результатом выполнения запроса 8.11 является пузырьковая диаграмма с наиболее распространенными формами правления (рис. 8.2).

Какая строка в запросе 8.11 является лишней? То есть её можно удалить и результат не изменится. Речь идёт не о первой строке с комментарием.

[15] Получено: 30 форм правления на 2017 год и 41 форма правления на 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/5BVh.

[16] Лексикографический (словарный) порядок — это способ упорядочивания и сортировки слов, который обычно используется в словарях, энциклопедиях и алфавитных указателях, например: А < АА < ААА < ААБ < ААВ < АБ < Б < . . . < ЯЯЯ.

Рис. 8.2. Соотношение числа разных форм правления по странам мира на 2020 год

Граф соседних стран

У стран существует такое свойство, как общая граница. В Викиданных это свойство называется shares border with (P47). Используя его, построим граф соседних стран (запрос 8.12).

Запрос 8.12. Соседние страны[17]

```
# Graph of countries which share border
#defaultView:Graph
SELECT DISTINCT ?country ?countryLabel ?border ?sharesBorderWith WHERE
{
    ?country p:P31 [ps:P31 wd:Q6256]. # instance of a country
    OPTIONAL {?country wdt:P47 ?sharesBorderWith}
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
```

В результате выполнения запроса мы получаем граф с 787 ребрами на 2017 год (рис. 8.3) и 912 ребрами на 2021 год (рис. 8.4), где ребро указывает на общую границу двух стран. Граф представляет из себя несколько связных компонент, так как есть островные страны, у которых нет соседей (например, Маврикий и Мальдивы). Также стоит упомянуть, что теперь свойство *shares border with* включает общую не только сухопутную, но и морскую границу. Поэтому теперь этот граф будет представлять одну связную компоненту.

Обратите внимание, что полученный граф (рис. 8.3) включает также исторические страны, например СССР. Постройте аналогичный граф, но только с современными странами. Для этого посмотрите на запрос 8.4, где показано, как «выключить» исторические страны и оставить только действующие.

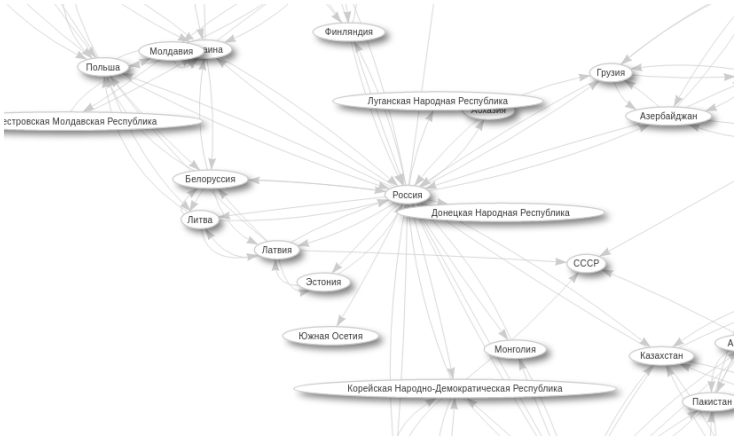


Рис. 8.3. Фрагмент графа соседних стран, в центре Россия, 2017 год

[17] Получено: 787 соседств на 2017 год и 912 соседств на 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/5Bsy.

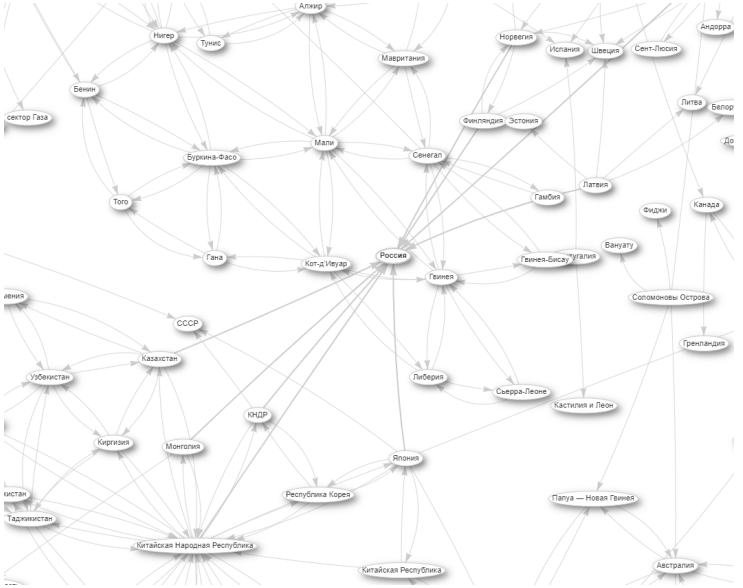


Рис. 8.4. Фрагмент графа соседних стран, в центре Россия, 2020 год

Карта соседних стран России

Построим карту соседних стран России с помощью запроса 8.13. Строка 6 запроса 8.13 фильтрует и оставляет в переменной `?border_country` только страны. Это позволило исключить, например, район Грузии (Рача-Лечхуми и Квемо-Сванети) и остров Японии (Хоккайдо), указанные в списке пограничных объектов России.

В результате выполнения запроса 8.13 мы получили карту соседних стран России (рис. 8.5), включающую 17 стран, а именно: Япония (Q17), Норвегия (Q20), США (Q30), Финляндия (Q33), Швеция (Q34), Польша (Q36), Литва (Q37), Китайская Народная Республика (Q148), Белоруссия (Q184), Эстония (Q191), Латвия (Q211), Украина (Q212), Азербайджан (Q227), Грузия (Q230), Казахстан (Q232), КНДР (Q423) и Монголия (Q711).

Запрос 8.13. Соседние страны России[18]

```
1  # Map of neighboring countries of Russia
2  #defaultView:Map
3  SELECT ?border_country ?border_countryLabel ?coords ?layer WHERE
4  {                                    # border_country
5    ?border_country p:P47 [ps:P47 wd:Q159]. # has border with Russia
6    ?border_country p:P31 [ps:P31 wd:Q6256].# is a country
7    OPTIONAL {?border_country wdt:P3896 ?coords.} # geoshape
8    BIND (?coords AS ?layer)
9    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru". }
10 }
```

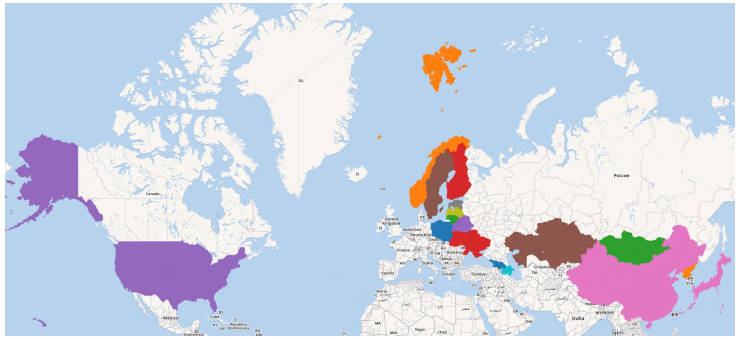


Рис. 8.5. Карта соседних стран России, включающая 17 стран, 2021 год

[18] Получено: 17 стран на 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/tg3.

Упражнения

1. Постройте список флагов и девизов стран. Девизы есть не у всех стран.
2. Отметьте на карте столицы современных стран.
3. В каждой части света вычислите первые пять стран с наибольшей плотностью населения.
4. Постройте столбчатую диаграмму, демонстрирующую распределение количества стран по формам правления.
5. Выведите список стран, упорядоченных по числу соседей. У каких стран больше всего и меньше всего соседей? Какое среднее число соседей? Есть ли корреляция между этим показателем и каким-либо другим параметром стран?

9. Регионы России

Исследуем в Викиданных регионы России, которые включают в себя множество земель разного типа: области, республики, края и др. Именно эти разнотипные регионы и были исследованы. Был построен граф субъектов России, граничащих с зарубежными странами (граф соседей), а также нарисована карта, на которой отмечена численность населения отдельных регионов. Оценка степени заполненности свойства Викиданных «граничит с» (shares border with) показала, что у каждого субъекта России эти данные заполнены полностью. Были построены две карты, на которых цветом обозначены граничные регионы и пограничные страны. Читатель познакомится с компьютерной обработкой Викиданных и визуализацией информации о регионах России.

Экземпляры объекта «Области России»

Для построения списка всех областей России нам потребуются объект «области России» (Q835714) и свойство «экземпляры» (P31) (запрос 9.1).

Запрос 9.1. Список всех областей России[1]

```
# List of regions of Russia
SELECT ?region ?regionLabel WHERE
{
  ?region wdt:P31 wd:Q835714. # is "oblast of Russia"
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

О флаге какого региона идёт речь: «Флаг этого субъекта представляет собой прямоугольное полотнище с отношением ширины к длине 2:3, красного цвета с двусторонним изображением в верхнем ближнем к древку углу основного элемента герба этого субъекта — развёрнутого к древку Святого Георгия Победоносца.

См. ответ на с. 170.

[1] Получено: 48 записей в 2017 году и 46 записей в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4D2V.

По данным сервиса ProWD, в Викиданных больше всего свойств в России у Ленинградской (Q2191) и Калининградской (Q1749) областей, по 43 свойства. Областями России с наименьшим числом свойств оказались: Орловская и Курская области (по 31 свойству) и Новосибирская область (32 свойства).

Субъекты Российской Федерации

Построим список всех субъектов России. Для этого выберем следующие объекты в Викиданных: республики, края, области, города федерального значения, автономные области и автономные округа (запрос 9.2). Далее будем использовать следующие объекты:

- «области России» (Q835714);
- «республики России» (Q41162);
- «города федерального значения России» (Q183342);
- «края России» (Q831740);
- «автономные области России» (Q309166);
- «автономные округа России» (Q184122);
- «бывшая административно-территориальная единица» (Q19953632).

Используемое свойство — «экземпляры» (P31).

Запрос 9.2. Список всех субъектов России[2]

```
# List of 'instances of' "subjects of Russia"
SELECT ?subject ?subjectLabel ?typeLabel WHERE
{
  VALUES ?type {wd:Q835714   # oblast of Russia
                wd:Q41162    # republic of Russia
                wd:Q183342   # federal city of Russia
                wd:Q831740   # krai of Russia
                wd:Q309166   # autonomus oblast of Russia
                wd:Q184122}  # autonomus okrug of Russia
  ?subject wdt:P31 ?type.  # selecting the type of object
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

[2] Получено: 85 записей в 2017 году и 86 записей в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4D2R. В 2021 году в список субъектов добавился город федерального значения Байконур на правах аренды комплекса «Байконур».

Для построения запроса 9.2 и проверки результатов учтём следующую информацию:

- По данным Конституции Российской Федерации, на 2021 год Россия состояла из 85 субъектов — республик, краёв, областей, городов федерального значения, автономной области, автономных округов.
- При построении списка объектов не нужно в него включать такие субъекты, которые на текущий момент не входят в состав РФ (например, Читинская область (Q182902)). Такие объекты легко отсеить, поскольку они не являются экземплярами объектов «области России» (Q835714), «республики России» (Q41162), «города федерального значения России» (Q183342), «края России» (Q831740), «автономные области России» (Q309166), «автономные округа России» (Q184122), а относятся к объекту «бывшая административно-территориальная единица» (Q19953632).
- На 2021 год в статье Русской Википедии «Субъекты Российской Федерации» и в статье Английской Википедии Federal subjects of Russia перечислено по 85 субъектов РФ.

Соседние субъекты

Построим граф соседних субъектов России, используя свойство shares border with (P47), с помощью запроса 9.3. Получено: 467 записей в 2017 году и 482 записи в 2021 году. Ссылку на SPARQL-запрос не приводим из-за её чрезмерной длины. Вот ссылка на исходный запрос — https://w.wiki/4DKD, от которого мы отталкивались для построения этого огромного запроса. Подумайте, как можно сократить или свернуть этот огромный запрос 9.3, сделать его компактным и элегантным?

Запрос 9.3. Граф соседних субъектов России[3]

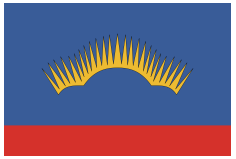
```
# Graph of "subjects of Russia"which "shares border with"
#defaultView:Graph
SELECT * WHERE
{
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru"}
  { # no borders with the objects of Russia
    SELECT ?subject ?subjectLabel ?rgb ?subjects ?subjectsLabel WHERE
    {
      # Oblast, Republic, Federal city, Krai, Autonomus oblast, Autonomus okrug of Russia
      VALUES ?type {wd:Q835714 wd:Q41162 wd:Q183342 wd:Q831740 wd:Q309166 wd:Q184122}
      ?subject wdt:P31 ?type.  # instance of
    }
  }
  UNION  # Autonomus okrug of Russia
```

Назовите регион России, расположенный на северо-западе страны и образованный в 1920 году. Он граничит с Вологодской и Архангельской областями. Также граничит с Финляндией на западе. Выберите среди следующих флагов флаг этого региона.

Ленинградская область

Московская область

Республика Карелия



Мурманская область

См. ответ на с. 170.

[3] Получено: 467 записей в 2017 году и 482 записи в 2021 году. Ссылку на SPARQL-запрос не приводим из-за её чрезмерной длины. Вот ссылка на исходный запрос (`https://w.wiki/4DKD`), от которого мы отталкивались для построения этого мегазапроса. Подумайте, как можно сократить или свернуть этот огромный запрос, сделать его компактным и элегантным?

```
15    {
16      SELECT ?subject ?subjectLabel ?rgb ?s ?sLabel WHERE
17      {
18        VALUES ?type {wd:Q835714 wd:Q41162 wd:Q183342 wd:Q831740 wd:Q309166 wd:Q184122}
19        ?okrug wdt:P31 wd:Q184122; # is instance of Autonomus okrug of Russia
20               wdt:P47 ?s.         # share border with ?s
21                       ?s wdt:P31 ?type.
22        BIND(IF(?okrug != '',"9932CC", IF(?rgb != '',?rgb,"FFFFFF")) AS ?rgb).
23        BIND(IF(?okrug != '',?okrug, ?s) AS ?subject).
24      }
25    }
26    UNION  # Autonomus oblast of Russia
27    {
28      SELECT ?subject ?subjectLabel ?rgb ?s ?sLabel WHERE
29      {
30        VALUES ?type {wd:Q835714 wd:Q41162 wd:Q183342 wd:Q831740 wd:Q309166 wd:Q184122}
31        ?auto_oblast wdt:P31 wd:Q309166; # instance of Autonomus oblast of Russia
32                wdt:P47 ?s.
33                        ?s wdt:P31 ?type.
34
35        BIND(IF(?auto_oblast != '',"ced685",IF(?rgb != '',?rgb,"FFFFFF")) AS ?rgb).
36        BIND(IF(?auto_oblast != '',?auto_oblast, ?s) AS ?subject).
37      }
38    }
39    UNION  # Krai of Russia
40    {
41      SELECT ?subject ?subjectLabel ?rgb ?s ?sLabel WHERE
42      {
43        VALUES ?type {wd:Q835714 wd:Q41162 wd:Q183342 wd:Q831740 wd:Q309166 wd:Q184122}
44        ?krai wdt:P31 wd:Q831740; # instance of Krai of Russia
45              wdt:P47 ?s.
46                      ?s wdt:P31 ?type.
47        BIND(IF(?krai != '',"7495db",IF(?rgb != '',?rgb,"FFFFFF")) AS ?rgb).
48        BIND(IF(?krai != '',?krai, ?s) AS ?subject).
49      }
50    }
51    UNION  # Federal city of Russia
52    {
53      SELECT ?subject ?subjectLabel ?rgb ?s ?sLabel WHERE
```

```
54      {
55        VALUES ?type {wd:Q835714 wd:Q41162 wd:Q183342 wd:Q831740 wd:Q309166 wd:Q184122}
56        ?fed_city wdt:P31 wd:Q183342; # instance of Federal city of Russia
57              wdt:P47 ?s.
58                      ?s wdt:P31 ?type.
59        BIND(IF(?fed_city != '','e8a2e8',IF(?rgb != '',?rgb,"FFFFFF")) AS ?rgb).
60        BIND(IF(?fed_city != '',?fed_city, ?s) AS ?subject).
61      }
62    }
63    UNION  # Republic of Russia
64    {
65      SELECT ?subject ?subjectLabel ?rgb ?s ?sLabel WHERE
66      {
67        VALUES ?type {wd:Q835714 wd:Q41162 wd:Q183342 wd:Q831740 wd:Q309166 wd:Q184122}
68        ?republic wdt:P31 wd:Q41162; # instance of Republic of Russia
69                  wdt:P47 ?s.
70                          ?s wdt:P31 ?type.
71        BIND(IF(?republic != '',"7FFF00",IF(?rgb != '',?rgb,"FFFFFF")) AS ?rgb).
72        BIND(IF(?republic != '',?republic, ?s) AS ?subject).
73      }
74    }
75    UNION  # Oblast of Russia
76    {
77      SELECT ?subject ?subjectLabel ?rgb ?s ?sLabel WHERE
78      {
79        VALUES ?type {wd:Q835714 wd:Q41162 wd:Q183342 wd:Q831740 wd:Q309166 wd:Q184122}
80        ?oblast wdt:P31 wd:Q835714; # instance of Oblasts of Russia
81                wdt:P47 ?s.
82                        ?s wdt:P31 ?type.
83        BIND(IF(?oblast != '',"e87b7b",IF(?rgb != '',?rgb,"FFFFFF")) AS ?rgb).
84        BIND(IF(?oblast != '',?oblast, ?s) AS ?subject).
85      }
86    }
87  }
```

Результатом работы запроса 9.3 является граф, отображающий соседние субъекты. Причём разные типы субъектов имеют вершины разного цвета, например республики — зелёные, а края — голубые. Часть графа представлена на рис. 9.1.

С помощью команды `BIND` (строки 83–84 запроса 9.3) записываем значения в переменные `?rgb` и `?subject` при условии, что эти переменные были пустыми.

Переменная `?oblast` ищется в строках 80–82. Подбирается такой объект `?oblast`, который одновременно является экземпляром объекта «области России» (Q835714) и «граничит с» (P47) объектом `?s`. При этом переменная `?s` является экземпляром какого-либо типа субъектов РФ (то есть `?s` — это экземпляр переменной `?type`, заданной в виде списка в строке 79).

Если найдена переменная `?oblast`, соответствующая условиям строк 80–82, тогда, во-первых, присваиваем `?subject := ?oblast` (строка 84). Во-вторых, если к тому же переменная `?rgb` ещё пуста, то присваиваем розовый цвет переменной `?rgb := "e87b7b"` (строка 83). Этот цвет на рис. 9.1 соответствует трём областям: Амурской, Магаданской и Сахалинской.

Если переменная `?oblast` уже не пустая, а цвет ещё не задан, то пишем `?rgb := "FFFFFF"` (строка 83), выбирая светло-серый цвет. Если переменная `?oblast` найдена, а цвет `?rgb` уже задан, то мы его оставляем, то есть перезаписываем `?rgb := ?rgb`.

Количество полученных записей формируется путём сложения количества соседних территорий для всех субъектов России.

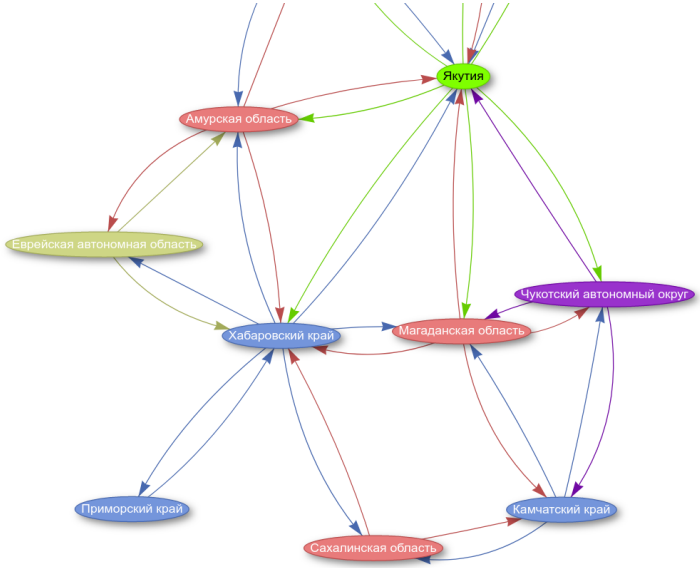


Рис. 9.1. Фрагмент графа субъектов России на 2021 год, включающий регионы Сибири и Дальнего Востока, при этом республикам соответствуют вершины зелёного цвета (Якутия); автономным округам — фиолетового (Чукотский автономный округ); краям — голубого (Хабаровский край); областям — розового (Амурская область); автономным областям — вершины салатового цвета (Еврейская автономная область)

Построим карту, на которой цветом обозначены субъекты России, граничащие с зарубежными странами. Более тёмным цветом обозначены субъекты, у которых большее количество заграничных соседей, более светлым — с меньшим количеством (запрос 9.4).

Запрос 9.4. Карта субъектов России, граничащих с зарубежными странами

```
# Map of countries around Russia with the number of neighboring regions of Russia
#defaultView:Map{"hide":["?shape "?rgb"], "layer": "?regionLabel"}
SELECT ?region ?regionLabel ?count ?shape ?rgb
{
  {
    SELECT ?region (COUNT(DISTINCT ?country) AS ?count) WHERE {
      VALUES ?type {
        wd:Q835714  # oblasts of Russia - 9 neighbours
        wd:Q41162   # republic of Russia - 4
        wd:Q183342  # federal city of Russia - 0 neighbours
        wd:Q831740  # krai of Russia - 4
        wd:Q309166  # autonomous oblast of Russia - 1
        wd:Q184122   # autonomous okrug of Russia - 1
      }
      ?region wdt:P31 ?type.

      # Russian region share border with some territory
      # of foreign country
      ?region wdt:P47 [ wdt:P17 ?country].
      FILTER (?country != wd:Q159) # foreign country is not Russia
    }
    GROUP BY ?region HAVING ((COUNT(?country)) > 0)
  }
  ?region wdt:P3896 ?shape.
  BIND(IF(?count = 3 , "6c2eab",
            IF(?count = 2 , "9b77bf",
                IF(?count = 1 , "c6b2db", "f5cbce"))) AS ?rgb)
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

На рис. 9.2 представлен результат работы запроса 9.4 (https://w.wiki/4e82). По этому запросу в 2021 году получено 37 субъектов России, граничащих с зарубежными странами.

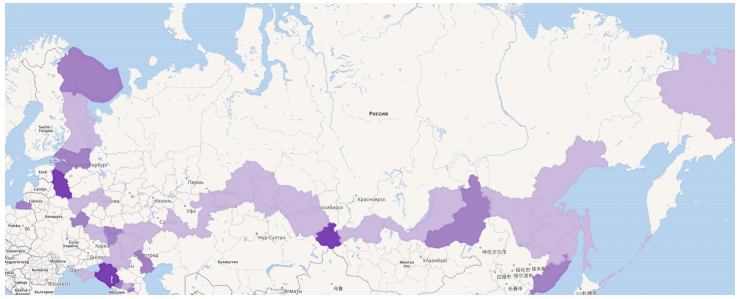


Рис. 9.2. Карта пограничных субъектов России, 2021 год

Построим карту (рис. 9.3), на которой цветом обозначены зарубежные страны, граничащие с субъектами России. Более тёмным цветом обозначены страны, граничащие с большим количеством субъектов России, более светлым — с меньшим количеством. Карта построена с помощью запроса 9.5.

Запрос 9.5. Карта стран, граничащих с субъектами России[4]

```
# Map of countries around Russia
# with the number of neighboring regions of Russia
#defaultView:Map{"hide":["?shape "?rgb"], "layer": "?countryLabel"}
SELECT ?country ?countryLabel ?count ?shape ?rgb WHERE {
{
    SELECT ?country (COUNT(DISTINCT ?region) AS ?count) WHERE {
      VALUES ?type {wd:Q835714 wd:Q41162 wd:Q183342
                    wd:Q831740 wd:Q309166 wd:Q184122}
      ?region wdt:P31 ?type;
              wdt:P47 [wdt:P17 ?country].
      FILTER (?country != wd:Q159) # foreign country is not Russia
    }
    GROUP BY ?country HAVING ((COUNT(?region)) > 0)
  }
  ?country wdt:P3896 ?shape.
  BIND(IF(?count > 9 , "4B0082",
        IF(?count > 5 , "800080",
         IF(?count > 2 , "8B008B",
          IF(?count > 1 , "9400D3",
           IF(?count > 0 , "DA70D6", "f5cbce"))))) AS ?rgb)
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru".}
}
```

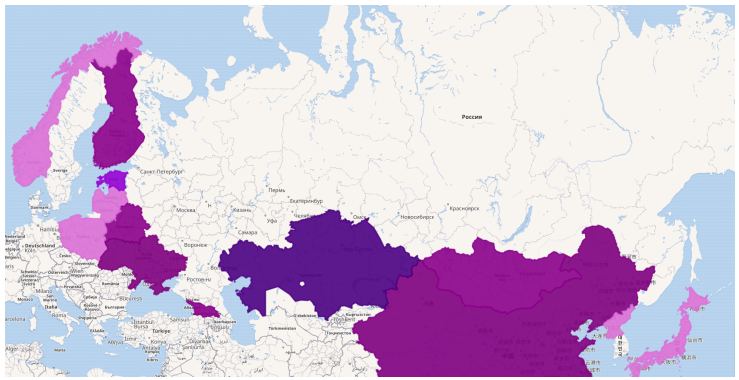


Рис. 9.3. Карта стран, граничащих с Россией, с учётом числа примыкающих пограничных субъектов РФ, 2021 год

[4] Получено: 15 записей в 2021 году. SPARQL-запрос: https://w.wiki/4e8T.

Полнота Викиданных по числу субъектов, имеющих границу с кем-либо

С помощью запроса 9.6 построим список субъектов России с пустым свойством shares border with (P47) (граничит с), то есть попробуем найти такие субъекты, которые ни с кем не граничат.

С помощью команды `FILTER` (строка 15 запроса 9.6) исключаем объекты, которые не находятся на территории России. Затем с помощью команды `MINUS` (строка 18) удаляем из рассмотрения объекты, у которых свойство «граничит с» (P47) заполнено.

Получено 0 записей, то есть на Викиданных для всех субъектов России свойство «граничит с» (P47) заполнено.

Запрос 9.6. Список субъектов РФ с пустым свойством *shares border with* (P47)[5]

```
1  # List of 'subjects of Russia' without 'shares border with'.
2  SELECT
3      ?subject ?subjectLabel
4      ?sharesBorderWith ?sharesBorderWithLabel
5  WHERE
6  {
7    VALUES ?type {wd:Q835714   # Oblast of Russia
8                  wd:Q41162    # Republic of Russia
9                  wd:Q183342   # Federal city of Russia
10                 wd:Q831740   # Krai of Russia
11                 wd:Q309166   # Autonomus oblast of Russia
12                 wd:Q184122}  # Autonomus okrug of Russia
13   ?subject wdt:P31 ?type.
14
15   FILTER EXISTS {?subject wdt:P17 wd:Q159;
16                          wdt:P31 ?type}
17
18   MINUS { ?subject  wdt:P47 [] }. # shares border with
19   SERVICE wikibase:label { bd:serviceParam wikibase:language "ru"}
20 }
```

Какие из перечисленных субъектов входят в состав Российской Федерации, а какие не входят?

- Республика Адыгея
- Камчатский край
- Читинская область
- Чукотский автономный округ

См. ответ на с. 171.

[5] Получено: 0 записей в 2017 и 2021 годах. SPARQL-запрос: https://w.wiki/5CHV.

Численность населения по субъектам Российской Федерации

Обозначим на карте субъекты Российской Федерации, разделив их на шесть групп по количеству населения. Субъектам, принадлежащим одной группе, будет соответствовать на карте один цвет.

Для запроса 9.7 нам потребуются свойства «координаты» (P625) и «численность населения» (P1082). Результат работы запроса 9.7 представлен на рис. 9.4.

Запрос 9.7. Карта населения России[6]

```
# Map of population of subjects of Russia
#defaultView:Map
SELECT DISTINCT ?subject ?subjectLabel ?population ?coord ?layer
{
  {
    { ?subject wdt:P31 wd:Q835714 } UNION  # Oblast of Russia
    { ?subject wdt:P31 wd:Q41162 } UNION  # Republic of Russia
    { ?subject wdt:P31 wd:Q183342 } UNION  # Federal city of Russia
    { ?subject wdt:P31 wd:Q831740 } UNION  # Krai of Russia
    { ?subject wdt:P31 wd:Q309166 } UNION # Autonomus oblast
                                                    of Russia
    { ?subject wdt:P31 wd:Q184122 } # Autonomus okrug of Russia
  }
  ?subject wdt:P625 ?coord; wdt:P1082 ?population.

  BIND(
    IF(?population < 500000, "< 500000",
    IF(?population < 1000000, "500000 - 1000000",
    IF(?population < 3000000, "1000000 - 3000000",
    IF(?population < 8000000, "3000000 - 8000000",
    IF(?population < 10000000, "8000000 - 10000000",
    "> 10000000")))))
    AS ?layer).

  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru"}
}
ORDER BY ?population
```

[6] Получено: 85 записей в 2017 году и 86 записей в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4bHe.

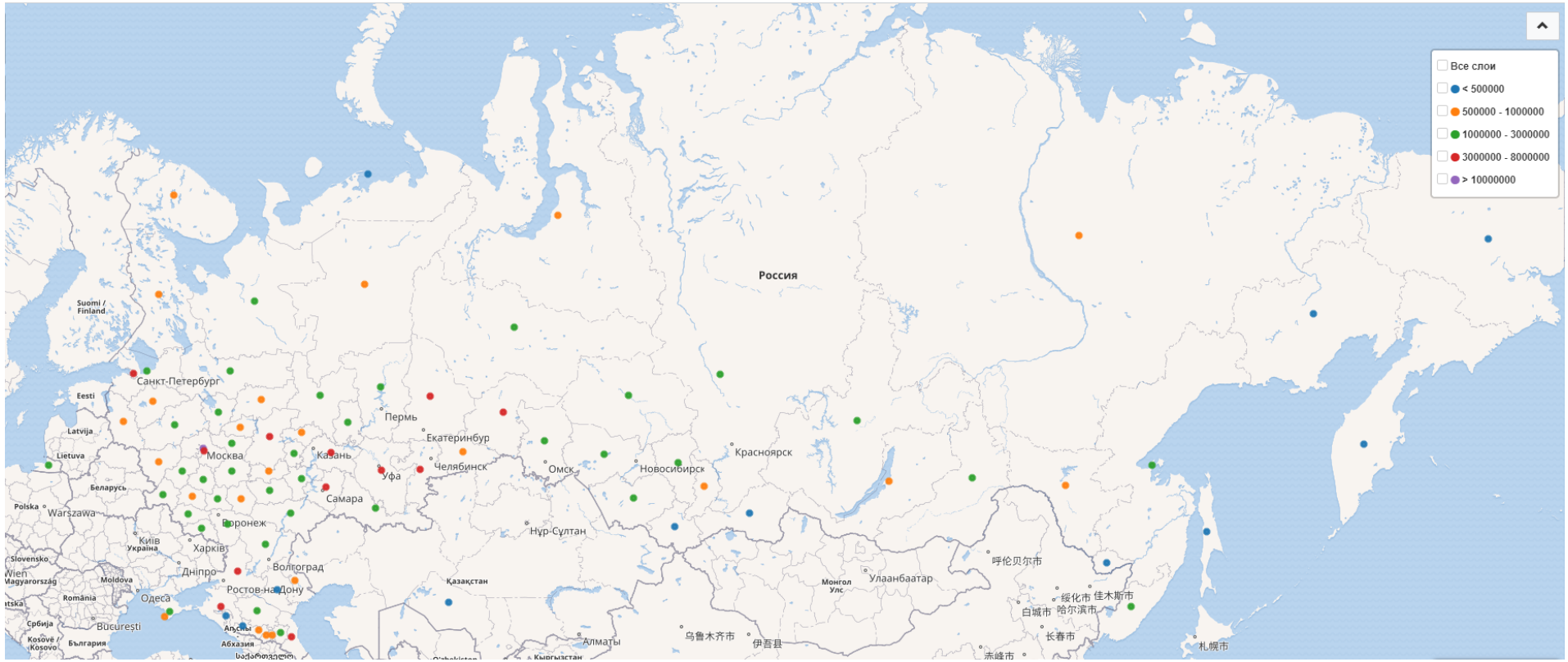


Рис. 9.4. Численность населения в субъектах России, 2021 год

10. На каких языках программирования пишутся операционные системы

В главе исследуется объект Викиданных «операционная система» (operating system) и его свойства. Построен список операционных систем (ОС), включающий информацию о их «предках»[1], времени создания, языках программирования, на которых написаны ОС. Также построена гистограмма, показывающая количество программ, написанных на том или ином языке программирования с указанием операционной системы. У большого количества ОС в Викиданных не указан язык программирования, на котором она разрабатывалась, а именно: на 2020 год это свойство указано лишь у 29 % из общего количества систем. Викиданные стремительно пополняются новой информацией, что особенно отрадно, поскольку они играют большую роль в документировании программного обеспечения.

[1] «Предки» — здесь операционные системы, послужившие основой для создания новых ОС.

Список операционных систем

С помощью запроса 10.1 получим список всех операционных систем.

Запрос 10.1. Список всех операционных систем[2]

```
SELECT ?os ?osLabel WHERE
{
    ?os wdt:P31 wd:Q9135. # instance of operating system
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en"}
}
```

[2] Получено: 510 операционных систем в 2017 году и 1086 систем в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/n89.

Наиболее полными и проработанными операционными системами в Викиданных на 2021 год были Microsoft Windows (Q1406), Windows 8 (Q5046), GNU (Q44571), Windows 10 (Q18168774), имеющие по 24 заполненных свойства. Почти пустыми и малоинформативными операционными системами

на 2021 год были: MagiC (Q1884068), KallistiOS (Q1722492), Novell DOS (Q3345389) и ряд других. У этих систем было заполнено лишь одно свойство. По данным ProWD, у единственной в Викиданных отечественной ОС Miraculix (Q4044344) заполнено 10 свойств.

Предшественники операционных систем и даты выпуска

Запрос 10.2 позволяет получить список базовых (P144) операционных систем. Этот запрос показывает соответствие между операционной системой (Q9135) и её «предком», то есть предшествующей операционной системой, на которой она основана.

Запрос 10.2. Список базовых операционных систем[3]

```
SELECT ?os ?osLabel ?base ?baseLabel WHERE
{
    ?os wdt:P31 wd:Q9135;   # is instance of operating system
        wdt:P144 ?base.     # is based on ?base
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
}
```

Список операционных систем с указанием даты их создания получим с помощью запроса 10.3.

Запрос 10.3. Список операционных систем с датой их создания[4]

```
#defaultView:Timeline{"hide": "?when"}
SELECT DISTINCT ?os ?osLabel ?when (YEAR(?when) as ?date) ?pic WHERE
{
    ?os wdt:P31 wd:Q9135;           # instance of operating system
        wdt:P571 ?when.             # inception date
    OPTIONAL { ?os wdt:P18 ?pic }
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
}
```

На основе какой из операционных систем — Debian, Android, Ubuntu, ядро Linux — было создано больше всего других операционных систем?
См. ответ на с. 171.

[3] Получено: 47 базовых операционных систем в 2017 году и 118 систем в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/5CKF.

Какую из операционных систем — Newton OS, Ubuntu Touch или JavaOS — разработала компания Apple?
См. ответ на с. 172.

[4] Получено: 30 дат создания ОС в 2017 году и 238 дат в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/5CKS.

Команда `#defaultView:Timeline` в строке 1 запроса 10.3 похожа на комментарий, а в действительности является инструкцией для сервиса WDQS[5]. Эта инструкция позволяет получить диаграмму на рис. 10.1 в виде временной шкалы.

В этой строке 1 запроса 10.3 параметры `{"hide": "?when"}` команды `#defaultView:Timeline` позволяют скрыть вывод переменной `?when`, оставив от даты только год (функция `YEAR()`) и название ОС (переменная `?osLabel`). Благодаря этому мы видим в центре рис. 10.1 под изображением смартфона не “1 January 1993”, а только “1993”.

Таким образом, результатом запроса 10.3 является временная шкала (англ. *timeline*) создания (точнее, выпуска[6]) операционных систем (рис. 10.1). Кроме временной шкалы, есть много других способов представления результатов запросов в графической форме, описанных в документации[7]. Количество полученных результатов запроса 10.3 составляет только 238 операционных систем (то есть 22 % от всего количества ОС) на 2020 год. Связано это с тем, что у 78 % ОС поле «дата создания» не заполнено.

[5] См. описание сервиса WDQS (Wikidata Query Service) на с. 16.

[6] Создание операционной системы — это длительный процесс, часто занимающий годы. Поэтому здесь указана дата первой версии системы.

[7] См. справочную страницу на сайте Викиданных *Wikidata:SPARQL query service/Wikidata Query Help/Result Views* о вариантах представления результатов SPARQL-запросов по ссылке: https://www.wikidata.org/?curid=29285671.

Количество операционных систем, написанных на языках программирования

В листинге 10.4 представлен SPARQL-запрос для получения списка языков программирования с выводом количества написанных на них ОС.

Запрос 10.4. Список языков программирования и количество написанных на них операционных систем[8]

```
# Languages and number of OS written in these languages
SELECT ?lang ?langLabel (COUNT(*) AS ?countOS) WHERE
{
  ?os wdt:P31 wd:Q9135. # is instance of operating system
  ?os wdt:P277 ?lang. # is written in programming language ?lang
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
}
GROUP BY ?lang ?langLabel
ORDER BY DESC(?countOS) ASC(?langLabel)
```

[8] Получено: 24 языка программирования с количеством написанных на них операционных систем в 2017 году и 36 языков в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/uat.

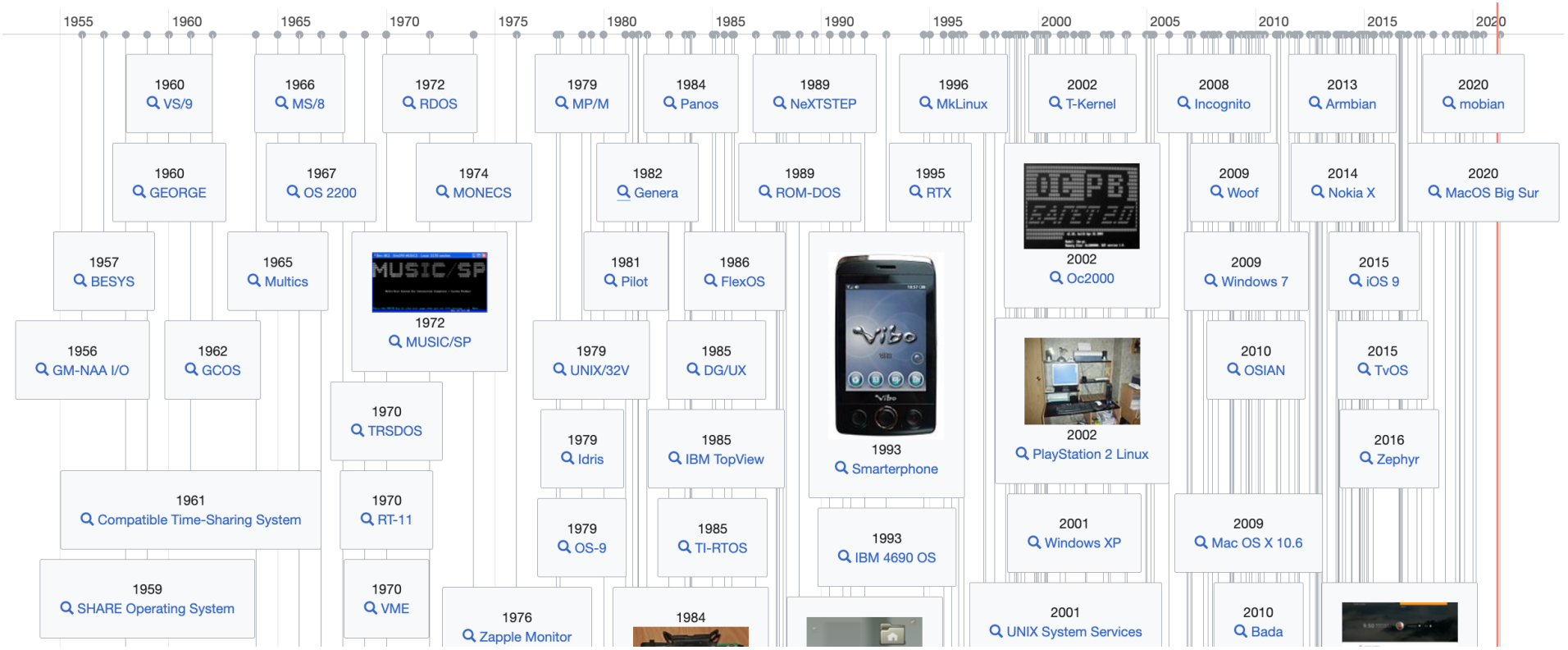


Рис. 10.1. Часть временной шкалы с датами выпуска операционных систем с 1955 по 2020 год

Команда `ORDER BY DESC(?countOS) ASC(?langLabel)` в запросе 10.4 используется для двойной сортировки значений в таблице. Все языки программирования в этой таблице сначала сортируются по убыванию количества операционных систем, которые написаны на них. Затем, если для каких-либо языков это число совпадает, сортировка идёт по именам языков.

Результат SPARQL-запроса 10.4 показывает, что преимущественно ОС пишут на языках Си (известно 46 операционных систем, написанных на языке Си) и ассемблер (42 ОС). На третьем месте — язык C++ (16 ОС).

С помощью запроса 10.5 можно получить диаграмму языков программирования и написанных на них операционных систем в виде дерева (рис. 10.2). В этом дереве каждая строчка — это язык программирования. По щелчку мыши можно развернуть язык и увидеть список операционных систем, написанных с использованием этого языка. Если язык или система имеют рисунок или логотип в Викиданных, то они используются в качестве иконок.

Запрос 10.5. Список языков программирования и написанных на них операционных систем[9]

```
# Languages and operating systems written in these languages
#defaultView:Tree
SELECT ?lang ?image ?logoImage ?langLabel
       ?os ?osImage ?osLogoImage ?osLabel
WHERE
{
  ?os wdt:P31 wd:Q9135. # is instance of operating system
  ?os wdt:P277 ?lang. # is written in programming language ?lang
  OPTIONAL { ?lang wdt:P18 ?image. }
  OPTIONAL { ?lang wdt:P154 ?logoImage. }
  OPTIONAL { ?os wdt:P18 ?osImage. }
  OPTIONAL { ?os wdt:P154 ?osLogoImage. }
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
}
```

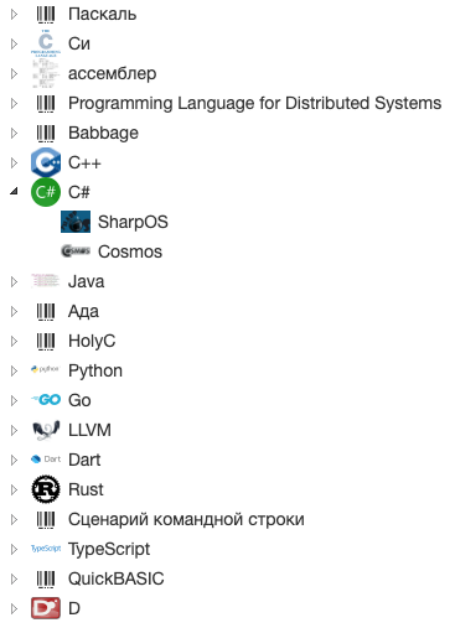


Рис. 10.2. Дерево языков программирования и написанных на них операционных систем


Сложное задание для самостоятельной работы. Измените запрос 10.5 (https://w.wiki/ucB) так, чтобы рядом с языком программирования было дано число операционных систем, написанных на этом языке.


Простое задание. «Переверните» запрос 10.5, то есть строчки верхнего уровня пусть содержат не языки программирования, а операционные системы. При «разворачивании» мы должны видеть список языков, на которых написана система.

[9] Получено: 146 языков программирования с написанными на них операционными системами в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/ucB.

Полнота Викиданных по числу операционных систем

По данным сайта www.operating-system.org, существует 613 операционных систем. Викиданные на 2017 год содержали информацию лишь о 510 операционных системах. Результаты запросов 10.3 и 10.4 показывают, что многие объекты ОС плохо заполнены, а то и вовсе практически пусты (например, у систем Novell DOS (Q3345389) и MagiC (Q1884068) заполнено всего одно свойство).

В 2020 году Викиданные содержали информацию о 1086 операционных системах (запрос 10.1), что свидетельствует о значительных изменениях, а именно: за три года (с 2017 года) количество операционных систем более чем удвоилось: с 510 до 1086. Однако большое количество объектов по-прежнему плохо заполнены, например по результатам запроса 10.3 информация о дате выпуска заполнена всего у 238 ОС. Из этого можно сделать вывод, что хотя Викиданные неполны, но они стремительно пополняются.

Языки программирования, используемые для написания операционных систем

Если взглянуть на количество операционных систем, для которых указано свойство «язык программирования», то можно увидеть, что это свойство заполнено лишь у 116 из 1086 объектов. По данным на 2020 год (рис. 10.3), больше всего операционных систем, а именно 44, написано на языке программирования Си.

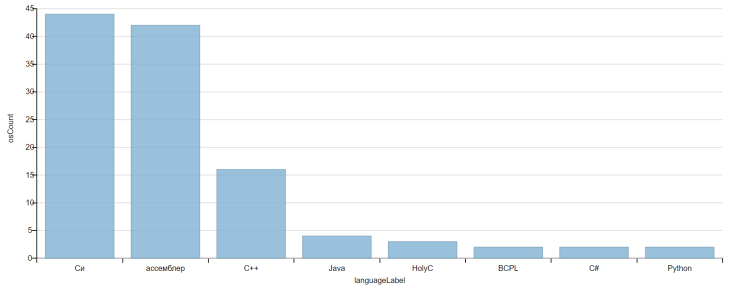


Рис. 10.3. Первые восемь языков, на которых написано больше всего операционных систем, 2020 год

Количество программ для каждой операционной системы

Запрос 10.6 позволяет подсчитать количество программ, написанных для каждой операционной системы. Лидером среди операционных систем по количеству написанных для них программ является Linux (Q388) — система, для которой написано 9223 программы. Для операционных систем Microsoft Windows (Q1406) и AIX (Q269856) (операционная система производства компании IBM) создано 3278 и 2337 программ соответственно.

Запрос 10.6. Количество программ для каждой операционной системы[10]

```
# Number of software for each operating system
SELECT ?os ?osLabel (COUNT(*) AS ?softCounter) WHERE
{   # the software ?soft works on the operating system ?os
    ?soft p:P306 [ps:P306 ?os].
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
}
GROUP BY ?os ?osLabel
ORDER BY DESC(?softCounter)
```

[10] Для 551 операционной системы существовали какие-либо написанные под них программы в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/ugp.

Сколько компьютерных программ было написано для операционной системы с использованием того или иного языка программирования

С помощью запроса 10.7 можно увидеть для каждой операционной системы, какова доля вклада разных языков программирования, использованных в программах для этих систем. Например, оказалось, что большая часть программ, написанных для macOS, создавалась с использованием C++ (374 программы), Си (276 программ), Python (107 программ). Программы для системы Android пишут на языках C++ (107 программ) и Java (80 программ). Под iOS большинство программ написано на C++ (63 программы).

Запрос 10.7. Гистограмма операционных систем с количеством программ, написанных на каком-либо языке программирования[11]

```
1  #defaultView:BarChart
2  SELECT ?os ?osLabel (COUNT(*) AS ?softCount) ?softLang ?softLangLabel WHERE
3  {
4      ?soft wdt:P306 ?os;       # software works on OS
5            wdt:P277 ?softLang. # it is written in progr. language
6      ?os wdt:P277 ?osLang.     # OS is written in progr. language
7      SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en"}
8  }
9  GROUP BY ?os ?osLabel ?softLang ?softLangLabel
10 ORDER BY DESC(?count) DESC(?osLabel)
```

Вопрос на внимание. Зачем нужна строка 6 в запросе 10.7, если переменная `?osLang` используется в скрипте только в этой строке? Можно ли её заменить на безымянную переменную?

[11] 704 компьютерные программы были написаны для какой-либо операционной системы с использованием того или иного языка программирования в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/93xD.

При построении гистограммы на рис. 10.4 использовался запрос, аналогичный запросу 10.7, разница только в строке 2, сравните эту строку в старом и новом запросе (всего лишь переставлены переменные):

```
SELECT ?os ?osLabel (COUNT(*) AS ?softCount) ?softLang ?softLangLabel WHERE
SELECT ?softLang ?softLangLabel (COUNT(*) AS ?softCount) ?os ?osLabel WHERE.
```

Запрос 10.7 строит «столбики» ОС, состоящие из «кирпичиков» языков программирования, на которых пишут программы для этих ОС. А новый запрос[12] строит «столбики» языков программирования, состоящие из «кирпичиков» — тех ОС, для которых пишут программы на этих языках (рис. 10.4).

Итак, гистограмма на рис. 10.4 построена с помощью нового запроса https://w.wiki/93xM. Этот рисунок позволяет увидеть для каждого языка программирования количество программ, которые были на нём написаны, а также под какими ОС работают данные программы. Из графика видно, что наибольшее число программ пишется на языках Си (2566 программ), C++ (2503 программы), Java (799 программ), Python (717 программ) и JavaScript (344 программы).

Рассмотрим эти языки подробнее. Большая часть программ на языке C++ пишется под Windows (472 программы) и macOs (300 программ), на языке C — под Windows (700 программ), macOS (400 программ) и Linux (400 программ). Язык программирования C был разработан в 1972 году, но, по-видимому, языки C и C++ будут ещё долго лидировать, в том числе благодаря тому, что используются для написания низкоуровневых приложений и взаимодействия с аппаратными устройствами[13].

Большая часть программ на языке Java пишется под macOS (196 программ) и Андроид (156 программ). Вероятно, язык Java пользуется популярностью за счёт переносимости кода[14], то есть код на языке Java можно запустить на любом компьютере, где установлена виртуальная машина[15].

На языке JavaScript пишут под macOS (100 программ), Андроид (60 программ) и iOS (40 программ). Как правило, JavaScript используется для написания клиентской части веб-приложений.

Язык Python популярен в операционных системах macOS (212 программ) и Linux (107 программ). Этот язык используется в том числе для написания веб-приложений и анализа данных.

Гистограмма на рис. 10.4 показывает, что каждый из рассмотренных языков занял свою «нишу» в области разработки программ и применяется для определённого круга задач. Отметим, что большая часть программ пишется под macOS (900 программ), Windows (1500 программ), Linux (1200 программ) или Андроид (300 программ), как это можно увидеть в результате скрипта 10.6.

[12] URL: https://w.wiki/93xM.

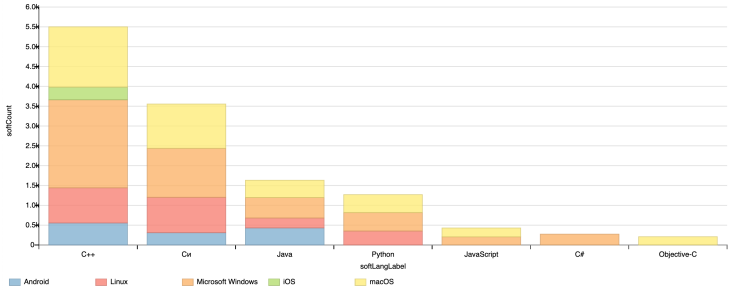


Рис. 10.4. Число программ с разбивкой по языкам программирования и операционным системам, 2020 год

[13] *Кемп Д.* Будущее языков программирования / пер. А. Панин. 2016. URL: http://rus-linux.net/MyLDP/algol/the_future_of_computer_languages.html.

[14] Переносимость — это возможность запускать код на множестве платформ без необходимости внесения каких-либо изменений в код.

[15] Java virtual machine (JVM) — это среда выполнения, в которой может выполняться байт-код, полученный в результате компиляции компьютерных программ, написанных на языке программирования Java.

Документирование программного обеспечения

Викиданные играют большую роль в документировании программного обеспечения. Это показано в статье 2020 года[16] на примере программ, входящих в среды GNOME и KDE. В этой статье показано, что если в английском разделе Википедии описаны почти все программы, входящие в состав графических интерфейсов GNOME и KDE, то в итальянском и французском разделах есть только часть статей. Документирование больших проектов — это известная и трудная задача. Для её решения нужна централизованная система. Именно в этой роли и выступает связка Википедия и Викиданные[17].

Упражнения

1. Вывести список операционных систем с информацией о разработчиках. Ответ на с. 173.
2. Вывести список операционных систем с их логотипами. Ответ на с. 173.
3. Найти страны происхождения операционных систем. Ответ на с. 173.
4. Создать запрос, создающий диаграмму в виде дерева (рис. 10.4). Строчки верхнего уровня должны содержать операционные системы. При «разворачивании» мы должны видеть список операционных систем, которые основывались на системе из строчки верхнего уровня. Ответ на с. 174.

[16] *Samuel J.* Documenting Software Applications on Wikidata. 2020. URL: https://jsamwrites.medium.com/documenting-software-applications-on-wikidata-197716d7f2d.

[17] Там же.

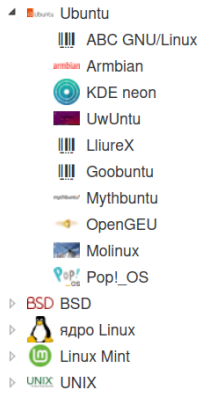


Рис. 10.5. Дерево операционных систем, при этом «вложенные» системы основаны на базовых системах, размещённых на верхнем уровне дерева

11. Где учатся и кем работают изобретатели языков программирования

В главе исследуются свойства языков программирования на основе Викиданных. С помощью SPARQL-запросов, вычисляемых на объектах типа «язык программирования», решён ряд задач. Получены перечни всех языков программирования под пермиссивными лицензиями и языков с закрытыми лицензиями и рассчитано их процентное соотношение. Построена пузырьковая диаграмма по количеству разных расширений файлов для одного языка программирования. Получены карты, отображающие месторасположение учебных заведений и компаний, в которых учились или работали люди, связанные с созданием языков программирования. Построена пузырьковая диаграмма, отображающая популярные профессии среди людей, причастных к созданию и разработке языков программирования. Получен список всех объектно-ориентированных языков программирования и сделан вывод об исчерпывающей полноте Викиданных относительно них. Проведены сравнение и анализ результатов SPARQL-запросов за разные годы, отмечены основные изменения.

Перечислим используемые в этой главе объекты Викиданных:

- programming language — язык программирования;
- object-based language — объектно-ориентированный язык программирования.

И используемые в SPARQL-запросах свойства объектов Викиданных:

- influenced by — какие языки программирования оказали влияние;
- developer — кто разработал язык программирования;
- copyright license — какая лицензия;
- instance of — к какому более общему типу (типам) относится этот язык;

- file extension — расширение файлов;
- headquarters location — место расположения штаб-квартиры разработчиков языка;
- coordinate location — географические координаты объекта;
- educated at — где учился объект;
- place of birth — где родился объект;
- occupation — род занятий объекта (профессия).

Список языков программирования

Выведем (запрос 11.1) список всех языков программирования, с которыми будем работать дальше. Для этого используем объект programming language (Q9143) и свойство instance of (P31).

Запрос 11.1. Список языков программирования[1]

```
#List of programming languages
SELECT ?lang ?langLabel WHERE
{
    ?lang wdt:P31 wd:Q9143. # instances of programming language
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru"}
}
```

На 2020 год наиболее проработанными на Викиданных языками программирования были: C++ (26 свойств), Java (26 свойств), JavaScript (25 свойств), R (25 свойств). По данным ProWD, почти пустыми и малоинформативными языками на 2020 год оказались: Tiny, Proteus, Comfy — всего по одному свойству. Недостаток полученного списка в том, что ряд объектов получился безымянным на Викиданных (No label defined). Попробуем получить список языков, у которых поле `label` на русском языке будет непустым (запрос 11.2).

Соотнесите язык программирования и его разработчиков.

Язык	Разработчик
Ада	Жан Ишбиа и С. Такер Тафт
Erlang	Чарльз Мур
Форт	Джо Армстронг, Роберт Вирдинг и шведская компания Ericsson

См. ответ на с. 174.

[1] Получено: 732 языка программирования в 2017 году и 1422 языка в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/uGs.

Запрос 11.2. Список языков программирования с заполненным свойством label на русском языке[2]

```
SELECT ?lang ?lang_label WHERE {
    ?lang wdt:P31 wd:Q9143
    ; rdfs:label ?lang_label FILTER (LANG(?lang_label) = "ru") .
}
```

[2] Получено: 630 языков программирования с заполненным свойством label на русском языке в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/v2a.

Операции над множествами в SPARQL

Получим список всех языков программирования, являющихся открытым программным обеспечением и испытавших на себе влияние хотя бы одного из следующих языков программирования: Си, Python, Java. При этом пусть в разработке этих языков не участвуют следующие две фирмы: Sun Microsystems, Космический центр имени Линдона Джонсона. Таким образом, из множества «открытых» языков нужно выбрать множество языков, окружающих Си, Python, Java, и вычесть языки двух фирм. В итоговом запросе 11.3 нужно выполнить операции над множествами с помощью команд `UNION` и `MINUS`.

Запрос 11.3. Список свободных языков программирования[3]

```
SELECT DISTINCT ?prog ?progLabel WHERE
{
  ?prog wdt:P31 wd:Q9143 # instance of programming language
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
  {
    { ?prog wdt:P737 wd:Q15777 } UNION # influenced by C
    { ?prog wdt:P737 wd:Q28865 } UNION # influenced by Python
    { ?prog wdt:P737 wd:Q251 } UNION # influenced by Java
    { ?prog wdt:P31 wd:Q341 } # is free software
  } MINUS
  {         # developer
    { ?prog wdt:P178 wd:Q14647 } UNION # developer Sun Microsystems
    { ?prog wdt:P178 wd:Q208371 } # developer Lyndon Johnson Space Center
  }
}
```

[3] Получено: в 2017 году 115 таких языков, в 2020 году — 125 языков. Ссылка на SPARQL-запрос: https://w.wiki/v2n.

Пермиссивные лицензии

С помощью запроса 11.4 построим список языков программирования, находящихся под пермиссивными лицензиями, то есть такими лицензиями на свободное программное обеспечение, которые почти не ограничивают свободу действий пользователей и разработчиков.

Запрос 11.4. Языки программирования с пермиссивными лицензиями[4]

```
SELECT DISTINCT ?lang ?langLabel WHERE
{
  ?lang wdt:P31 wd:Q9143. # is programming language
  {?lang wdt:P275 wd:Q308915 } UNION # Mozilla Public
  {?lang wdt:P275 wd:Q334661 } UNION # MIT license
  {?lang wdt:P275 wd:Q191307 } UNION # BSD licenses
  {?lang wdt:P275 wd:Q6905323}       # Creative Commons

  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}.
}
```

Рассмотрим соотношение числа языков с пермиссивной лицензией и языков с проприетарными (закрытыми) лицензиями. Запрос 11.5 состоит из нескольких частей:

строки 5–12: в первом подзапросе подсчитываем число `?free` свободных языков программирования;

строки 14–22: во втором — число `?not_free` «закрытых» языков;

строка 3: вычисляем отношение числа свободных и несвободных языков.

Подсчитайте, больше ли пишут программ под пермиссивными лицензиями те, кто выбирает для работы язык с пермиссивной лицензией?

[4] Получено: 37 языков программирования, находящихся под пермиссивными лицензиями, в 2017 году и 83 языка в 2020 году. В этот список из 83 «свободных» языков попали, например, CoffeeScript, Go, Haml. Ссылка на SPARQL-запрос: https://w.wiki/5CLF.

Запрос 11.5. Расчёт отношения числа свободных языков к числу закрытых[5]

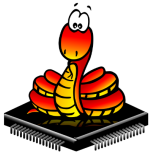
```
#The script calculates the percentage of programming languages
#with a free license in relation to languages with a closed license
SELECT (COUNT(?not_free)* 100 / (COUNT(?free)) as ?total) WHERE
{{
    SELECT ?free WHERE {
      ?free wdt:P31 wd:Q9143 # instances of programming language
       SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
       { ?free wdt:P275 wd:Q308915  }  UNION  # license Mozilla Public
       { ?free wdt:P275 wd:Q334661  }  UNION  # license MIT
       { ?free wdt:P275 wd:Q191307  }  UNION  # license BSD
       { ?free wdt:P275 wd:Q6905323 }         # license CC
    }
} UNION {
    SELECT ?not_free WHERE {
      ?not_free wdt:P31 wd:Q9143 # instances of programming language
      SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
      { ?not_free wdt:P275 wd:Q6165015 } UNION # Java Research License
      { ?not_free wdt:P275 wd:Q218616 } UNION # proprietary software
      { ?not_free wdt:P275 wd:Q3238057 } UNION # proprietary license
      { ?not_free wdt:P275 wd:Q31202214 } UNION # proprietary software
      { ?not_free wdt:P275 wd:Q979794 } # Aladdin Free Public License
    }}
}
```

Количество форматов файлов исходного кода

В зависимости от языка программирования файлы с исходным кодом программ могут иметь разные расширения. Расширение имени файла — это последовательность символов, добавляемых к имени файла и предназначенных для идентификации типа (формата) файла. Это один из распространённых способов, с помощью которых пользователь или программное обеспечение компьютера может определить тип данных, хранящихся в файле, например: `file.jpg` — это фотография, `file.avi` — видеофайл.

Приведённые ниже изображения — логотипы разных языков программирования. Какое из изображений является логотипом языка LOLCODE?



См. ответ на с. 174.

[5] Запрос подсчитывает отношения числа языков программирования со свободной лицензией к числу языков с закрытой лицензией. На 2020 год это значение равно 26 %. Ссылка на SPARQL-запрос: https://w.wiki/v3B.

С помощью запроса 11.6 построим пузырьковую диаграмму, показывающую долю различных форматов файлов исходного кода. Сравним диаграммы, построенные в 2017 году (рис. 11.1) и в 2020 году (рис. 11.2) по этому запросу.

На рис. 11.1 видно, что на 2017 год самыми исторически богатыми на форматы и расширения файлов оказались такие языки программирования, как: C++ (10 форматов), Geometric Description Language (8 форматов), Racket (7 форматов). Таким образом, одному языку программирования может соответствовать больше одного расширения имени файла. Например, файлы с программой на языке Racket могут иметь расширения rkt, rktl, rktd, scrbl, plt, ss или scm.

Запрос 11.6. Количество форматов файлов исходного кода

```
#defaultView:BubbleChart
SELECT ?lang_name (count(*) as ?count) WHERE
{
    ?lang wdt:P31 wd:Q9143. # instance of programming language
  ?lang wdt:P1195 ?count. # file extension
  ?lang rdfs:label ?lang_name FILTER (lang(?lang_name) = "ru,en").
}
GROUP BY ?lang_name
ORDER BY DESC(?count)
```

К 2020 году (рис. 11.2) C++ и Geometric Description Language (GDL) остались на лидирующем месте (10 и 8 форматов-расширений). За три года подтянулись и вошли в первую восьмёрку также такие языки, как Racket, Raku (9 форматов), REXX и Scratch (по 6 форматов), Java и Wolfram Language (по 5 форматов).

Из этого можно сделать вывод, что развитие языков программирования продолжается непрерывно, постоянно возникают новые форматы файлов исходного кода, но лидеры по числу расширений файлов с трудом уступают лидирующие позиции. Это может быть связано с тем, что, раз возникнув, язык уже с трудом может избавиться от расширения ввиду опасности потери совместимости с более ранними версиями языка. Таким образом, наличие множества расширений, скорее, указывает на отсутствие единого стандарта и строгих правил на первых порах.

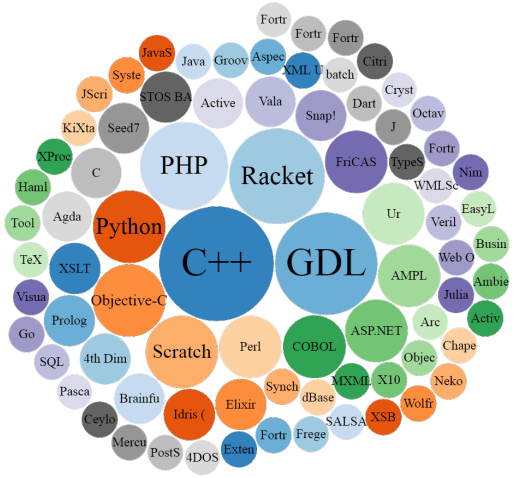


Рис. 11.1. Соотношение числа разных форматов файлов исходного кода по языкам программирования, 2017 год

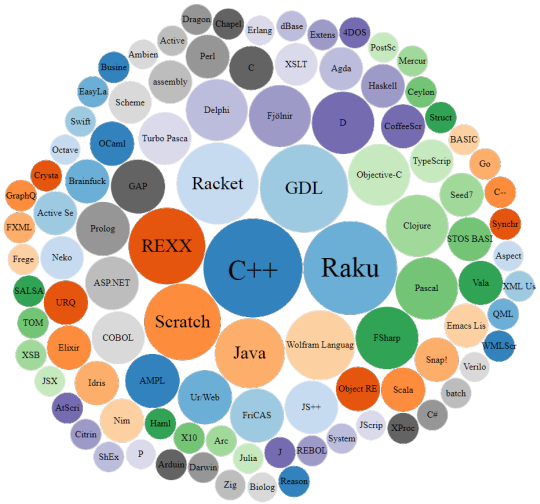


Рис. 11.2. Соотношение числа разных форматов файлов исходного кода для 108 языков программирования, 2020 год

Страны, в которых располагаются организации и проживают люди, связанные с разработкой языков программирования

Отобразим на карте мира те страны, в которых живут люди и располагаются организации, связанные с созданием языков программирования. Заметим, что разработчиком языка (свойство: developer (P178), см. запрос 11.7) может выступать как организация, так и отдельный человек (далее по тексту под «разработчиком» понимается одно из этих двух понятий). Для определения месторасположения (свойство: coordinate location (P625)) организации будем использовать координаты её штаб-квартиры (свойство: headquarters location (P159)), для человека — координаты места его рождения (свойство: place of birth (P19)).

Более точным подходом было бы использование не только места рождения, но и тех мест, где человек живёт. В Викиданных это описывается свойством residence (P551). Например, фрагмент страницы Альберта Эйнштейна со свойством *residence* в Викиданных (рис. 11.3) можно перевести на человеческий язык так: «Альберт Эйнштейн жил в городке Капут в Германии с 1929 по 1932 год».

Трудность в том, что это свойство *residence* редакторы Викиданных достаточно редко заполняют, сейчас его можно увидеть только у самых известных людей с наиболее проработанными объектами, например у Альберта Эйнштейна. Поэтому в запросе 11.7 мы учитываем только место рождения.

Запрос 11.7. Карта с указанием места жительства или работы разработчиков языков программирования[6]

```
#defaultView:Map
SELECT ?lang_label ?developerLabel ?locationLabel ?coord
WHERE
{
   ?lang wdt:P31 wd:Q9143. # instances of programming language
   ?lang wdt:P178 ?developer. # developer
   { ?developer wdt:P159 ?location. } UNION # headquarters location
   { ?developer wdt:P19 ?location. } # place of birth
   ?location wdt:P625 ?coord. # coordinate location
   SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

По рис. 11.4 и 11.5 можно сделать вывод, что наиболее благоприятными местами для разработки языков программирования являются Восточное побережье США, Центральная Европа и Великобритания.

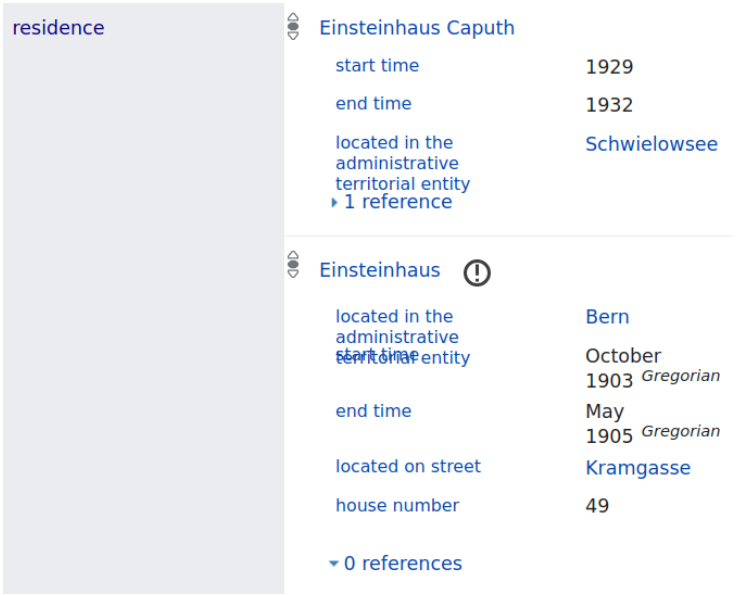


Рис. 11.3. Место жительства Альберта Эйнштейна, фрагмент страницы объекта Albert Einstein (Q937), 2023 год

[6] Результатом запроса является карта, где красными точками указаны места проживания людей или штабы фирм, причастных к разработке языков программирования. На рис. 11.4 изображен результат запроса на 2017 год, а на рис. 11.5 показаны аналогичные данные на 2020 год.

Рис. 11.4. Страны, в которых живут люди или расположены организации, связанные с созданием языков программирования, 2017 год

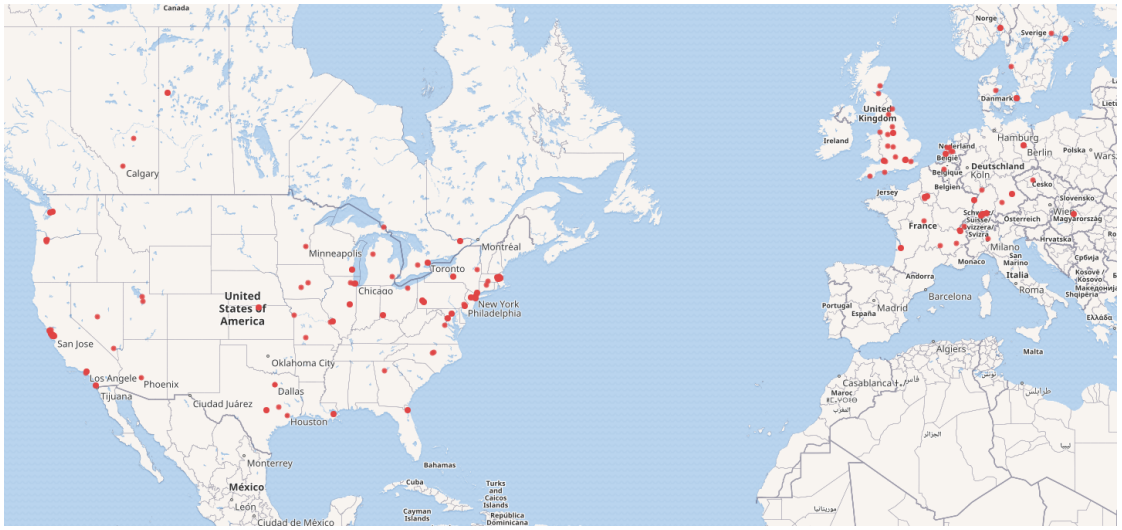


Рис. 11.5. Страны, в которых живут люди или расположены организации, связанные с созданием языков программирования, 2020 год

С помощью запроса 11.8 построим пузырьковую диаграмму стран, наиболее благоприятных для размещения штаб-квартир и/или рождения разработчиков языков программирования (рис. 11.6). Размер пузырька на такой диаграмме соответствует числу людей в стране, причастных к разработке языков программирования.

Видим на рис. 11.6, что такой благоприятной страной оказались, в первую очередь, Соединённые Штаты Америки (241 штаб-квартира и/или человек). Далее (около двух дюжин штаб-квартир и/или человек) идут Швейцария, Великобритания и Франция. В России подобных штаб-квартир и/или родившихся разработчиков языков оказалось 4. С помощью модификации запроса 11.8 получим следующий список языков программирования, разработанных в России, вместе с разработчиками и штаб-квартирами (см. ссылку на SPARQL-запрос: https://w.wiki/6g2c):

- язык программирования РЕФАЛ (Q2626418); первую версию языка придумал в 1968 году В. Ф. Турчин, родившийся в Подольске в Московской области;
- язык программирования Аналитик (Q4064746) был разработан также в 1968 году в Институте кибернетики АН УССР под руководством В. М. Глушкова, родившегося в Ростове-на-Дону;
- встроенный язык программирования 1С:Предприятие (Q65065977) начали разрабатывать в 1996 году, штаб-квартира компании 1С находится в Москве;
- язык программирования Factor (Q1391724) придуман и разрабатывается С. Пестовым, родившемся в Томске.

Запрос 11.8. Пузырьковая диаграмма благоприятных стран для появления разработчиков языков программирования[7]

```
#defaultView:BubbleChart
SELECT ?countryLabel (count(*) as ?count) WHERE
{ ?lang wdt:P31 wd:Q9143;    # instance of programming language
        wdt:P178 ?developer. # developer of language
  { ?developer wdt:P159 ?location. } UNION # headquarters location
  { ?developer wdt:P19  ?location. }       # place of birth
  ?location wdt:P17 ?country. # location in country
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
GROUP BY ?countryLabel
ORDER BY DESC(?count)
```

Рис. 11.6. Наиболее благоприятные страны для разработчиков языков программирования, 2020 год

Сколько языков программирования появилось за последние три года? В каком году было изобретено максимальное число языков?

[7] Ссылка на SPARQL-запрос: https://w.wiki/94QJ.

Университеты, в которых учились разработчики языков программирования

Запрос 11.9 строит карту учебных заведений, в которых учились студенты, впоследствии разработавшие языки программирования. По рис. 11.7 и 11.8 видно, что большая часть разработчиков языков программирования училась в Европе и США и в 2017-м, и в 2020 году.

Запрос 11.9. Карта с указанием университетов, в которых учились разработчики языков программирования[8]

```
1  #defaultView:Map
2  SELECT ?langLabel ?developerLabel ?eduLabel ?coord WHERE {
3    ?lang wdt:P31 wd:Q9143;    # instances of programming language
4          wdt:P178 ?developer. # developed by developer
5    ?developer wdt:P69 ?edu.   # who educated at ?edu
6                               ?edu wdt:P625 ?coord.  # coordinates of ?edu (university)
7    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en" }
8  }
```

Запрос 11.10. Карта с указанием координат университетов разработчиков языков программирования[9]

```
1  #defaultView:Map
2  SELECT ?langLabel ?developerLabel ?coord WHERE {
3    ?lang wdt:P31 wd:Q9143;    # instances of programming language
4          wdt:P178 ?developer. # developed by developer
5     ?developer wdt:P69 [wdt:P625 ?coord]. # who educated at somewhere with coordinates
6    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en" }
7  }
```

Мы привели два запроса 11.9 и 11.10 и выделили в них соответственно строки 5–6 и 5, чтобы показать — как можно сворачивать код за счёт использования безымянной переменной.

Безымянной переменной в запросе 11.10 в строке 5 соответствует переменная с именем `?edu` в запросе 11.9. К сожалению, конкретно в этом примере «сворачивание» утрачивает переменную `?eduLabel` и при наведении курсора на точку на карте окошечко с информацией всплывает, но без названия университета.

Рис. 11.7. Учебные заведения, в которых учились разработчики языков программирования, 2017 год

[8] Результатом запроса будет карта, на которой красными точками отмечены места расположения университетов, в которых учились люди, создавшие языки программирования. На 2017 год получено 142 университета, к 2020 число записей увеличилось до 282. Ссылка на SPARQL-запрос: https://w.wiki/94hh.

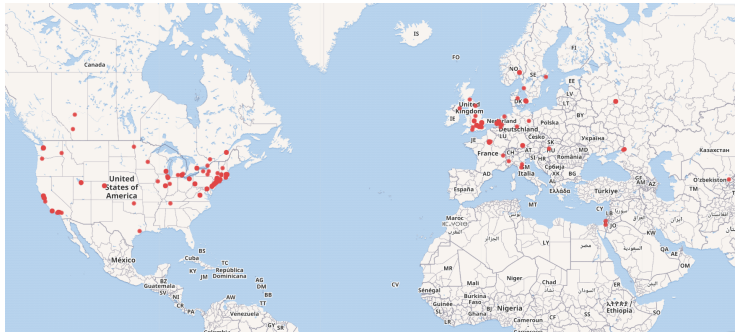


Рис. 11.8. Учебные заведения, в которых учились разработчики языков программирования, 2020 год

[9] Результатом запроса будет та же карта, что и в предыдущем запросе, но при наведении курсора на красную точку название университета не показывается, поскольку переменной `?eduLabel` нет в запрашиваемых результатах этого запроса. Ссылка на SPARQL-запрос: https://w.wiki/94hY.

Построим пузырьковую диаграмму (запрос 11.11) по самым популярным среди будущих создателей языков программирования учебным заведениям. На первых местах по числу обучающихся разработчиков оказались: Принстонский университет и Стэнфордский университет (по 8 студентов). Отметим, что в этот достойный список университетов попал Московский государственный университет. В МГУ учились три таких студента (см. запрос: https://w.wiki/6gC3):

- Энтони Ричард Хоар (Q92602), разработавший язык ALGOL W (Q1538458) в 1958 году;
- снова В. Ф. Турчин, разработавший РЕФАЛ (Q2626418);
- снова В. М. Глушков, под чьим руководством разработан язык Аналитик (Q4064746).

Запрос 11.11. Университеты, в которых учились разработчики языков программирования[10]

```
#defaultView:BubbleChart
SELECT ?eduInstitutionLabel (count(*) as ?count)
WHERE {
  ?lang wdt:P31 wd:Q9143;      # is programming language
        wdt:P178 ?developer.   # developer
  ?developer wdt:P69 ?eduInstitution. # educated at
  ?eduInstitution wdt:P625 ?coord. # location
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
GROUP BY ?eduInstitutionLabel
ORDER BY DESC(?count)
```

Профессии создателей языков программирования

С помощью запроса 11.12 построим пузырьковую диаграмму, отображающую информацию о том, какие профессии преобладают среди людей, разрабатывающих языки программирования. Результаты этого запроса в 2017 и 2020 годах можно видеть на рис. 11.9 и 11.10 соответственно. Размер пузырька на рисунках показывает число разработчиков с соответствующей профессией.

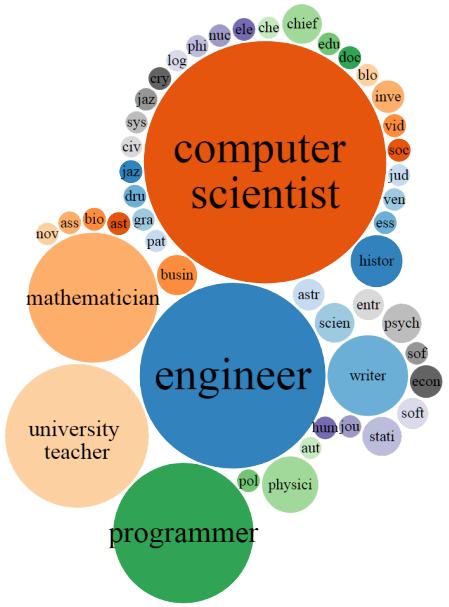


Рис. 11.9. Профессии разработчиков языков программирования, 2017 год

[10] Получено: 168 университетов в 2022 году и 396 университетов в 2024 году. Ссылка на SPARQL-запрос: https://w.wiki/5CMt.

Запрос 11.12. Профессии создателей языков программирования[11]

```
#defaultView:BubbleChart
SELECT ?occupationLabel (count(*) as ?occupation) WHERE
{ ?lang wdt:P31 wd:Q9143;           # instances of programming language
        wdt:P178 ?developer.        # developed by developer
                 ?developer wdt:P106 ?occupation. # with occupation
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
GROUP BY ?occupationLabel
ORDER BY DESC(?count)
```

Наиболее распространенными профессиями оказались: специалист в области компьютерных наук, инженер и учитель. Встречаются даже такие профессии, как джазовый музыкант и политик. Например, обеими этими профессиями владел Герберт Александер Саймон (1916–2001), разработчик Языка обработки информации (IPL). На 2020 год среди разработчиков языков программирования оказалось больше всего специалистов в области компьютерных наук (172 человека), а также 96 инженеров, 57 учителей, 56 программистов и 43 математика.

Объектно-ориентированные языки программирования

С помощью запроса 11.13 получим список всех объектно-ориентированных языков программирования. Подсчитаем отношение числа объектно-ориентированных языков программирования (запрос 11.13 вернул 118 языков) ко всем языкам программирования (запрос 11.1 вернул 1422 языка). Получили, что в 2020 году 8 % языков программирования были объектно-ориентированными.

Запрос 11.13. Список объектно-ориентированных языков программирования[12]

```
SELECT DISTINCT ?lang ?langLabel
WHERE {
 ?lang wdt:P31 wd:Q899523 # is object-oriented programming language
 SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

В этом запросе переменная `?developer` нужна только как мостик, чтобы соединить язык программирования (`?lang`) и профессию разработчика (`?occupation`). По аналогии с запросами 11.9 и 11.10 преобразуйте этот запрос так, чтобы сохранить упомянутый мостик, но убрать переменную `?developer`, то есть сделать её безымянной. При этом запрос станет короче на одну строчку.

[11] Получено: 48 профессий в 2017 году и 74 профессии в 2020 году. Ссылка на SPARQL-запрос: https://w.wiki/94jS.

Рис. 11.10. Профессии разработчиков языков программирования, 2020 год

[12] Получено: 116 объектно-ориентированных языков программирования на 2017 год и 118 языков на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/v47.

Степень заполненности имён разработчиков языков программирования на русском языке

Сравним конструкции `serviceParam` и `rdfs:label`, с помощью которых можно получить метки (имена) объектов. В запросе 11.14 используется конструкция serviceParam, которая позволила получить имена 261 разработчика языков. В запросе 11.15 в строках 4 и 7 используется конструкция `rdfs:label`, которая позволила получить имена 261 разработчика на английском языке (языковой код — en) и 150 разработчиков с именами на русском языке. Таким образом, если метка на русском языке не заполнена, то конструкция `serviceParam` возвращает безымянный объект с номером объекта вместо имени. Запросы с конструкциями `rdfs:label` являются более строгими и пропускают объекты без меток на заданном языке.

Запрос 11.14. Список названий языков программирования и их разработчиков, найденных с помощью конструкции serviceParam[13]

```
SELECT ?lang ?langLabel ?developer ?developerLabel WHERE {
  ?lang wdt:P31 wd:Q9143;    # ?lang is programming language
        wdt:P178 ?developer. # has developer
  ?developer wdt:P31 wd:Q5.  # developer is human
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

[13] Получено: 261 разработчик в 2023 году. Ссылка на SPARQL-запрос: https://w.wiki/6gDu.

Запрос 11.15. Список названий языков программирования и их разработчиков, найденных с помощью конструкции rdfs:label[14]

```
1 SELECT ?lang ?langLabel ?developer ?developerLabel WHERE {
2   ?lang wdt:P31 wd:Q9143.   # ?lang is programming language
3   ?lang wdt:P178 ?developer # has developer
4   ; rdfs:label ?langLabel FILTER (LANG(?langLabel) = "ru").
5
6   ?developer wdt:P31 wd:Q5 # instances of human
7   ; rdfs:label ?developerLabel FILTER (LANG(?developerLabel) = "ru").
8 }
```

[14] Получено: 150 разработчиков с именами на русском языке в 2023 году. Ссылка на SPARQL-запрос: https://w.wiki/6gDx.

Упражнения

1. Вывести все языки программирования со свойством «персонаж-талисман (P822)» (узнаваемый персонаж, олицетворяющий собой некий коллектив: школу, спортивную команду, сообщество, воинское подразделение, мероприятие или бренд). Ответ на с. 175.

2. Подчитать количество языков программирования, созданных ранее 1992 года (используйте свойство: «дата-основания/создания (P571)»). Ответ на с. 175.

3. Построить столбчатую диаграмму, отражающую количество известных хештегов[15] в Твиттере для каждого языка программирования (используйте свойство «хештег Твиттера (P2572)»). Ответ на с. 175.

[15] Хештег (#) — ключевое слово, используемое в микроблогах и социальных сетях, облегчающее поиск сообщений по теме или содержанию и начинающееся со знака решётки.

12. Военные корабли и их операторы

Глава посвящена отечественным и зарубежным кораблям. Они могут иметь как военное, так и гражданское назначение. Гражданские суда используются в грузоперевозках, рыболовстве, туризме, разведке полезных ископаемых, спасательных работах, а также в спортивных, культурных и других целях. Для хранения информации о судах и других объектах ведутся базы знаний. Одной из таких баз знаний являются Викиданные. В этой главе изучены хранимые в Викиданных объекты кораблей и проведена оценка качества и полноты их описания.

Список кораблей

Анализ экземпляров объекта корабль (Q11446) с помощью сервиса ProWD в 2020 году показал, что индекс Джини равен 0.239 (рис. 12.1). Сравнение кораблей со странами в Викиданных (у стран индекс Джини равен 0.091, см. рис. 8.1 на с. 82) говорит, что корабли не так хорошо и равномерно заполнены, как страны, есть достаточно кораблей как с большим количеством заполненных свойств, так и с малым.

Построим список всех кораблей с помощью запроса 12.1.

Запрос 12.1. Список кораблей[1]

```
# List of ships
SELECT ?ship ?shipLabel WHERE {
  ?ship wdt:P31 wd:Q11446. # instance of ship
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
}
```

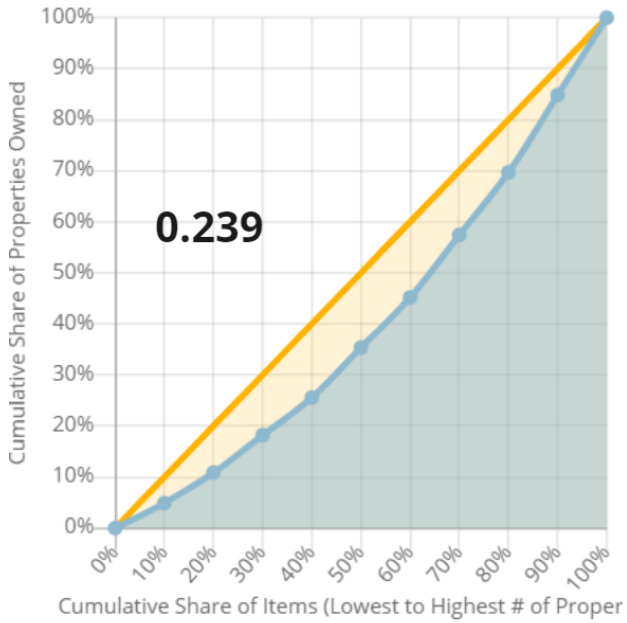


Рис. 12.1. Индекс Джини — равномерность заполнения свойств «кораблей», 2020 год

[1] Получено: 19 820 кораблей в 2017 году, 50 681 — в 2020-м, 71 203 — в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/wX6.

По данным ProWD, больше всего свойств (34 свойства) имеет ледокол «Красин» (Q281147), а меньше всего, по пять свойств, у кораблей «Ливень» (Q99198666) и Dispatch (Q28155282).

В Викиданных, как правило, записывается не прямая принадлежность корабля стране, а принадлежность некоторому оператору. Чаще всего это какая-либо организация, например Военно-морской флот Российской Федерации (Q465283). Составим список кораблей, операторы которых находятся или находились в России, СССР или Российской империи (см. запрос 12.2).

Запрос 12.2. Список кораблей с отечественными операторами[2]

```
1 # List of ships from Russia, Soviet Union and Russian Empire
2 SELECT ?ship ?shipLabel WHERE {
3   VALUES ?country {wd:Q34266 # Russian Empire
4                    wd:Q15180 # Soviet Union
5                    wd:Q159}  # Russia
6   ?ship wdt:P31 wd:Q11446;         # instance of ship
7         wdt:P137/wdt:P17 ?country; # operator in country
8   SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
9 }
```

Обратите внимание на строку 7 в запросе 12.2. В этой строке записаны последовательно два свойства `wdt:P137/wdt:P17`:

- `wdt:P137` — свойство operator (P137) связывает корабль и оператора судов;
- `wdt:P17` — свойство country (P17) связывает оператора судов и страну.

Таким образом, эта строка позволяет кратко и напрямую задать страну для корабля посредством свойства «оператор». Такое последовательное указание свойств (`wdt:P137/wdt:P17`) является аналогом безымянных переменных (`[]`), поскольку в текущем запросе 12.2 мы не получаем в явном виде никаких операторов судов.

Найдите «корабль Гиннесса» по какому-либо параметру. Например, напишите скрипт для поиска самого большого, самого длинного или самого вместительного корабля.

См. ответ на с. 177.

[2] Получено: 107 кораблей в 2017 году, 578 кораблей в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/6gHD.

Пример использования безымянной переменной (`[]`) см. в запросе 6.8 на с. 57.

Полнота Викиданных по числу кораблей

Поиск точного количества кораблей в мире — трудная задача. Ведь данные о некоторых из них являются совершенно секретными, другие — частные суда, информации о них тоже нет. Предположим, что общее число кораблей примерно равно 1 600 000, как указано в базе данных судов[3]. На 2021 год Викиданные содержали только 71 206 кораблей, что составило 4,5 % от их общего числа.

Что касается российских кораблей, то в состав российского военного и гражданского флотов входит 17 657 кораблей[4]. В это же время запрос 12.2 возвращает лишь 578 кораблей, что составляет 3,27 % от общего числа российских кораблей.

В обоих случаях разница между фактическим количеством кораблей и результатом запросов огромная, что говорит о неполноте Викиданных.

Полнота свойств объектов военных кораблей

Составим список кораблей, участвовавших в каких-либо конфликтах, с помощью запроса 12.3.

Запрос 12.3. Список кораблей, участвовавших в каких-либо конфликтах[5]

```
# List of ships with countries and war conflicts
SELECT ?ship ?shipLabel ?countryLabel ?conflict ?conflictLabel
WHERE
{
  ?ship wdt:P31 wd:Q11446;         # instance of ship
        wdt:P137/wdt:P17 ?country;# belongs to country
        wdt:P607 ?conflict.        # engaged in some conflict
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
}
```

[3] FleetMon Tracking the Seven Seas. URL: https://www.fleetmon.com/vessels/.

[4] URL: http://russianships.info/today/.

На марке, выпущенной в 1982 году, изображён самый известный советский эсминец проекта 7, удостоенный звания «Гвардейский». Назовите его.

См. ответ на с. 178.

[5] Получено: 1400 кораблей в 2017 году, 3567 кораблей в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/vum.

У военных кораблей, участвовавших в сражениях, указывается свойство conflict (P607) (война/сражение). В то же время военные конфликты и военные операции, которые являются частью войн, — это разные понятия. Корабли в Викиданных можно условно поделить на два типа:

1. Корабли, у которых военные операции перечисляются в одном ряду с военными конфликтами. Например, у эсминца «Гремящий» (Q4148613) 10 войн/сражений, см. запрос 12.4. Такое большое число связано с тем, что корабль принял участие во многих арктических конвоях, которые являются военными операциями.

2. Корабли, у которых военные операции отделены от военных конфликтов. Например, у британского крейсера HMS Trinidad (Q1565575) участие в военной кампании и арктическом конвое указаны как часть Второй мировой войны с помощью квалификатора including (P1012). Таким образом, в Викиданных у этого крейсера указана одна война/сражение.

У кораблей первого типа при поиске по свойству conflict (P607) будет отображаться больше войн и сражений, чем у кораблей второго типа.

Запрос 12.4. Военные конфликты, в которых участвовали эсминец «Гремящий» и крейсер HMS Trinidad[6]

```
# List of military conflicts of the two ships
SELECT ?ship ?shipLabel ?conflict ?conflictLabel
WHERE
{
  VALUES ?ship {wd:Q4148613   # Soviet destroyer Gremyashchiy
                wd:Q1565575}  # United Kingdom's HMS Trinidad
  ?ship wdt:P607 ?conflict.   # conflict
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
}
```

Составим список отечественных кораблей, участвовавших в каких-либо военных конфликтах, с помощью запроса 12.5.

Найдите изображения кораблей, которые были сняты в кинофильмах или описаны в книгах.

См. ответ на с. 178

[6] Получено: 10 конфликтов у эсминца «Гремящий» (Q4148613) и один конфликт у крейсера HMS Trinidad (Q1565575), 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/vuo.

Запрос 12.5. Список отечественных кораблей, участвовавших в каких-либо военных конфликтах, использовано «угасающее» свойство “operator”[7]

```
1  # List of ship with countries and war conflicts
2  SELECT ?ship ?shipLabel ?countryLabel ?conflict ?conflictLabel WHERE
3  {
4    VALUES ?country {wd:Q34266 # Russian Empire
5                     wd:Q15180 # Soviet Union
6                     wd:Q159}  # Russia
7    ?ship wdt:P31 wd:Q11446;        # instance of ship
8          wdt:P137/wdt:P17 ?country;# belongs to operator
9          wdt:P607 ?conflict.       # engaged in some conflict
10   SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
11 }
```

Особенность результатов запроса 12.5 в том, что полученные корабли не обязательно связаны *только* с Российской империей, СССР или Россией. Например, корабль Kasato Maru (Q653477) — японский. Дело в том, что этот корабль принадлежал России в 1900–1905 годах, а Японии — с 1906 года.

В сноске к запросу 12.5 видно, что число воевавших кораблей, имеющих свойство *operator*, неуклонно снижается, корабли куда-то пропадают. Предложим читателям гипотезу, почему кораблей с каждым годом становится меньше.

В 2020 году было создано свойство *country of registry*. В этом свойстве у кораблей указывают страну, к которой приписано судно. С 2020 года редакторы Викиданных постепенно заполняют это свойство у кораблей, удаляя свойство *operator*. Идя в ногу со временем, мы заменим у кораблей свойство operator (P137) (строка 8 в запросе 12.5) на свойство country of registry (P8047) (строка 7 в запросе 12.6). И сразу число кораблей вырастет и с учётом новых данных станет даже больше того, что было в 2017 году.

Эти рассуждения позволяют сформулировать следующий *закон сохранения Викиданных*: уже добавленные объекты *массово* не исчезают. Новые свойства Викиданных дополняют или заменяют старые свойства. Нужно менять и уточнять старые SPARQL-запросы, чтобы число находимых объектов не уменьшалось.

Обратите внимание на утверждения в строках 10–11 запроса 12.6, объединённые командой `UNION`. Этим объединением мы смогли собрать и экземпляры объекта “ship”, и экземпляры объекта “ship type” в переменную `?ship`.

[7] Получено: 105 кораблей в 2017 году, 82 корабля в 2021 году, 8 кораблей в 2024 году. Ссылка на SPARQL-запрос: https://w.wiki/97Dq.

Запрос 12.6. Список отечественных кораблей, участвовавших в военных конфликтах, использовано новое свойство “country of registry”[8]

```
# List of Russian ships registered in countries and engaged in war conflicts
SELECT ?ship ?shipLabel ?countryLabel ?conflict ?conflictLabel WHERE
{
  VALUES ?country {wd:Q34266 # Russian Empire
                   wd:Q15180 # Soviet Union
                   wd:Q159}  # Russia
  ?ship wdt:P8047 ?country;  # country of registry of this ship
        wdt:P607 ?conflict.  # engaged in some conflict

  {?ship wdt:P31 wd:Q11446.} UNION     # instance of ship
  {?ship wdt:P31/wdt:P31 wd:Q2235308.} # instance of ship type

  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
}
```

[8] Получено: 122 корабля в 2024 году. Ссылка на SPARQL-запрос: https://w.wiki/97TG.

Корабли-музеи в странах мира

Корабль-музей (Q575727) — это корабль, на котором размещена музейная экспозиция, посвященная истории корабля. Такие корабли используются в общеобразовательных и мемориальных целях. Участие корабля в военном конфликте (Q180684) может послужить поводом для создания корабля-музея в память о прошедших событиях.

Чтобы понять большой и сложный запрос 12.7, рассмотрим сначала строки 21–32 этого запроса как отдельный скрипт, который можно запустить по ссылке: https://w.wiki/984K. В этом фрагменте запроса строятся связи между кораблём-музеем и страной, где расположен музей. На рис. 12.2 представлен фрагмент графа со страной, имеющей больше всего кораблей-музеев. Конечно, это Великобритания.

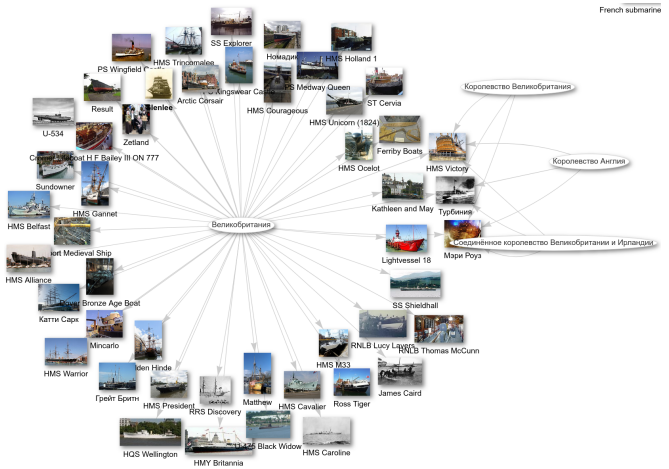


Рис. 12.2. Двойное кольцо кораблей-музеев Великобритании, фрагмент графа стран и кораблей-музеев, 2024 год

Запрос 12.7. Граф кораблей-музеев и стран, в которых они находятся (фрагмент скрипта)[9]

```
#defaultView:Graph
SELECT ?v1 ?v1Label ?v2 ?v2Label ?edgeLabel ?img
WHERE {
  {SELECT ?c ?cLabel ?v1 ?v1Label ?v2 ?v2Label (STR(COUNT(?c)) as ?edgeLabel)
   WHERE
   { VALUES ?cTypes
            {wd:Q180684 # conflict
             wd:Q831663 # military campaign
             wd:Q645883 # military operation
             wd:Q198    # war
            }
     ?c wdt:P31 ?cTypes.
     ?v1 wdt:P31 wd:Q6256. ?v2 wdt:P31 wd:Q6256. # country
     ?c wdt:P710 ?v1, ?v2. # in war
     FILTER (?v1 != ?v2 && STR(?v1) < STR(?v2))
     SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
  }
  GROUP BY ?c ?cLabel ?v1 ?v1Label ?v2 ?v2Label
  }
 UNION
  {SELECT DISTINCT ?v1 ?v1Label ?v2 ?v2Label ?img
   WHERE
   {
     ?v2 wdt:P31 wd:Q575727. # museum ship
     {?v2 p:P17 [ps:P17 ?v1]} UNION # ?v2 has country ?v1
     {
       ?v2 wdt:P131 ?loc.      # in ?loc
       ?loc p:P17 [ps:P17 ?v1].# ?loc in country ?v1
     }
     OPTIONAL {?v2 wdt:P18 ?img}
     SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en"}
   }
 }
}
```

[9] Получено: 117 вершин графа в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/wz8.

Рассмотрим теперь запрос 12.7 в целом. Построим граф кораблей-музеев и воевавших стран (рис. 12.3 и рис. 12.4). Вершинами графа будут страны (Q6256) и корабли-музеи (Q575727). Ребро между кораблём и страной означает, что корабль-музей находится в этой стране. А ребро между двумя странами означает, что эти страны участвовали в одних и тех же конфликтах, число которых равно весу ребра (вес подписан на рёбрах, соединяющих страны). Итак, запрос 12.7 строит граф по описанным выше правилам.

При определении *страны* для переменных $v1$ и $v2$ посредством свойства государство (P17) в строках 25 и 28 запроса 12.7 используется конструкция `p:/ps:` (точнее: `{?v2 p:P17 [ps:P17 ?v1]}`). Это нужно потому, что у стран в поле экземпляр (P31), кроме значения «страна», могут быть и другие значения. Эта конструкция позволяет перебрать все значения поля P31 и найти значение «страна». См. подробности в главе 8 на с. 83.

Функция `FILTER` в строке 15 запроса 12.7 (`FILTER (?v1 != ?v2 && STR(?v1) < STR(?v2))`) позволяет избежать появления петель и дублирующих рёбер на графе.

Из фрагмента графа на рис. 12.4 видно, что корабли-музеи есть у России и США. Эти страны участвовали в военных конфликтах и они имеют флот, а значит, и корабли для создания музеев.

При просмотре полного графа (и даже на фрагменте, рис. 12.4) видна уникальность научно-исследовательского судна «Витязь» (Q1516653). По Викиданным (запрос 12.7), это единственный музей-корабль России, связывающий её с другими странами. Изначально этот теплоход был построен и находился на службе в Германии, что отражено в свойстве «Витязя» operator (P137)=“Кригсмарине” (это название военно-морских сил в Третьем рейхе).

Вершины графа на рис. 12.4 кликабельны. Например, можно кликнуть на крейсер «Аврора» (Q168713), тогда появятся дополнительные вершины, соответствующие свойствам этого крейсера (рис. 12.3).

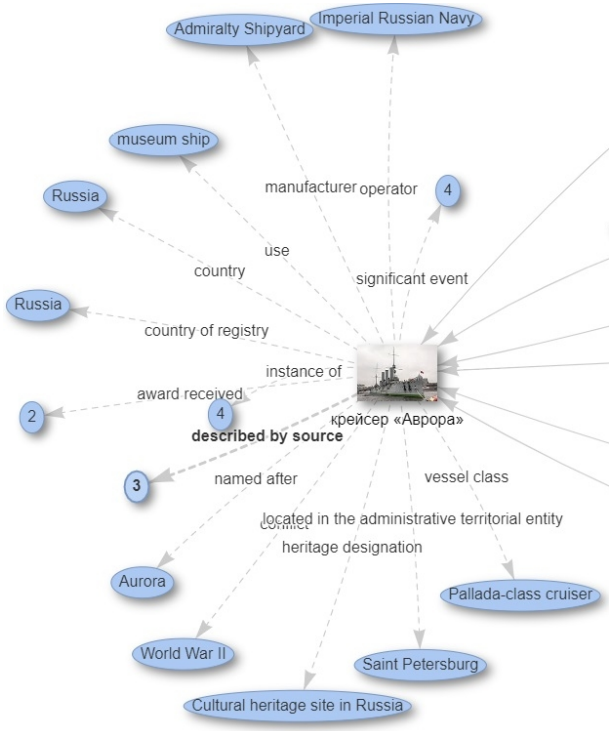


Рис. 12.3. Граф свойств крейсера «Аврора», 2021 год

Упражнения

1. Измените запрос 12.7 так, чтобы граф не содержал вершин-стран без кораблей-музеев.

2. Измените запрос 12.7 так, чтобы граф содержал только те страны, которые участвовали больше чем в одном конфликте.

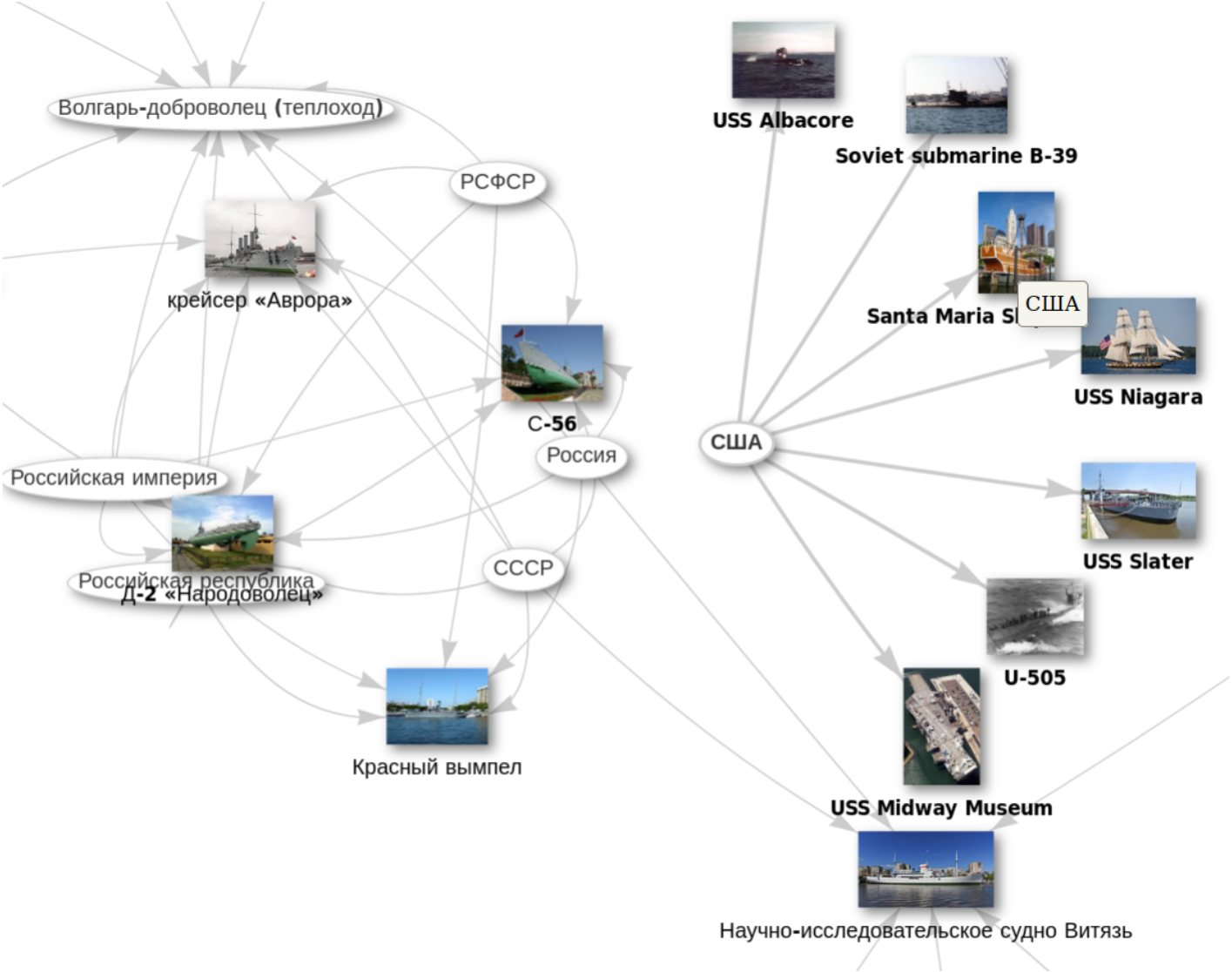


Рис. 12.4. Фрагмент графа стран, участвовавших в войнах, и кораблей-музеев России и США, 2021 год

13. Космические корабли и станции: современные реалии 50-летней давности

Глава посвящена исследованию космических кораблей и станций на основе Викиданных. С помощью SPARQL-скриптов построен список отечественных кораблей и станций, нарисованы временные графики запуска кораблей в нашей стране и в мире с 1960 по 2025 год. Выполнена оценка полноты Викиданных, показавшая, что многие объекты имеют неправильное значение свойства частный случай понятия (P31)[1]. В ходе работы было обнаружено, что текущие показатели росийской космонавтики по количеству запусков космических кораблей за последние годы соответствуют показателям в СССР пятьдесят лет назад.

Список космических кораблей и станций

Построим запрос 13.1 для вывода списка всех космических кораблей и станций. Нам потребуются объекты космическая станция (Q25956), космический корабль (Q40218) и отношение экземпляр (P31).

Запрос 13.1. Список кораблей и станций[2]

```
# List of spacecraft (Q40218) and space station (Q25956)
SELECT  ?s ?sLabel ?typeLabel WHERE {
  VALUES ?type {wd:Q40218 wd:Q25956}
  ?s wdt:P31 ?type.  # Selecting the type of object
  SERVICE wikibase:label {bd:serviceParam wikibase:language"ru,en"}
}
```

В какой стране спроектирован этот аппарат?

См. ответ на с. 180

[1] «Частный случай понятия», или «экземпляр» (англ. *instance of*), — это конкретный объект класса, категории, одно из основных свойств (отношений), используемых в Викиданных и позволяющих классифицировать объекты, связывая их с разными «классами», «категориями».

[2] Получено: 145 объектов в 2021 году. SPARQL-запрос: `https://w.wiki/4d4f`.

Глубина проработки объектов

Сервис ProWD показал, что заполнение космических объектов неравномерное, большая часть заполнена менее чем на 30 %. По состоянию на 2021 год наиболее полным и проработанным в Викиданных является космический корабль «Аполлон-8» (Q184201), имеющий 30 свойств. При этом всего по одному свойству имеют такие корабли, как:

- Europa Astrobiology Lander (Q10491365),
- Project Orbiter (Q6514453),
- LRK (Q5961734),
- EarthForce One (Q5327028)[3],
- «Союз ГВК» (Q60767924),
- CubeSat for Solar Particles (Q22907583).

Список отечественных кораблей и станций

Найдём корабли и станции, сконструированные в СССР или в России, с помощью запроса 13.2.

Запрос 13.2. Список отечественных кораблей[4]

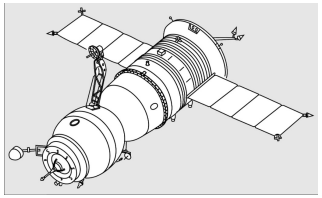
```
# List of Russian and USSR spacecrafts and stations
SELECT  ?spacecraft  ?spacecraftLabel WHERE
{
  {?spacecraft wdt:P31 wd:Q40218.} UNION #spacecraft
  {?spacecraft wdt:P31 wd:Q25956.} # and space station

  # Soviet Union and Russia
  VALUES ?ruCountries {wd:Q15180 wd:Q159}
  ?spacecraft wdt:P17 ?ruCountries. # related to Russian countries

  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
```

[3] Отметим непростую судьбу объекта EarthForce One (Q5327028). Этот объект был создан в Викиданных ботом автоматически в 2013 году, поскольку существовала статья в Английской Википедии EarthForce One. Статья была удалена из-за отсутствия надёжных источников, доказывающих значимость объекта, а в Викиданных объект остался как неприкаянный. Подумайте над запросом, который вывел бы список аналогичных объектов Викиданных, не соответствующих ни одной статье Википедии.

В какой стране спроектирован этот аппарат?



См. ответ на с. 180

[4] Получено: 3, 25 и 16 отечественных кораблей и станций в 2017, 2021 и 2024 годах соответственно. Ссылка на SPARQL-запрос: https://w.wiki/4d9c.

Полнота Викиданных по числу отечественных космических кораблей

Проанализируем степень заполнения Викиданных в области отечественных космических кораблей и станций. Если по запросу 13.1 было получено 145 кораблей и станций во всём мире, то по запросу 13.2 — всего 25 отечественных объектов. Информации о советских и российских объектах в 2021 году стало в несколько раз больше по сравнению с 2017 годом, когда по запросу 13.2 выдавалось всего 3 объекта.

На сайте www.buran.ru в разделе о космических кораблях СССР и России на 2021 год содержится информация о 36 кораблях. В советской энциклопедии «Космонавтика» от 1985 года упоминаются 94 космических корабля и орбитальные станции СССР и США, запущенные с 1961 по 1983 год, из них 51 корабль принадлежит СССР[5]. Это означает, что в Викиданных представлено менее половины советских космических кораблей. В американском справочнике по космонавтике и астрономии приведены 10 советских и российских запусков космических станций, 126 запусков шаттла НАСА с 1981 по 2008 год и характеристики 31 ракеты-носителя[6].

Временные графики освоения космоса в нашей стране и в мире

Временной (с 1960-х годов) график запуска космических аппаратов в нашей стране (рис. 13.1) построен с помощью запроса 13.3.

Ранее в запросах для получения каких-либо списков мы использовали свойство экземпляр (P31). В запросе 13.3 мы обошлись без этого свойства за счёт использования отношения «дата запуска космического корабля» (P619) в строке 10 для обхода и подсчёта таких объектов, которые были запущены в космос.

Логический оператор `UNION` в строках 5–8 позволяет объединить советские и российские запуски космолётов.

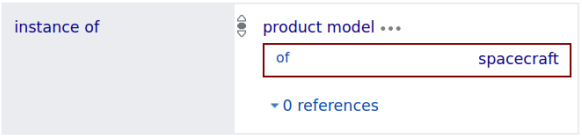
Если бы в запросе 13.3 вместо переменной `?lapse` оставалась переменная `?year` (строки 3 и 12), то мы получили бы на рис. 13.1 число запусков за каждый год. Благодаря функции округления `FLOOR()` и оператору `GROUP` выполняется группировка[7] кораблей, запущенных в течение пяти лет. При этом в переменную `?quantity` записывается число этих объектов за пятилетку, подсчитанное с помощью функции `COUNT()`.

Для представления результатов в виде столбчатой диаграммы используется стиль отображения BarChart (см. строку 2 запроса 13.3).

[5] *Глушко В. П.* Космонавтика : энциклопедия. Москва : Сов. энциклопедия, 1985. С. 498.

[6] *Joseph A., Angelo J.* The facts on file space and astronomy handbook. Facts on file, 2009.

В предыдущих запросах мы искали экземпляры космических кораблей, то есть искали в Викиданных утверждения *is instance of spacecraft*. На рисунке ниже показан вариант утверждения о космических кораблях, входящих в какую-либо линейку, а именно: *is instance of product model of spacecraft*. Когда есть линейка моделей, в нашем случае — серия космических кораблей, тогда нужен объект product model (Q10929058). На этом рисунке *of* и *spacecraft* — это соответственно свойство и значение квалификатора, характеризующего объект *product model* (см. справку Викиданных Help:Квалификаторы). Найдите серии космических кораблей с использованием квалификаторов.



См. ответ на с. 180

[7] В строке 15 запроса 13.3 запущенные в космос объекты группируются командой: `GROUP BY ?lapse`. То, что группировка идёт именно по пятилеткам, а не, скажем, шестилеткам, определяется в строке 12, где переменная `?year` делится на 5, округляется, умножается на 5.

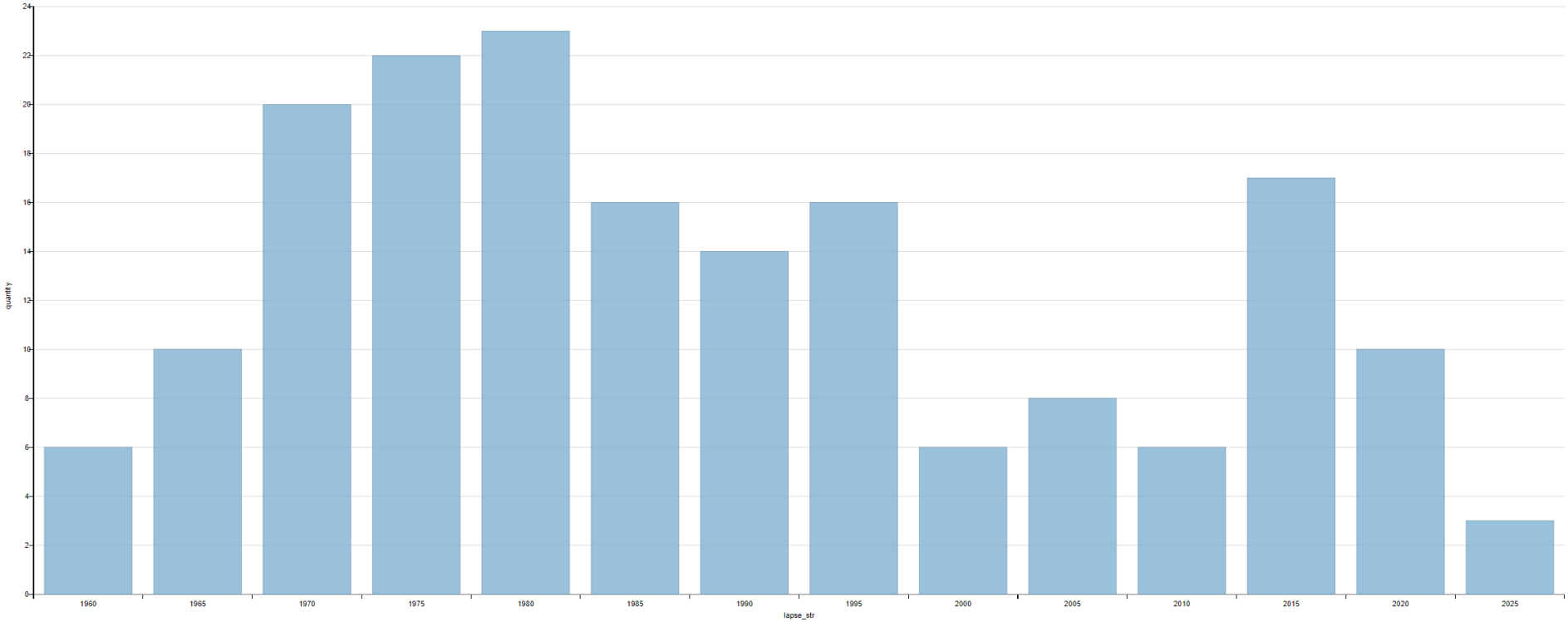


Рис. 13.1. Диаграмма количества запусков космических кораблей в СССР и в России по пятилеткам, 2022 год

Горизонтальной оси́ на рис. 13.1 отвечает переменая `?lapse_str` в запросе 13.3. Если не преобразовывать число `?lapse` в текстовую переменную `?lapse_str`[8], то ось X имеет диапазон от 0 до 2200 вместо требуемого диапазона от 1960 до 2025 года, а результаты обозначаются точками с координатами (пятилетка, число запусков), в итоге график становится нечитаемым.

На рис. 13.1 видно, что самый активный период развития отечественной космонавтики был в 1970–1995 годах.

[8] Преобразование числа в текст в строке 3 запроса 13.3: `(STR(?lapse) AS ?lapse_str)`.

Запрос 13.3. Запуски космических кораблей в СССР и в России[9]

```
# The number of spacecraft launches in Russia every 5 years
#defaultView:BarChart
SELECT (STR(?lapse) AS ?lapse_str) (COUNT(?item) AS ?quantity)
WHERE {                                  # spacecraft belongs to
        {?item wdt:P17 wd:Q15180}             # country = USSR
  UNION {?item wdt:P17 wd:Q159}               # country = Russia
  UNION {?item wdt:P495 wd:Q159}    # country of origin = Russia
  UNION {?item wdt:P495 wd:Q15180}. # country of origin = USSR

  ?item wdt:P619 ?launch.           # date of spacecraft launch
  BIND( YEAR(?launch) AS ?year)
  BIND(FLOOR(?year/5)*5 AS ?lapse)  # count for each 5 years
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
GROUP BY ?lapse
ORDER BY ?lapse # Order 1970, 1975, 1980, ...
```

Сколько кораблей было запущено в СССР в 1960-е, 1970-е и 1980-е годы? См. ответ на с. 181.

[9] Получено: 14 пятилеток. Ссылка на SPARQL-запрос: https://w.wiki/4eGg.

Запрос 13.4. Запуски космических кораблей в мире по годам и странам[10]

```
#defaultView:BarChart
SELECT ?year (COUNT(?obj) AS ?count) ?country ?countryLabel WHERE {
  ?obj wdt:P17 ?country. # spacecraft belongs to country
  ?obj wdt:P619 ?launch. # date of spacecraft launch
  BIND(str(YEAR(?launch)) AS ?year)
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
GROUP BY ?year ?country ?countryLabel
```

Запрос 13.4 строит график запусков космических кораблей в мире по годам и странам. Рис. 13.2 показывает, что больше всего космических аппаратов запускали Индия и США (только у них зафиксировано более 10 ежегодных запусков) в 2017–2018 годах. Пик запусков в мире был в 2018 году (59 запусков).

[10] Получено: 328 результатов в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4bEu.

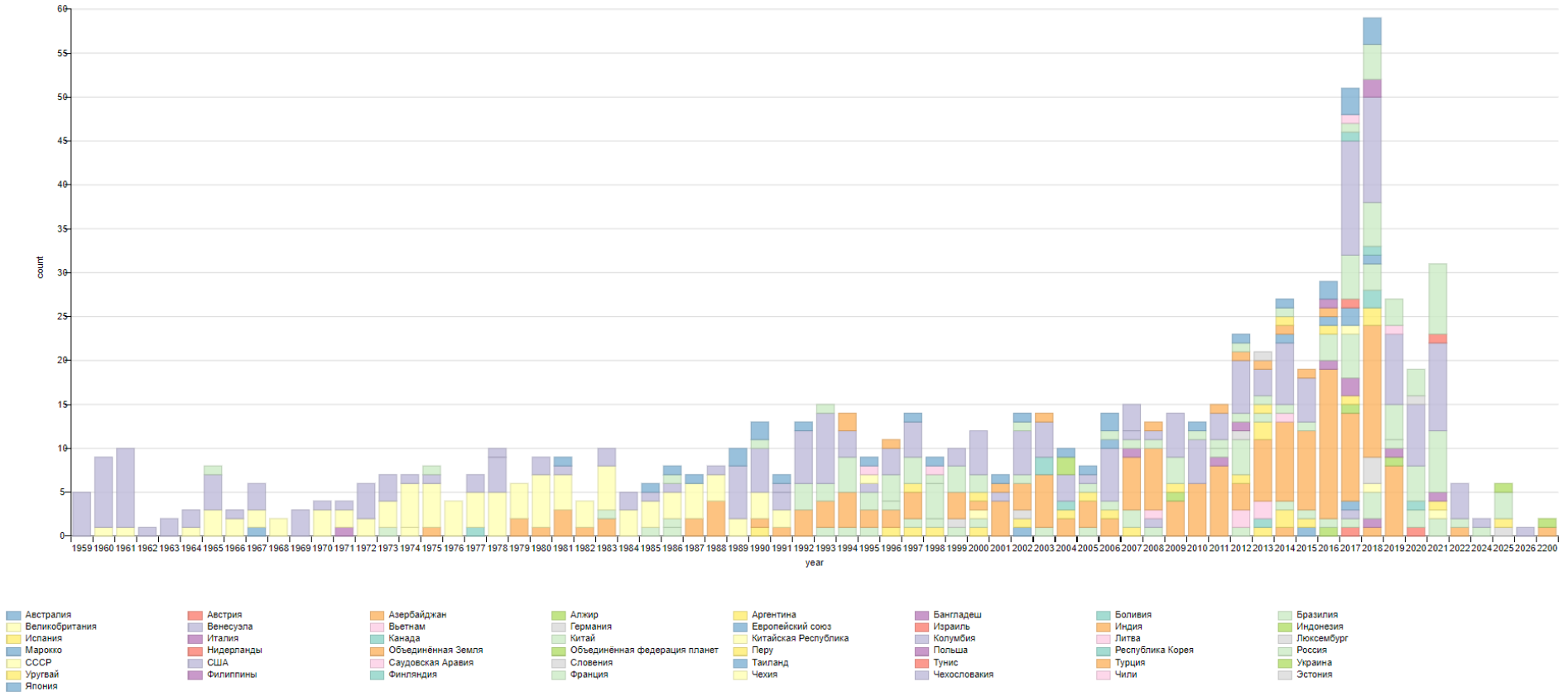


Рис. 13.2. Диаграмма ежегодного суммарного количества запусков космических кораблей по странам, 2021 год

По Викиданным, российская космонавтика занимает средние позиции по количеству запусков, её численные показатели за 2016–2019 годы схожи с показателями СССР в 1970-е и 1980-е годы и составляют 3–5 запусков в год.

Космонавты в международных полётах

Запрос 13.5 строит граф (рис. 13.3) с вершинами «ракеты» и «космонавты» с раскраской по странам.

Для работы скрипта необходимо указать начальную точку — космонавта, который принимал участие в международных космических полётах. Начальная точка задаётся в строке 11 скрипта и записывается в переменную `?naut_seed`. Далее, в строке 12, выполняется поиск астронавтов, летавших совместно с тем, кого мы указали ранее. В строках 15–17 подгружаются данные о космических аппаратах, полётах и астронавтах. Булева переменная `?toggle` имеет значение `?true`, если найденный объект является космическим аппаратом, или `?false`, если найденный объект является космонавтом. В переменную `?item` в строке 19 записывается выбранный объект (космический аппарат или космонавт). В строке 20 в переменную `?itemLabel` записывается название объекта, в строке 21 в переменную `?rgb_source` записываются данные для раскраски вершин графа, а именно: белый цвет (FFFFFF) для космических кораблей, цвет «зелёный шартрез» (7FFF00) для вершин-космонавтов.

Если выбран корабль, то в строке 22 в переменную `?link` ничего не записывается; если выбран космонавт, то в переменную `?link` записываем его корабль. На графе (рис. 13.3) этой переменной будет соответствовать дуга от космонавта к кораблю.

В строке 23 подгружаются данные о гражданстве космонавта, а в строках 26–34 вершинам космонавтов на графе присваивается цвет в зависимости от их гражданства (рис. 13.3).

Упражнения

1. Постройте список кораблей, которые отправились или отправятся на Марс (Q111).
2. Подсчитайте долю кораблей (нарисуйте график по десятилетиям), отправленных на Марс, по отношению к числу кораблей, отправленных на Луну (Q405).
3. Подсчитайте количество успешных (Q7632586) космических запусков относительно числа неудачных (Q1121708) запусков[11].

Какое наибольшее и наименьшее количество запусков космических кораблей за десятилетие совершило человечество с 1970 по 2010 год?

См. ответ на с. 182.

Если для космонавтов мы выбрали цвет ликёра, то почему на рис. 13.3 не все космонавты зелёные? Вопрос на внимательность.

[11] Например, у объекта «космическая программа Луна» (Q192372) в свойстве «ключевое событие» (P793) указано число успешных и неудачных запусков.

Запрос 13.5. Космонавты в международных полётах[12]

```
# Graph of astronauts as crew of flights of different countries
#defaultView:Graph
SELECT DISTINCT ?item ?itemLabel ?rgb ?link ?naut_seed
WHERE
{
  VALUES ?toggle { true false }
  # Let's select a subset of astronauts
  {
    SELECT DISTINCT ?naut WHERE
    {
      VALUES ?naut_seed {wd:Q313815}.  # Sergei Krikalev
      ?s wdt:P1029 ?naut_seed, ?naut;
    }       # ?naut_seed and ?naut are member of a same crew
  }
  ?s  wdt:P31/wdt:P279* wd:Q40218; # spacecraft and subclasses
      wdt:P31/wdt:P279* wd:Q752783;# human spaceflight and subclasses
          wdt:P1029 ?naut;         # has member of the crew ?naut
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
  BIND(IF(?toggle,?s,?naut) AS ?item).
  BIND(IF(?toggle,?sLabel,?nautLabel) AS ?itemLabel).
  BIND(IF(?toggle,"FFFFFF","7FFF00") AS ?rgb_source).
  BIND(IF(?toggle,"",?s) AS ?link).
  ?naut wdt:P27 ?country. # astronaut is citizen of country
  # ?toggle = true then spacecraft node
  # ?toggle = false then astronaut node
  BIND(             # Soviet and Russian astronauts have red nodes
    IF(!?toggle && (?country=wd:Q15180||?country=wd:Q159),"FF0000",
    IF(!?toggle && ?country=wd:Q30,"FF00FF",  # USA - fuchsia
    IF(!?toggle && ?country=wd:Q183,"C0C0C0", # Germany - silver
    IF(!?toggle && ?country=wd:Q142,"008080", # France - teal
    IF(!?toggle && ?country=wd:Q40,"800000", # Austria - maroon
    IF(!?toggle && ?country=wd:Q38,"00FFFF", # Italy - aqua
    ?rgb_source))))))
    AS ?rgb).
}
```

[12] Получено: 68 результатов в 2022 году. Ссылка на SPARQL-запрос: https://clck.ru/agioQ.

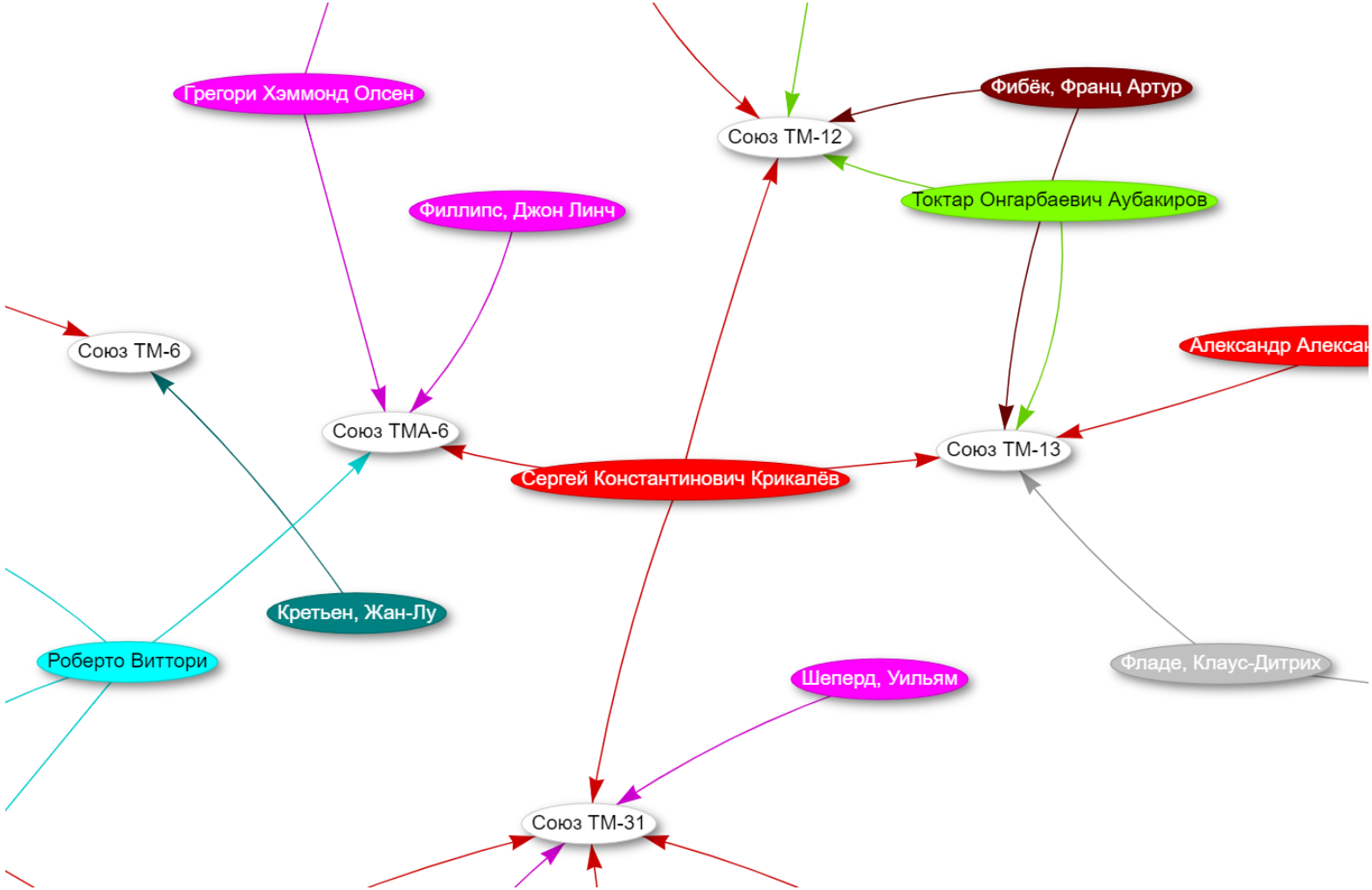


Рис. 13.3. Фрагмент графа с вершинами типа «ракеты» и «космонавты»: красный цвет вершин — космонавты СССР и России, розовый — США, серый — Германии, бирюзовый — Франции, бордовый — Австралии, голубой — Италии

Часть III

Специальные возможности вики-проектов и Викиданных

14. Сервис балансировки ProWD

С помощью сервиса ProWD можно анализировать объекты Викиданных. Возьмём, например, объект «архив» (Q166118). Мы можем увидеть:

- какой архив мира наиболее проработан по числу свойств;
- значение коэффициента Джини, показывающего, насколько равномерно по числу свойств заполнены экземпляры объекта «архив». На полях приведены два рисунка, полученные с помощью сервиса ProWD. На рис. 14.1 показана высокая степень равномерности заполнения свойств у стран на Викиданных. Это можно объяснить важностью объектов-стран и тем, что сотни пользователей многих википедий их редактируют. На рис. 14.2 видно, что у объектов «языки программирования» число свойств распределено неравномерно. Есть небольшое число языков, у которых заполнено максимальное число свойств, но у подавляющего большинства языков-объектов заполнено мало свойств.

Сервис ProWD позволяет сравнивать подклассы объекта по количеству свойств. Например, в статье Элизабет Гиземанн[1] с помощью сервиса ProWD сравнивают мужчин и женщин, специалистов в области информатики.

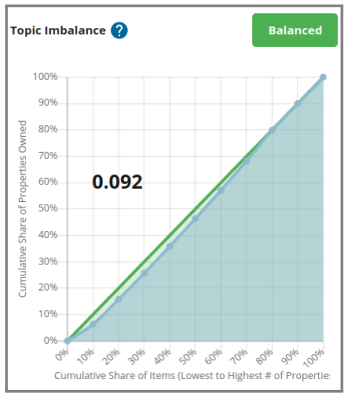


Рис. 14.1. Высокая степень равномерности заполнения по числу свойств объекта Викиданных страна (Q6256), 2020 год. Коэффициент Джини равен 0.092

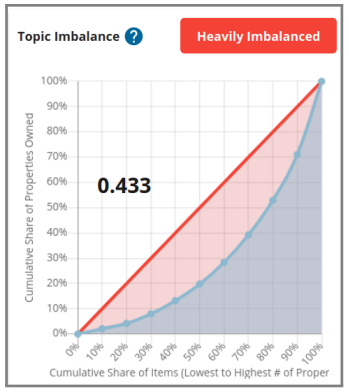


Рис. 14.2. Низкая степень равномерности заполнения по числу свойств объекта язык программирования (Q9143), 2020 год. Коэффициент Джини равен 0.433

[1] *Giesemann E.* ProWD: Detecting Knowledge Imbalances on Wikidata. 2020. URL: https://blog.wikimedia.de/2020/09/16/prowd-detecting-knowledge-imbalances-on-wikidata/.

15. Защита страниц

На страницы Викиданных устанавливается защита для предотвращения повторяющегося вандализма или спама. Существует несколько видов защиты:

- частичная защита или полузащита (обозначается серым замком) разрешает редактировать страницу только автоподтверждённым/подтверждённым участникам;
- полная защита (обозначается оранжевым или красным замком) ограничивает круг редакторов администраторами;
- защита от переименования (обозначается зелёным замком) не ограничивает возможность редактировать страницу, однако переименовать её могут только администраторы. Большинство популярных страниц защищено от переименования. Защита от переименования не может быть применена к страницам элементов или свойств;
- защита от создания (как полная, так и частичная защита обозначается синим замком) может применяться к удалённым или несуществующим страницам. Однако, как и защита от переименования, она не может применяться к удалённым элементам или свойствам.
  - При полной защите от создания страницу не может создать никто, кроме администраторов.
  - При частичной защите от создания страницу могут создать также автоподтверждённые и подтверждённые участники.

Рис. 15.1. Частичная защита или полузащита

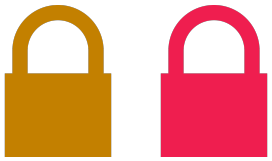


Рис. 15.2. Полная защита

Рис. 15.3. Защита от переименования

Рис. 15.4. Защита от создания

В крайне редких случаях Фонд Викимедиа может защитить страницу в качестве официального действия (*office action*, обозначается чёрным замком). Официальные действия совершаются только в результате формальной вневикипедийной жалобы, всегда публично объявляются и выполняются только сотрудниками Фонда Викимедиа или членами Совета попечителей.

Дополнительные материалы о защите страниц см. на странице Wikidata:Protection policy на сайте Викиданных.

Рис. 15.5. Официальное действие

16. Боты в Викиданных

В этой главе рассматривается автоматизация процессов в Викиданных. Часто нужно исправить повторяющиеся ошибки или ввести большие объёмы данных вместо того, чтобы изменять свойства по одному. Для ввода данных в нашем распоряжении есть несколько инструментов, облегчающих работу, таких как OpenRefine[1] или QuickStatements[2], но регулярно повторяющиеся добавления и вставки должны выполняться с помощью бота.

Требования

Ботов программируют на разных языках, мы для упрощения задачи воспользуемся библиотекой Pywikibot, написанной на языке Python и предназначенной для облегчения доступа к информации в проектах Викимедиа. Есть три варианта запуска наших программ:

1) использовать веб-оболочку наподобие PAWS[3];
2) установить облачную инфраструктуру наподобие Toolforge[4];
3) запустить на своём собственном компьютере.

Благодаря наличию документации[5] можно воспользоваться любым из этих трёх способов. Для работы с ботами со своего компьютера нужно предварительно установить среду и язык программирования Python. После установки мы можем проверить, корректно ли она установлена, набрав

```
python --version
```

в командной строке Windows или в консоли Linux (в дальнейшем всё делаем в консоли), и тогда в консоли мы увидим версию Python, с которой будем работать. Затем устанавливаем и настраиваем Pywikibot.

[1] OpenRefine — программа для обработки табличных данных. С 2021 года доступна редакторам Википедии в виде онлайн-сервиса (https://openrefine.org).

[2] QuickStatements — сервис пакетной обработки Викиданных (https://quickstatements.toolforge.org).

[3] PAWS (a web shell) — веб-оболочка и сервис, включает блокноты Jupyter для использования всеми участниками Викимедиа.

[4] Toolforge — платформа энтузиастов-программистов, разрабатывающих сервисы и программы для обработки данных проектов Викимедиа (https://wikitech.wikimedia.org/wiki/Portal:Toolforge).

[5] См. руководство по установке Pywikibot: https://www.mediawiki.org/wiki/Manual:Pywikibot/Installation/ru.

Эта настройка Pywikibot включает несколько этапов:

- Создайте файл конфигурации: `py pwb.py generate_user_files`.
- Выполните команду входа в систему: `py pwb.py login`.
- Просмотрите файлы `user-config.py` и `user-password.py`, где мы можем указать имена пользователей, которые будут вносить изменения.

Для просмотра информации или выполнения небольших тестов мы можем воспользоваться нашим именем пользователя[6], но в случае внесения большого количества изменений в Викиданные рекомендуется создать учетную запись бота и запросить разрешение на его запуск. Чтобы получить этот флаг, обычно требуются некоторые знания о редактировании в проектах Викимедиа, подтверждающие, что мы не сделаем серьёзных ошибок и у нас есть поддержка других пользователей.

Обычно для получения этого флага нас просят выполнить определённое количество тестовых правок, чтобы убедиться, что такие правки полезны, а ошибок практически нет. Мы также можем продолжать выполнять задачи через нашу учётную запись, но с небольшой частотой правок.

[6] Имя пользователя (логин) — это то имя, под которым мы зарегистрировались в проектах Викимедиа. В этих проектах (Википедия, Викиданные, Викиверситет и др.) используется единая пара логин/пароль.

Наши первые скрипты

После того как мы правильно настроили Pywikibot, мы можем запустить нашу первую программу (листинг 16.1). Перейдём в папку, где находится программный код Pywikibot, и создадим новый файл с именем `test.py`.

Мы можем вносить изменения в файл при помощи любых текстовых редакторов или редакторов кода, таких как Visual Studio Code, Notepad++, или обычного редактора Notepad. Открыв файл, добавим в него следующий код (листинг 16.1).

Листинг 16.1. Получение содержимого страницы Викиданных о реке Обнице в Польше

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q16583338")      # river in Poland
print(page.get())
```

Сохраним и запустим этот файл, набрав следующую инструкцию: `py pwb.py test.py`.

Отметим, что в Викиданных есть две реки с одинаковым названием Обница — одна в Польше, другая в Сербии. Одноимённые объекты легко различимы в Викиданных по номерам: Obnitsa (Q16583338) и Obnitsa (Q959190).

Если всё работает корректно, мы увидим в консоли весь текст, написанный на странице реки Обница (Q16583338) в Викиданных. В первой строке программы 16.1 мы импортировали библиотеку `Pywikibot`. Во второй строке указали название проекта Викимедиа, над которым хотим работать, и языковую версию проекта Викимедиа или особо — Викисклад[7]. В третьей строке указываем искомый объект и, наконец, указываем, что хотим получить весь контент.

Page (страница) — это модуль, включающий множество полезных методов[8]. В дополнение к методу `get()` в нашем распоряжении есть возможность узнать заголовок элемента с помощью метода `title()` (листинг 16.2).

Листинг 16.2. Получение заголовка объекта с номером Q16583338

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q16583338")      # river in Poland
print(page.title())
```

С помощью метода `isRedirectPage()` мы можем узнать, является ли этот объект страницей-перенаправлением. Для этого в листинге 16.3 укажем объект, соответствующий странице-перенаправлению (Q16583333 → Q2149254)[9].

Листинг 16.3. Получение ответа на вопрос, является ли элемент страницей-перенаправлением (истина или ложь)

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q16583333")
print(page.title())
print(page.isRedirectPage())
```

Обратите внимание, что до сих пор упражнения были очень простыми, но по мере усложнения мы должны быть осторожны с отступами, которые позволяют указать в языке Python, какая информация входит в структуры управления.

Предыдущее упражнение (листинг 16.3) возвращает только истину или ложь. Такой запрос нужен, чтобы узнать, является ли заданная страница перенаправлением, и если «да», то предупредить об этом пользователя (листинг 16.4).

[7] Например, Русская Википедия (код ru) и Английская Википедия (код en) — это языковые версии Википедии. А у Викисклада (его также называют Commons) нет языковых версий и есть только один код: commons.

[8] См. дополнительную информацию о модуле Page: https://doc.wikimedia.org/pywikibot/master/api_ref/pywikibot.html#pywikibot.Page.

[9] См. как выглядит страница-перенаправление: https://www.wikidata.org/w/index.php?title=Q16583333&redirect=no.

Листинг 16.4. Предупреждение пользователя о том, что объект является страницей-перенаправлением

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q16583333")     # redirect page
if (page.isRedirectPage()):
  print("Обратите внимание, это перенаправление, а не объект")
```

Но что произойдёт, если мы проанализируем несколько объектов и встретим несуществующий? Возможно, что программа вернёт ошибки, потому что не сможет проанализировать содержимое объекта. Мы можем это предусмотреть и отловить ошибки в Python, то есть проверить, существует ли объект (листинг 16.5).

Листинг 16.5. Проверка существования объекта

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q107043778")     # non-existent object
if not page.exists():
  print("К сожалению, этот объект не существует!")
```

Измените программу 16.5 так, чтобы отлавливать возможные ошибки с помощью исключений (конструкция try — except).

Посмотрим с помощью метода `properties()` в листинге 16.6, какие свойства содержит наш объект.

Листинг 16.6. Получение списка свойств объекта

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q16583338")     # river in Poland
print(page.properties())
```

При выполнении программы 16.6 мы видим, что метод `properties` возвращает словарь Python, то есть набор свойств и соответствующих им значений в этом объекте. Например, свойство `page_image_free` возвращает изображение, видимое на странице объекта, свойство `wb_claims` — количество утверждений (свойств), а свойство `wb_sitelinks` — количество ссылок, связывающих этот объект с проектами Викимедиа.

Метод `contributors()` также возвращает словарь, содержащий пользователей, редактировавших этот объект Викиданных. Он покажет нам имена (ники) этих редакторов, а также число правок каждого из них. Ещё один метод `revision_count()` возвращает общее количество правок в объекте. Оба этих метода задействованы в листинге 16.7.

Листинг 16.7. Перечисление редакторов объекта Викиданных и числа правок

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q16583338")      # river in Poland
print(page.contributors())
print(page.revision_count())
```

Существуют и другие методы, которые возвращают более сложную информацию, чем словарь данных. Например, при обращении к методу `linkedPages` (связанные страницы) мы получим объект. Если мы просто обратимся к этому методу, как в листинге 16.8, и выведем результат на печать с помощью функции `print()`, то получим строчку: `<pywikibot.data.api.PageGenerator object at 0x000002C6C7DB2FD0>`. Эта строка сообщает, что получен объект типа `PageGenerator`. Прочитать данные этого объекта мы можем в цикле `for` (листинг 16.8).

Листинг 16.8. Чтение объекта типа `PageGenerator`

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q16583338")      # river in Poland
for linked in page.linkedPages():
  print(linked)
```

Итак, с помощью метода `page.linkedPages()` мы получаем список тех ссылок, которые перечислены на исследуемой странице `page`. Сначала будут показаны объекты, а затем свойства (включая информацию, содержащуюся в ссылках).

Чтобы узнать количество объектов, ссылающихся на заданный объект, воспользуемся методом `backlinks()`, возвращающим массив объектов. Так же как в предыдущем случае (листинг 16.8), обойдём этот список страниц-объектов и напечатаем их (листинг 16.9). Существуют страницы-сироты[10], на которые никто не ссылается. Для таких объектов метод `backlinks()` вернёт пустой список. Подберём такой объект в Викиданных (например, Pisueña (Q6980876)), на который есть ссылки, чтобы было что выводить на печать в программе 16.9.

[10] О статьях-сиротах см. руководство на странице Википедия:Статьи-сироты. С этими статьями связан внутренний проект Википедии по повышению связности: `https://w.wiki/4cMm`.

Листинг 16.9. Использование метода `backlinks()` для получения объектов, ссылающихся на заданный объект

```
import pywikibot
site=pywikibot.Site('wikidata', "wikidata")
page=pywikibot.Page(site, u"Q6980876")      # river in Spain
for links in page.backlinks():
  print(links)
```

Итак, мы выбрали испанскую реку Pisueña (Q6980876), на которую ссылаются два других объекта Викиданных: это река Пас (Q14360) и безымянная страница-перенаправление (Q23993869). И мы знаем число ссылок на реку Pisueña.

Кроме модуля `Page`, библиотека Pywikibot даёт нам множество других инструментов. Отметим, что работа только с одним объектом малополезна. Обычно в одном скрипте анализируют и меняют сразу несколько объектов. Для этого нам пригодятся все те запросы, которые мы написали в этой главе.

Напишите свою программу, творчески объединив листинги 16.1–16.9 для получения данных о каком-либо выбранном вами объекте Викиданных.

SPARQL-запросы внутри программы на Python

Сначала откроем страницу сервиса WDQS[11] и напишем свой запрос на языке SPARQL. Викиданные заполняются в основном вручную, поэтому неизбежны ошибки. Попробуем найти с помощью SPARQL-запроса такую ошибку. Например, часто у людей (соответствующих им объектов на Викиданных) вместо свойства гражданство (P27) ошибочно указывают свойство страна (P17). Запрос для поиска такой ошибки может выглядеть так (листинг 16.10):

Листинг 16.10. Запрос для получения персон с ошибочным отнесением к стране вместо указания гражданства[12]

```
SELECT ?item ?itemLabel WHERE
{
  ?item wdt:P31 wd:Q5;  # instance of human
        wdt:P17 wd:Q36. # has country Poland
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
```

[11] См. пояснения о службе WDQS на с. 16.

[12] Получено: 3 профессии в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4cBg.

Мы должны проверить, возвращает ли запрос 16.10 какое-либо значение, поскольку может оказаться, что ошибок нет. В этом случае мы можем изменить страну. Например, можем указать следующую непроверенную страну Литву (Q37) вместо проверенной Польши (Q36), поскольку наверняка будут другие страны с такой же ошибкой.

После того как мы построили SPARQL-запрос 16.10 для получения объектов-людей с ошибками, мы можем добавить запрос в наш листинг 16.11 на языке Python.

Листинг 16.11. Идентификация персон с ошибкой страна — гражданство на примере Польши (Q36) (включение SPARQL-запроса в программу Python)

```
import pywikibot
from pywikibot import pagegenerators
site=pywikibot.Site('wikidata', "wikidata")
query = """
  SELECT DISTINCT ?item WHERE {
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
    ?item wdt:P31 wd:Q5.
    ?item wdt:P17 wd:Q36.
  } """

pages = pagegenerators.WikidataSPARQLPageGenerator(query,site=site)
for item in pages:
  print(item.title())
```

Из предыдущих упражнений мы взяли только модуль `pagegenerators`, который позволяет нам получить список объектов с помощью фильтра в виде SPARQL-запроса (листинг 16.10). Мы сохранили текст запроса в переменной `query` (листинг 16.11), а затем передали её в метод генераторов страниц `WikidataSPARQLPageGenerator()`. Данные, возвращаемые этим методом, были обработаны с помощью цикла `for`, и мы напечатали названия объектов. Таким образом, эта программа (листинг 16.11) печатает список имён людей, у которых в свойстве «страна» указана Польша.

Мы уже научились получать несколько элементов сразу. Займёмся этим. Вместо того чтобы отображать заголовок страницы, отобразим некоторые её части. Например, ссылки на проекты Викимедиа, синонимы, метки или утверждения (листинг 16.12).

Листинг 16.12. Отображение ссылок, синонимов, меток и утверждений для персон из Польши

```
import pywikibot
from pywikibot import pagegenerators
site=pywikibot.Site('wikidata', "wikidata")
query = """
  SELECT DISTINCT ?item WHERE {
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
    ?item wdt:P31 wd:Q5.
    ?item wdt:P17 wd:Q36.
  } """

pages = pagegenerators.WikidataSPARQLPageGenerator(query, site=site)
for item in pages:
  item.get()
  print(item.sitelinks)
  print(item.aliases)
  print(item.labels)
  print(item.claims)
```

У нас получилось извлечь информацию из объектов Викиданных, теперь попробуем изменить её.

Изменение значений, введённых в Викиданные

Поиск повторяющихся с течением времени ошибок — одна из задач программ-ботов. Ботов регулярно запускают, они ищут и исправляют ошибки. Разработчики ботов стараются улучшать поиск ошибок и делать работу ботов более надёжной. Рассмотрим следующий запрос, который возвращает все профессии, отсортированные по количеству их использования в Викиданных (листинг 16.13).

Наберите и запустите в консоли на своём компьютере предложенные в этой главе листинги программ. Текст, который будут отображать эти программы в консоли, поможет вам понять, как они работают.

Листинг 16.13. Список профессий, отсортированных по количеству их использования в Викиданных[13]

```
SELECT ?instanceLabel ?value WHERE {
  {
    SELECT ?instance (COUNT(DISTINCT ?item) AS ?value)
    WHERE { ?item wdt:P106 ?instance. } # P106 - occupation
    GROUP BY ?instance
    ORDER BY DESC (?value)
  }
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
ORDER BY DESC (?value)
```

Запрос 16.13 позволяет найти ошибки (случайные или намеренные) в значении свойства профессия (P106). В полученном списке у некоторых объектов в свойстве «профессия» указана неверная или бессмысленная информация, например Salinas de Ibargoiti (Q7404672), Order of Saint Basil the Great (Q7319129), Promoted to Glory (Q7249866), Princes (Q7244433) и Point Loma Nazarene University (Q7208069). Мы можем составить список этих ошибок (массив `delete_occupation`) и найти объекты тех персон, у которых в поле «профессия» есть ошибки (листинг 16.14).

[13] Получено: 16 979 профессий в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4bLo.

Листинг 16.14. Отображение персон с ошибками в поле «профессия»

```
import pywikibot
from pywikibot import pagegenerators
site=pywikibot.Site('wikidata', "wikidata")
delete_occupation={"Q7404672", "Q7319129", "Q7249866", "Q7244433",
"Q7208069"}
for occupation in delete_occupation:
  query = """
    SELECT DISTINCT ?item WHERE {
      SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en
    ".}
      ?item wdt:P31 wd:Q5.      # is human
      ?item wdt:P106 wd:""" + occupation + """.
    } """

  pages=pagegenerators.WikidataSPARQLPageGenerator(query, site=site)
  for item in pages:
    print("Люди, чья профессия {occupation}: {title}".format(occupation=
    occupation, title=item.title()))
```

Итак, список «ошибочных» профессий (объектов Викиданных), подлежащих удалению, сохранён в массиве `delete_occupation`. Мы обходим этот список с помощью цикла `for`, чтобы с помощью SPARQL-запроса получить список людей для каждой «подозрительной» профессии. В отличие от листинга 16.13, в листинге 16.14 есть переменная `occupation`, которая берёт значение из массива `delete_occupation` и подставляет его в SPARQL-запрос.

Теперь, если мы уверены, что информация о профессии неверна, и хотим её удалить, то достаточно указать свойство, которое нужно удалить.

Есть ещё одна трудность. Дело в том, что персона может иметь несколько профессий, а удалить мы хотим только одну. Для удаления этой одной профессии нам нужно знать её значение. Пример такого удаления профессии по известному значению показан в листинге 16.15.

Обратите внимание, что до сих пор мы обращались к функции `print()` для печати значений переменных, но в листинге 16.14 для большей наглядности использована функция `format()`.

Листинг 16.15. Корректное удаление значения в поле «профессия»

```
import pywikibot
from pywikibot import pagegenerators
site=pywikibot.Site('wikidata', "wikidata")
delete_occupation={"Q7404672", "Q7319129", "Q7249866", "Q7244433",
"Q7208069"}
for occupation in delete_occupation:
  query = """
    SELECT DISTINCT ?item WHERE {
      SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
      ?item wdt:P31 wd:Q5.
      ?item wdt:P106 wd:""" + occupation + """.
    } """

  pages = pagegenerators.WikidataSPARQLPageGenerator(query, site=site)
  for item in pages:
    print("Люди, чья профессия {occupation}: {title}".format(occupation=
    occupation, title=item.title()))
    item.get()

    for valor in item.claims['P106']:
      if (str(valor.getTarget())=='[[wikidata:' + occupation + ']]'):
        print("<<<<< Удалить {occupation} от {title} >>>>>" . format(
    occupation=occupation, title=item.title()))
        item.removeClaims(valor, summary=u'Удаление ошибочных знач. в P106')
```

В листинге 16.15 мы сохранили код листинга 16.14, но добавили последние строки, в которых просматриваем значения, содержащиеся в свойстве (P106), с помощью свойства `claims`. Если это совпадает с искомой профессией, мы удаляем его методом `removeClaims`. При вызове этого метода нужно указать удаляемое значение и комментарий к редакторской правке[14], сохраняемый в истории правок вики-страницы.

Обратите внимание, что такие изменения могут вызвать множество проблем с Викиданными и администраторы могут заблокировать вашу учётную запись. Вначале внесите несколько изменений и всегда проверяйте историю изменений, чтобы узнать, какая информация была изменена.

Добавьте больше ошибочных и неправильных профессий в переменную `delete_occupation` и запустите бота для удаления этих свойств.

[14] Комментарии к правкам (или summary) позволяют другим редакторам быстро понять, какие изменения были внесены на страницу или в объект Викиданных.

Заключительная часть

Ответы

В каждой из глав на полях даются задания, а ответы на них собраны здесь.

◈ Вопрос на с. 15.

Экземпляр объекта[1] в Викиданных и в объектно-ориентированном программировании (ООП) сходны в сути, а именно: есть модель базового объекта D, создаётся новая единица I, обладающая свойствами той же модели D. В программировании о создании I говорят, что класс D проинициализирован и получен объект I[2].

В чём разница? В ООП в исходном коде программы мы видим, как во времени последовательно (в разных строках программы) происходит объявление переменной, инициализация класса, присвоение значений экземплярам класса. В Викиданных в тот момент, когда выполняется скрипт и происходит обращение к данным, объекты, являющиеся экземплярами других объектов, уже представлены и обычно не происходит их изменение, связанное с работой SPARQL-скриптов.

Таким значком-лампочкой отмечены на полях вопросы. Не на все из них, но на большинство ответы представлены ниже.

[1] См. статью Instance (computer science) в Английской Википедии.

[2] I является экземпляром D, или I is instance of D — по-английски.

Таким орнаментом мы разделим ответы на вопросы из разных глав.

◈ Вопрос на с. 28.

Веб-сайты есть у следующих российских производителей: «МиГ», «Туполев» и «Сухой». Эти сайты можно получить с помощью запроса 16.16.

Запрос 16.16. Веб-сайты российских авиазаводов[3]

```
SELECT ?manufacturer ?manufacturerLabel ?site WHERE
{
  ?manufacturer wdt:P31 wd:Q936518. # instance of aerospace manufacturer
  ?manufacturer wdt:P17 wd:Q159. # country Russia
  ?manufacturer wdt:P856 ?site # official website
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
```

[3] Получено: 14 авиазаводов в России с веб-сайтами на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/t4H.

◈ Вопрос на с. 29.

Компания «Туполев» была основана в 1922 году, «МиГ» и «Сухой» — в 1939-м, «Вымпел» — в 1949 году. Ответ на вопрос можно получить, выполнив запрос 16.17.

Запрос 16.17. Даты основания отечественных авиазаводов[4]

```
# Russian aircraft factories sorted by inception years
SELECT ?manufacturer ?manufacturerLabel (YEAR(?inception) AS ?year) WHERE
{
  ?manufacturer wdt:P31 wd:Q936518;  # is aerospace manufacturer
                wdt:P17 wd:Q159;     # country Russia
                wdt:P571 ?inception. # foundation date
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
ORDER BY ?year
```

[4] Получено: 15 отечественных авиазаводов с датой основания на 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/vaH.

◈ Вопрос на с. 30.

Штаб-квартира компании «Камов» находится в городе Люберцы, «Авиадвигатель» — в Перми, «Улан-Удэнский авиационный завод» — в городе Улан-Удэ, «Сухой» — в Москве. Ответ на вопрос можно получить, выполнив запрос 16.18.

Запрос 16.18. Штаб-квартиры компаний[5]

```
SELECT ?manufacturer ?manufacturerLabel ?inceptionLabel WHERE
{
    ?manufacturer wdt:P31 wd:Q936518. # instance of aerospace manufacturer
    ?manufacturer wdt:P17 wd:Q159. # country Russia
    ?manufacturer wdt:P159 ?inception # headquarters location
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
```

[5] Получено: 16 российских авиазаводов, имеющих штаб-квартиры в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/t4X.

◈ Вопрос на с. 31.

Воздушным судном, удерживаемым в воздухе огромным баллоном с горючим и смертельно опасным газом, расположенным прямо над головами пассажиров, является дирижабль.

◈ Вопрос на с. 32.

Воздушное судно, изображенное на с. 32, — это дирижабль СССР–В6 «Осоавиахим» (1934–1938), который установил мировой рекорд в 1937 году, пролетев 130 с половиной часов без посадки. Для построения «плиточки» (ImageGrid) из иллюстраций дирижаблей выполните запрос 16.19.

Запрос 16.19. Изображения дирижаблей[6]

```
#defaultView:ImageGrid
SELECT ?airship ?airshipLabel ?image WHERE
{
    ?airship wdt:P31 wd:Q133585. # instance of airship
    ?airship wdt:P18 ?image # image airship
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
```

[6] Получено: 18 дирижаблей с иллюстрациями в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/t4c.

◈ Вопрос на с. 47.

Плотность населения в Алейске составляет 648 чел./км^2, в Барабинске — 413 чел./км^2, то есть в Алейске плотность выше.

Плотность населения по населённым пунктам России можно получить с помощью запроса 16.20.

Обратите внимание, что площадь может быть задана в разных единицах. Например, у Алейска (Q11304591) площадь в Викиданных указана в гектарах, у Барабинска (Q104609) — в квадратных километрах. Для нормализации данных и перевода площади в метры в строке 8 запроса 16.20 указан префикс `psn:`.

В строке 2 делим `?area` на миллион, чтобы перевести квадратные метры в километры. Результат деления записываем в переменную `?popArea`, которая показывает плотность населения.

Запрос 16.20. Плотность населения «населенных пунктов» России[7]

```
1  # population density of human settlements in Russia
2  SELECT ?hum ?humLabel ?population ?area (?population / (?area / 1000000) as ?popArea)
3  WHERE {
4      ?hum wdt:P31 wd:Q486972; # is human settlement
5           wdt:P17 wd:Q159;    # in Russia
6
7      # The psn: prefix normalizes the values to a common unit of area
8      p:P2046/psn:P2046/wikibase:quantityAmount ?area;  # get the area
9
10     wdt:P1082 ?population. # has ?population
11     FILTER(?population>0 && ?area).
12     SERVICE wikibase:label{bd:serviceParam wikibase:language "ru,en"}
13 }
14 ORDER BY DESC (?popArea)
```

[7] Получено: 131 населённый пункт в России с известной плотностью населения на 2021 год, 152 пункта на 2023 год. SPARQL-запрос: https://w.wiki/6ibN.

◈ Вопросы на с. 47, 48, 50, 51, 54.

Гербы, изображенные на с. 47, 48 и 54, являются гербами отечественных населённых пунктов. Герб на с. 50 принадлежит чешскому населённому пункту, герб на с. 51 — украинскому.

Ответ можно получить с помощью запроса 16.21. Значение свойства coat of arms image (P94) содержит изображение герба населённого пункта.

Запрос 16.21. Гербы населённых пунктов Российской Федерации[8]

```
# Emblems of human settlements in Russia
#defaultView:ImageGrid
SELECT ?hum ?humLabel ?image WHERE {
  ?hum wdt:P31 wd:Q486972; # instance of human settlement
       wdt:P17 wd:Q159;    # in Russia
       wdt:P94 ?image.     # coat of arms
  SERVICE wikibase:label{bd:serviceParam wikibase:language "ru, en"}
}
```

[8] Получено: 148 гербов на 2022 год. SPARQL-запрос: https://w.wiki/4e8A.

◈ Вопрос на с. 71.

В честь географических объектов были названы Тула (река Тулица), Курильск (Курильские острова) и Вологда (река Вологда). Ответ на вопрос можно получить, выполнив запрос 16.22. Значение свойства named after (P138) показывает, в честь какого объекта Викиданных был назван город.

Запрос 16.22. Города, названные в честь географических объектов[9]

```
1  SELECT DISTINCT ?city ?cityLabel ?namedAfter ?namedAfterLabel
2  WHERE {
3    ?city wdt:P31/wdt:P279* wd:Q7930989. # instances/subclasses of "city/town"
4    ?city wdt:P138 ?namedAfter.          # with filled property "named after"
5    FILTER(?city = wd:Q1341 || ?city = wd:Q156046 ||
6           ?city = wd:Q2770 || ?city = wd:Q1957 ||
7           ?city = wd:Q5655 || ?city = wd:Q175651)
8    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
9  }
```

В строке 3 запроса 16.22 после конструкции `wdt:P31/wdt:P279*` следует объект Викиданных, объединяющий city (Q515) и town (Q3957) и называющийся city/town (Q7930989). Такой поиск с помощью подклассов позволяет найти экземпляры сразу обоих типов городов. Подробности см. в главе «Полнота и недостатки Викиданных: дублирование объектов», в тексте, предшествующем запросу 7.19 на с. 81.

◈ Вопрос на с. 76.

Более 400 лет назад были основаны Казань (1005 год), Москва (1147), Астрахань (1558), Воронеж (1586) и Самара (1586). Самым молодым городом оказался Саров, основанный в 1691 году. Ответ на вопрос можно получить, выполнив запрос 16.23.

[9] Для 6 городов, заданных в строках 5–7, получено 6 объектов, в честь которых названы города. Из них 3 города названы в честь географических объектов (см. выше), 2 — в честь известных людей и 1 город — в честь железнодорожной станции. Ссылка на SPARQL-запрос: https://w.wiki/6icv.

Запрос 16.23. Города, основанные более 400 лет назад в России[10]

```
1  SELECT ?city ?cityLabel (YEAR(?inceptionDate) AS ?year)
2  WHERE {
3      ?city wdt:P31/wdt:P279* wd:Q7930989. # instances of "city/town"subclasses
4      ?city wdt:P17 wd:Q159.               # in Russia
5      ?city wdt:P571 ?inceptionDate.       # with filled property "inception"
6      FILTER (YEAR(?inceptionDate) < 1620). # 2020 - 400 years
7      FILTER(?city = wd:Q649 || ?city = wd:Q193522 || ?city = wd:Q900 ||
8             ?city = wd:Q3927 || ?city = wd:Q894 || ?city = wd:Q3426)
9      SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
10 }
11 GROUP BY ?city ?cityLabel ?inceptionDate
12 ORDER BY ASC(?year)
```

Подумайте, какие строки в запросе 16.23 нужно закомментировать, чтобы получить, во-первых, список всех отечественных городов, во-вторых, список городов всего мира, основанных более 400 лет назад.

Как вы думаете, может ли в запросе 16.23 переменная `?year` принимать отрицательные значения? Если «да», то почему?

Обычно значение свойства inception (P571) содержит дату[11] основания города. Однако у объекта Викиданных Москва (Q649) свойство inception принимает значение unknown value (неизвестное значение) с квалификатором «самое позднее упоминание» (latest date (P1326)), равным 4 апреля 1147 года. Вероятно, по этой причине в запросе 16.23 переменная `?year` принимает для Москвы пустое значение, и Москва ошибочно не попадает в список правильных ответов. Таким образом, чтобы извлечь 1147 год, необходимо доработать имеющийся скрипт, что мы и предоставим читателю.

◈ Вопрос на с. 78.

Флаг, изображенный на рисунке, принадлежит городу Карабулак. Ответ на вопрос можно получить с помощью запроса 16.24. Значение свойства flag image (P41) содержит изображение флага города.

[10] Из 6 городов, заданных в строках 7–8, получено 4 города в России, основанные до 1620 года. Ссылка на SPARQL-запрос: https://w.wiki/6idC.

[11] О типах данных и наборах свойств, связанных с датой и временем, см. справку на странице Викиданных Help:Dates.

Запрос 16.24. Флаги городов России[12]

```
#defaultView:ImageGrid
SELECT ?city ?cityLabel ?flag ?countryLabel WHERE {
    ?city wdt:P31/wdt:P279* wd:Q7930989; # instances/subclasses of "city/town"
          wdt:P17 wd:Q159;               # in Russia
          wdt:P41 ?flag.                 # with filled property "flag"
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
```

[12] Получено: 980 городов России имеют флаги на 2023 год. Ссылка на SPARQL-запрос: https://w.wiki/8orr.

◈ Вопрос на с. 83.

Речь идёт о количестве административно-территориальных единиц в каждой из стран, например: штаты, края или области. Количество таких единиц по странам можно получить с помощью запроса 16.25.

Запрос 16.25. Список стран, упорядоченных по количеству административно-территориальных единиц[13]

```
# Countries sorted by number of administrative territories
SELECT ?country ?countryLabel  (count(*) as ?count) WHERE
{
    ?country p:P31 [ps:P31 wd:Q6256]; # is a country
             wdt:P150 [].         # has some administrative territory
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }
}
GROUP BY ?country ?countryLabel
ORDER BY DESC(?count)
```

[13] Получено: 199 стран на 2021 год. Ссылка на SPARQL-запрос: https://w.wiki/8ory.

◈ Вопрос на с. 85.

Результатом выполнения запроса 16.26 будет список «исторических государств (Q3024240)», то есть империй, цивилизаций, городов-государств, канувших в лету. Две с половиной тысячи лет и больше просуществовали только 6 из них: город-государство Угарит (4810 лет), Древний Египет (3971 год), островная страна Тамла (3740 лет), цивилизация Майя (3521 год), Мероитское царство (2529 лет) и город-государство Дильмун (2500 лет).

Запрос 16.26. Список исторических стран, упорядоченных по дате основания[14]

```
# List of historical countries sorted by inception date
SELECT ?country ?countryLabel
(MIN(?start) AS ?min_year)
(MAX(?end)   AS ?max_year)
(?max_year - ?min_year as ?age) WHERE
{
    ?country p:P31 [ps:P31 wd:Q3024240]. # instance of a historical country

    FILTER EXISTS {?country wdt:P571 []}.# skip countries without inception date
    FILTER EXISTS {?country wdt:P576 []}.# skip countries without dissolution date

    OPTIONAL {?country p:P571 [ps:P571 ?inception].}# any inception date
    OPTIONAL {?country p:P576 [ps:P576 ?dissolution].}# any dissolution date

    BIND(YEAR(?inception) AS ?start)
    BIND(YEAR(?dissolution) AS ?end)
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en" }
}
GROUP BY ?country ?countryLabel ?min_year ?max_year ?age
ORDER BY DESC(?age)
```

[14] Ссылка на SPARQL-запрос: https://w.wiki/tYc.

◈ Вопрос на с. 85.

Площадь Израиля составляет 20 770 км^2 с населением 9.8 млн чел., площадь Монголии — 1 564 116 км^2 с населением 3.4 млн чел., площадь Республики Корея — 100 295 км^2 с населением 51.47 млн чел., а площадь Сингапура — 719.1 км^2 с населением 5.87 млн человек. Таким образом, по возрастанию плотности населения страны будут упорядочены так:

1. Монголия (2.18 чел. на км^2);
2. Израиль (471.7 чел. на км^2);
3. Корея (513.15 чел. на км^2);
4. Сингапур (8157.6 чел. на км^2).

Ответ на вопрос можно получить с помощью запроса 16.27.

Запрос 16.27. Плотность населения в странах Азии[15]

```
# Population density in Asian countries
SELECT ?country ?countryLabel ?flag ?area ?population
(?population / ?area as ?populationDensity)
{
    ?country p:P31 [ps:P31 wd:Q6256]; # this is a country
             wdt:P30 wd:Q48;          # on the Asian continent
             wdt:P41 ?flag;           # has flag
             wdt:P2046 ?area;         # has area
             wdt:P1082 ?population. # has population
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
}
ORDER BY DESC(?populationDensity)
```

[15] Получено: 52 страны в 2024 году. Ссылка на SPARQL-запрос: https://w.wiki/8p8n.

Результаты работы вы можете увидеть в виде «плиточки» из флагов. Для этого на странице *Wikidata Query Service* под кнопкой запуска скрипта в выпадающем списке выберите *Image grid*.

◈ Вопрос на с. 85.

Официальными языками России являются абазинский, мокшанский и эрзянский языки. Ответ на вопрос можно проверить, выполнив запрос 16.28. В этом запросе в строке 4 дважды используется свойство official language (P37). Сначала свойство `p:P37` позволяет извлечь у объекта России (Q159) конструкцию, описывающую официальный язык. Затем с помощью строки `[ps:P37 ?lanquage]` мы извлекаем из этой конструкции сам язык.

Если бы в строке 4 запроса 16.28 было записано обычное выражение с отношением `wdt:P`[16], то результатом запроса был бы один язык — русский — поскольку он имеет преимущество в ранжировании (preferred rank), а у остальных официальных языков объекта Россия указан обычный ранг (normal rank).

[16] Выражение: `wd:Q159 wdt:P37 ?lanquage.`

Запрос 16.28. Официальные языки в России[17]

```
1 # Official languages in Russia
2 SELECT ?lanquage ?lanquageLabel WHERE
3 {
4   wd:Q159 p:P37 [ps:P37 ?lanquage]. # Russia has the official language
5   SERVICE wikibase:label {bd:serviceParam wikibase:language "ru"}
6 } ORDER BY ?lanquageLabel
```

[17] Получено: 37 языков в 2020–2024 годах. Ссылка на SPARQL-запрос: https://w.wiki/tky.

◈ Вопрос на с. 93.

Приведённому описанию соответствует флаг Московской области. Иллюстрации флагов для каждого из субъектов России можно получить с помощью запроса 16.29.

Запрос 16.29. Флаги субъектов России[18]

```
# List of flags of the subjects of Russia
#defaultView:ImageGrid
SELECT ?subject ?subjectLabel ?flag
WHERE
{
  { ?subject wdt:P31 wd:Q835714 } UNION  # Oblast of Russia
  { ?subject wdt:P31 wd:Q41162 } UNION  # Republic of Russia
  { ?subject wdt:P31 wd:Q183342 } UNION  # Federal city of Russia
  { ?subject wdt:P31 wd:Q831740 } UNION  # Krai of Russia
  { ?subject wdt:P31 wd:Q309166 } UNION # Autonomus oblast of Russia
  { ?subject wdt:P31 wd:Q184122 } # Autonomus okrug of Russia

  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }

  ?subject wdt:P41 ?flag
}
```

[18] Получено: 86 записей в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/4cPg.

◈ Вопрос на с. 95.

Таким регионом является Республика Карелия. Она расположена на северо-западе России, образована в 1920 году. Граничит с Ленинградской, Вологодской, Архангельской и Мурманской областями. Также граничит с Финляндией на западе. Ответ на вопрос можно получить с помощью запроса 16.30.

Запрос 16.30. Границы и даты образования субъектов РФ[19]

```
# Borders and date of origin of the subjects of the Russian Federation
SELECT ?subject ?subjectLabel ?sharesBorderWith ?sharesBorderWithLabel ?year
WHERE {
  { ?subject wdt:P31 wd:Q835714 } UNION  # Oblast of Russia
  { ?subject wdt:P31 wd:Q41162 } UNION  # Republic of Russia
  { ?subject wdt:P31 wd:Q183342 } UNION  # Federal city of Russia
  { ?subject wdt:P31 wd:Q831740 } UNION  # Krai of Russia
  { ?subject wdt:P31 wd:Q309166 } UNION # Autonomus oblast of Russia
  { ?subject wdt:P31 wd:Q184122 } # Autonomus okrug of Russia

  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru" }

  ?subject wdt:P47 ?sharesBorderWith.
  ?subject wdt:P571 ?year.      # ?year is inception date of ?subject
}
```

[19] Получено: 485 пар граничащих друг с другом регионов России в 2021 году. Ссылка на SPARQL-запрос: https://w.wiki/6hwN.

◈ Вопрос на с. 101.

Сейчас в состав Российской Федерации входят Республика Адыгея, Камчатский край, Чукотский автономный округ. Читинская область не входит, поскольку это бывший субъект РФ. Территория упразднённой Читинской области является частью современного Забайкальского края. Ответ на вопрос можно проверить, выполнив запрос 9.2 на с. 94.

◈ Вопрос на с. 105.

Запрос 16.31 подсчитывает для каждой операционной системы количество последующих операционных систем, основанных на её коде. На 2020 год максимумом было 10 операционных систем, разработанных на основе системы Ubuntu. На 2024 год — 25 операционных систем, созданных на основе системы Darwin.

Отметим, что обе эти системы развиваются до сих пор. Первая версия Darwin вышла в 2000 году, Ubuntu — в 2004 году.

Запрос 16.31. Список базовых операционных систем[20]

```
1 SELECT ?baseLabel (COUNT(*) AS ?count) WHERE
2 {
3   ?os wdt:P31 wd:Q9135. # is instance of operating system
4   ?os wdt:P144 ?base.   # is based on ?base
5   SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
6 }
7 GROUP BY ?baseLabel
8 ORDER BY DESC(?count) ASC(?baseLabel)
```

Запрос 16.31 интересен тем, что мы явно не задаём экземпляром какого объекта должна быть искомая базовая операционная система `?base`. Мы только указываем, что некоторая операционная система `?os` (строка 3) основана на искомой системе `?base` (строка 4).

Поэтому мы получили в списке базовых операционных систем как Darwin, являющуюся экземпляром *Unix-подобной операционной системы*, так и *операционную систему для мобильных устройств* Android.

◈ Вопрос на с. 105.

Компания Apple разработала операционную систему Newton OS. Ответ на вопрос можно получить, выполнив запрос 16.32.

Запрос 16.32. Разработчики операционных систем[21]

```
SELECT ?os ?osLabel ?developer ?developerLabel WHERE {
  ?os wdt:P31 wd:Q9135. # instance of operating system
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
  OPTIONAL { ?os wdt:P178 ?developer. }
}
```

[20] Получено: 118 операционных систем на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/uLR.

[21] Получено: 1115 операционных систем с указанием разработчиков и без них на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/n8a.

◈ Задание 1 на с. 112.

Список разработчиков операционных систем формируется запросом 16.33.

Запрос 16.33. Разработчики операционных систем[22]

```
SELECT ?os ?osLabel ?developer ?developerLabel WHERE {
    ?os wdt:P31 wd:Q9135. # os is instance of operating system
    ?os wdt:P178 ?developer. # os developed by developer
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
}
```

[22] Получено: 548 операционных систем с заполненным свойством «разработчик» на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/vGQ.

◈ Задание 2 на с. 112.

Список логотипов операционных систем можно получить с помощью запроса 16.34.

Запрос 16.34. Логотипы операционных систем[23]

```
SELECT ?os ?osLabel ?image WHERE {
    ?os wdt:P31 wd:Q9135.
    ?os wdt:P18 ?image.
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
}
```

[23] Получено: 182 операционные системы с заполненным свойством «логотип» на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/vGQ.

◈ Задание 3 на с. 112.

Список стран происхождения операционных систем можно получить с помощью запроса 16.35.

Запрос 16.35. Страны происхождения операционных систем[24]

```
SELECT ?os ?osLabel ?country ?countryLabel WHERE {
  ?os wdt:P31 wd:Q9135.
  ?os wdt:P495 ?country.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
}
```

[24] Получено: 10 операционных систем с заполненным свойством «страна происхождения» на 2020 год. Ссылка на SPARQL-запрос: https://w.wiki/vGX.

◈ Задание 4 на с. 112.

Для получения дерева, где операционные системы его верхнего уровня основаны на ОС, перечисленных на нижнем уровне, выполните запрос 16.36. Такое дерево представлено на рис. 10.4.

Запрос 16.36. Дерево операционных систем и их основ[25]

```
#defaultView:Tree
SELECT ?base ?baseLabel ?baseImage ?baseLogo
       ?os ?osLabel ?osImage ?osLogo WHERE
{
  ?os wdt:P31 wd:Q9135; # is instance of operating system
      wdt:P144 ?base.   # and based on ?base
  OPTIONAL { ?base wdt:P18 ?baseImage. }
  OPTIONAL { ?base wdt:P154 ?baseLogo. }
  OPTIONAL { ?os wdt:P18 ?osImage. }
  OPTIONAL { ?os wdt:P154 ?osLogo. }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
}
```

[25] Получено: 363 операционные системы с «предками» на 2024 год. Ссылка на SPARQL-запрос: https://w.wiki/8pfp.

◈ Вопрос на с. 114.

Язык программирования Ада разработали Жан Ишбиа и С. Такер Тафт, язык Форт — Чарльз Мур, а создателями языка Erlang считаются Джо Армстронг, Роберт Вирдинг и шведская компания Ericsson. Ответ на вопрос можно получить с помощью запроса 16.37.

Запрос 16.37. Создатели языков программирования[26]

```
# Get developers of programming languages
SELECT ?itemLabel ?developerLabel WHERE
{
  ?item wdt:P31 wd:Q9143;    # is programming language
        wdt:P178 ?developer. # has developer
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en" }
}
ORDER BY DESC (?item_label)
```

[26] Получено: 520 разработчиков в 2020 году, 553 разработчика в 2023 году. Ссылка на SPARQL-запрос: https://w.wiki/6i28.

◈ Вопрос на с. 117.

Логотипом языка программирования LOLCODE является третья картинка на с. 117. Ответ на вопрос можно получить, выполнив запрос 16.38.

Запрос 16.38. Логотипы языков программирования[27]

```
# Get logo images of programming languages
#defaultView:ImageGrid
SELECT ?item ?itemLabel ?image WHERE
{
  ?item wdt:P31 wd:Q9143;  # is programming language
        wdt:P154 ?image.   # has logo image
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en" }
}
```

[27] Получено: 174 логотипа языков программирования в 2024 году. Ссылка на SPARQL-запрос: https://w.wiki/8pu3.

◈ Задание 1 на с. 126.

Список языков программирования со свойством персонаж-талисман (P822) (маскóт) можно получить, выполнив запрос 16.39. Подумайте, как расширить запрос, чтобы увидеть иллюстрации этих маскотов.

Запрос 16.39. Персонажи-талисманы языков программирования[28]

```
# List of programming languages with mascot
SELECT ?lang ?langLabel ?mascot ?mascotLabel
WHERE {
    ?lang wdt:P31 wd:Q9143.
    ?lang wdt:P822 ?mascot.
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en" }
}
```

[28] Получено: 3 языка программирования на 2020 год, 6 языков на 2023 год. Ссылка на SPARQL-запрос: https://w.wiki/6i8q.

◈ Задание 2 на с. 126.

Получить список языков программирования, разработанных до 1992 года, можно с помощью запроса 16.40.

Запрос 16.40. Языки программирования, появившиеся до 1992 года[29]

```
# Programming languages developed before 1992
SELECT DISTINCT ?lang ?langLabel (year(?inception) as ?year) WHERE
{
  ?lang wdt:P31 wd:Q9143;    # is programming language
        wdt:P571 ?inception. # date of inception
  FILTER(year(?inception) < 1992)
  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en" }
}
```

[29] Получено: 207 языков программирования на 2020 год, 377 языков на 2023 год. Ссылка на SPARQL-запрос: https://w.wiki/8qNZ.

◈ Задание 3 на с. 126.

Чтобы построить столбчатую диаграмму числа различных хештегов для каждого из языков программирования, используем свойство hashtag (P2572) в запросе 16.41.

Отметим, что в 2024 году только два языка имели больше одного хештега: язык C++ (хештеги `#cpp` и `#cplusplus`) и язык Go (хештеги `#golang` и `#GoogleGo`).

Запрос 16.41. Число хештегов у языков программирования[30]

```
# Number of hashtags for programming languages
#defaultView:BarChart
SELECT DISTINCT ?lang ?langLabel (count(*) as ?count) WHERE
{
    ?lang wdt:P31 wd:Q9143; # is programming language
          wdt:P2572 ?count. # has hashtag
    SERVICE wikibase:label { bd:serviceParam wikibase:language "ru,en" }
}
GROUP BY ?lang ?langLabel
ORDER BY DESC(?count)
```

[30] Получено: 3 языка программирования с хештегами в 2020 году, 11 языков — в 2024 году. Ссылка на SPARQL-запрос: https://w.wiki/8qRA.

◈ Задание на с. 128.

«Корабли Гиннесса». Выполним поиск выдающихся по каким-либо параметрам кораблей в Викиданных. Например, известно, что Seawise Giant — это самый длинный нефтяной танкер.

Приведём примеры возможных решений через Викиданные. При этом данные могут быть неточными из-за неполноты информации об объектах в Викиданных, см. запросы 16.42 и 16.43.

Запрос 16.42. Самые длинные корабли[31]

```
# The ship with maximum length
SELECT ?ship ?shipLabel ?max_length WHERE
{
  { SELECT (MAX(?length) as ?max_length) WHERE
    {
        ?ship wdt:P31 wd:Q11446; # is ship
              wdt:P2043 ?length
    }
  }
  {?ship wdt:P31 wd:Q11446;      # is ship
         wdt:P2043 ?max_length}

  SERVICE wikibase:label { bd:serviceParam wikibase:language "ru, en" }
}
```

Внутренний SELECT запроса 16.42 находит максимальную длину корабля, а внешний SELECT находит корабль с такой длиной.

Более простой и прямолинейный скрипт (URL: https://w.wiki/6iAd) находит 167 кораблей длиной более 350 метров на 2023 год.

[31] Самым длинным кораблём оказался непостроенный «Хабаккук» длиной 1200 метров (проект британских кораблестроителей). Ссылка на SPARQL-запрос: https://w.wiki/3L2R.

Запрос 16.43. Изображения кораблей, отсортированных по ширине корабля[32]

```
# List of ships' images sorted by width of ship
#defaultView:ImageGrid
SELECT ?ship ?shipLabel ?image ?beam WHERE
{
    ?ship wdt:P31 wd:Q11446; # is ship
          wdt:P2261 ?beam;   # width of ship is ?beam
    OPTIONAL { ?ship wdt:P18 ?image }
    SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
}
ORDER BY DESC(?beam)
```

[32] Самым широким кораблем также оказался проект авианосца «Хабаккук» из льда и опилок, ширина которого составила 180 метров. Среди реально существовавших кораблей самым широким был норвежский корабль M/S Isosaari (Q11987454) шириной 99 метров. SPARQL-запрос: https://w.wiki/wKv.

◈ Задание на с. 129.

На марке изображён советский эсминец «Гремящий».

◈ Задание на с. 130.

Найдём корабли, снятые в фильмах, с помощью запроса 16.44.

Запрос 16.44. Изображения кораблей, снятых в фильмах[33]

```
1  # Images of ships used in movies
2  #defaultView:ImageGrid
3  SELECT DISTINCT ?film ?filmLabel ?ship ?shipLabel ?image WHERE
4  {
5    VALUES ?ship_boat {wd:Q2235308   # is ship
6                       wd:Q16103215} # or boat type
7    ?film wdt:P31 wd:Q11424; # is film
8          wdt:P921 ?ship.    # with ship as main subject
9    ?ship wdt:P31+ ?ship_boat. # is ship or boat type
10
11   OPTIONAL { ?ship wdt:P18 ?image } # ship's image
12   SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en"}
13 }
```

[33] Найдено 11 кораблей, снятых в кинолентах, при этом 9 из них имеют изображения, по данным на 2024 год. Ссылка на SPARQL-запрос: https://w.wiki/8qat.

В строке 8 запроса 16.44 используется свойства main subject (P921), то есть основной или главной темой фильма должен быть указан корабль (переменная `?ship`).

В строке 9 запроса 16.44 знак + аналогичен квантификаторам регулярного выражения и означает, что свойство is instance (P31) может повториться один или более раз. То есть `wdt:P31+` можно раскрыть как `wdt:P31` (переменная `?ship` является экземпляром корабля) или `wdt:P31\wdt:P31` (переменная `?ship` является экземпляром некоторого объекта, который является экземпляром корабля) и так далее.

Обратите внимание на использование в той же строке 9 массива `?ship_boat`, который состоит из двух элементов и задаётся в строках 5–6 с помощью оператора `VALUES`.

Найдём корабли, о которых написано в книгах, с помощью запроса 16.45. Поясним для этого и предыдущего скрипта, что будут найдены только такие пары корабль-книга (корабль-фильм), для которых существуют в Викиданных и объект-корабль, и объект-книга. Таких пар не может быть много.

В строке 9 запроса 16.45 предложен следующий *путь поиска*: у книги в качестве главной темы (main subject (P921)) указан некоторый объект Викиданных (квадратные скобки здесь — это *анонимная переменная*), и у этого объекта должно быть заполнено свойство «судно» (vessel (P1876)). Недостаток этого «пути» в том, что судном в Викиданных являются и космический корабль, и Ноев ковчег, но основную массу составляют морские корабли.

Для построения этого «пути» мы проанализировали такие объекты, как: (1) книга «Путешествие на „Кон-Тики“» (Q3203414) и (2) плот Кон-Тики (Q204025), на котором плыл Тур Хейердал. Анализ связи этих объектов и позволил построить путь поиска в строке 9. От частного мы пришли к общему.

Запрос 16.45. Изображения кораблей, о которых писали в книгах[34]

```
1  # Images of vessels related to main subjects in books and written works
2  #defaultView:ImageGrid
3  SELECT ?book ?bookLabel ?ship ?shipLabel ?image WHERE
4  {
5    VALUES ?book_work {wd:Q571       # book
6                       wd:Q47461344} # written work
7
8    ?book wdt:P31 ?book_work. # is book or written work
9    ?book wdt:P921 [wdt:P1876 ?ship]. # main subject (P921) is something with vessel
10                                     # (P1876)
11   OPTIONAL { ?ship wdt:P18 ?image } # ship's image
12   SERVICE wikibase:label {bd:serviceParam wikibase:language "ru, en, de"}
13 }
```

[34] Найдено 30 судов, о которых было написано в книгах, на 2024 год. Из них 27 судов имели изображения. Ссылка на SPARQL-запрос: https://w.wiki/8rxm.

◈ Вопросы на с. 136 и 137.

Космические аппараты, изображённые на с. 136 («Союз-19») и с. 137 («Союз-Т»), принадлежат СССР. Убедиться в этом можно, выполнив запрос 16.46.

Запрос 16.46. Набор изображений космических аппаратов СССР[35]

```
# List of images of USSR spacecraft
#defaultView:ImageGrid
SELECT ?rocket ?rocketLabel ?img WHERE
{
  ?rocket wdt:P31+ wd:Q40218; # is spacecraft
          wdt:P17 wd:Q15180;  # from USSR
          wdt:P18 ?img.       # has image
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
```

[35] Получено: 23 космических корабля в 2024 году. Ссылка на SPARQL-запрос: https://w.wiki/8sMK.

◈ Вопрос на с. 138.

Анализ Викиданных показывает, что объект *product model* используется в разных областях. На 2024 год с помощью запроса 16.47 была найдена только одна серия космических кораблей, а именно: серия кораблей «Союз 7К-Л1» (Q1959194), разрабатываемых для полёта к Луне. Обратите внимание, что в запросе 16.47 нет ограничений на страну. Полагаем, что с ростом Викиданных серий будет становиться больше.

Сравните в запросе 16.47 строки 4 и 5. В закомментированной строке 4 показан фрагмент привычного способа поиска экземпляра через отношение `wdt:P31`. Новое отношение `p:P31` в строке 5 позволяет получить утверждение (`?statement`) с квалификатором.

«Раскручиваем» объект с квалификатором в два шага. В строке 7 проверяем, что переменная `?statement` является экземпляром product model (Q10929058), при этом обращаемся к тому же свойству `P31`, что и в строке 5. В строке 8 проверяем, что квалификатор of (P642) имеет значение spacecraft (Q40218).

Запрос 16.47. Серии космических кораблей[36]

```
# List of product models of spacecrafts
SELECT ?rocket ?rocketLabel WHERE
{
# wdt:P31
  ?rocket p:P31 ?statement.      # is instance of something

  ?statement ps:P31 wd:Q10929058. # is product model
  ?statement pq:P642 wd:Q40218.  # of spacecraft

  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
```

[36] Получено: 1 серия космических кораблей на 2024 год. Ссылка на SPARQL-запрос: https://w.wiki/8xzi.

◈ Вопрос на с. 140.

Запрос 16.48 строит график запуска отечественных космических аппаратов по десятилетиям. В 1960-е годы совершено 19 запусков, в 1970-е — 43, а в 1980-е — 45 запусков космических аппаратов.

Запрос 16.48. Диаграмма запусков отечественных космических кораблей по десятилетиям[37]

```
#defaultView:BarChart
SELECT (STR(?lapse) AS ?lapse_str) (COUNT(?item) AS ?quantity)
WHERE {                            # spacecraft belongs to
        {?item wdt:P17 wd:Q15180}   # country = USSR
  UNION {?item wdt:P17 wd:Q159}     # country = Russia
  UNION {?item wdt:P495 wd:Q159}    # country of origin = Russia
  UNION {?item wdt:P495 wd:Q15180}. # country of origin = USSR
  ?item wdt:P619 ?launch. # date of spacecraft launch (P619)
  BIND( YEAR(?launch) AS ?year)
  BIND(FLOOR(?year/10)*10 AS ?lapse) # count for each 10 years
  SERVICE wikibase:label {bd:serviceParam wikibase:language "ru,en"}
}
GROUP BY ?lapse
ORDER BY ?lapse # Order 1960, 1970, 1980, ...
```

[37] Диаграмма построена для 9 десятилетий, включая запланированные полёты на 2030-е, данные на 2024 год. SPARQL-запрос: https://w.wiki/4eQK.

◈ Вопрос на с. 142.

Подсчитаем число запусков ракет за каждые 10 лет с 1970 по 2020 год с помощью запроса 16.49.

Запрос 16.49. Число запусков космических кораблей в мире по десятилетиям (1970–2020)[38]

```
# Get number of launches for each 10 years from 1970 to 2020
SELECT (STR(?lapse) AS ?lapse_str) (COUNT(?item) AS ?quantity)
WHERE {
    ?item wdt:P619 ?launch.       # date of spacecraft launch
    BIND( YEAR(?launch) AS ?year)
    ?item wdt:P17 ?country.       # check existing country
    BIND(FLOOR(?year/10)*10 AS ?lapse) # count for each 10 years
    FILTER (?year > 1969 && ?year < 2020) # check date
}
GROUP BY ?lapse
```

[38] Получено: в 1970-е годы было запущено минимальное число космических кораблей в мире — 66, в 2010-е годы состоялось максимальное число запусков — 341, по данным на 2024 год. Ссылка на SPARQL-запрос: https://w.wiki/6iFk.

Список литературы

Глушко, В. П. Космонавтика : энциклопедия / В. П. Глушко. — Москва : Сов. энциклопедия, 1985.

Журавлева, Н. Ю. Особенности перевода катойконимов (на материале испанского и русского языков) / Н. Ю. Журавлева // Вестник Российского университета дружбы народов. Серия: Лингвистика. — 2012. — № 3. — С. 41—48. — ISSN 2686-8024. — URL: http://journals.rudn.ru/linguistics/article/view/9530.

Кемп, Д. Будущее языков программирования / Д. Кемп ; пер. А. Панин. — 2016. — URL: http://rus-linux.net/MyLDP/algol/the_future_of_computer_languages.html.

Крижановский, А. А. Работа в вики-среде на примере Русской Википедии. Ч. 1 : учеб. пособие / А. А. Крижановский. — Петрозаводск : ПетрГУ, 2015. — URL: https://commons.wikimedia.org/?curid=86907665.

Кузьмина, Д. П. Аниме: от социализации к межкультурной коммуникации / Д. П. Кузьмина // Традиции и инновации в пространстве современной культуры. — Липецк, 2021. — С. 235—238. — URL: https://www.elibrary.ru/item.asp?id=46107074.

Магера, Ю. А. История появления первых японских комиксов на русском языке / Ю. А. Магера // Японские исследования. — 2018. — № 4. — С. 6—23. — ISSN 2500-2872. — DOI: 10.24411/2500-2872-2018-10025. — URL: https://publications.hse.ru/pubs/share/direct/229921306.pdf.

Falcon 2.0: An entity and relation linking tool over Wikidata / A. Sakor [et al.] // Proceedings of the 29th ACM International Conference on Information & Knowledge Management. — 2020. — P. 3141–3148. — URL: https://arxiv.org/pdf/1912.11270.pdf.

Fogarolli, A. Word Sense Disambiguation Based on Wikipedia Link Structure / A. Fogarolli // Semantic Computing, 2009. ICSC '09. IEEE International Conference on. — 2009. — P. 77–82. — DOI: 10.1109/ICSC.2009.7. — URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.1696&rep=rep1&type=pdf.

Giesemann, E. ProWD: Detecting Knowledge Imbalances on Wikidata / E. Giesemann. — 2020. — URL: https://blog.wikimedia.de/2020/09/16/prowd-detecting-knowledge-imbalances-on-wikidata/.

Joseph, A. The facts on file space and astronomy handbook / A. Joseph, J. Angelo. — Facts on file, 2009.

Knowledge Graphs / A. Hogan [et al.]. — Springer Cham, 2022. — (Synthesis Lectures on Data, Semantics, and Knowledge). — DOI: 10.2200/S01125ED1V01Y202109DSK022. — URL: https://kgbook.org/.

Piscopo, A. Structuring the world's knowledge: Socio-technical processes and data quality in Wikidata : PhD thesis / Piscopo Alessandro. — 2019. — DOI: 10.6084/m9.figshare.10998791.v2. — URL: https://figshare.com/articles/thesis/Structuring_the_world_s_knowledge_Socio-technical_processes_and_data_quality_in_Wikidata/10998791/2.

Samuel, J. Documenting Software Applications on Wikidata / J. Samuel. — 2020. — URL: https://jsamwrites.medium.com/documenting-software-applications-on-wikidata-197716d7f2d.

The network structure of scientific revolutions / H. Ju [et al.] // Computing Research Repository in arXiv (CoRR). — 2020. — arXiv: 2010.08381. — URL: https://arxiv.org/abs/2010.08381.

Velitchkov, I. Buckets and Balls // strategic structures / I. Velitchkov. — 07/2020. — URL: https://www.strategicstructures.com/?p=1889.

Vrandečić, D. Wikidata: A Free Collaborative Knowledgebase / D. Vrandečić, M. Krötzsch // Communications of the ACM. — 2014. — Vol. 57, no. 10. — P. 78–85. — ISSN 0001-0782. — DOI: 10.1145/2629489.

Wikidata: A large-scale collaborative ontological medical database / H. Turki [et al.] // Journal of Biomedical Informatics. — 2019. — Vol. 99. — P. 1–13. — ISSN 1532-0464. — DOI: https://doi.org/10.1016/j.jbi.2019.103292. — URL: https://www.sciencedirect.com/science/article/pii/S1532046419302114.

Ключевые слова и понятия

SPARQL
  [], 73
    / последовательность свойств, 128
    безымянная переменная, 57, 58, 62, 110, 122, 124
  AS, 74, 76
  BIND, 38, 78, 98, 102, 140, 143, 181
    IF, 38
  COUNT, 28, 31, 48, 56, 57, 59, 60, 73–78, 106, 110, 140
  DISTINCT, 38, 59, 60, 77, 143
  FILTER, 56, 62, 78, 87, 165
    !=, 99, 100
    <, 36
    >, 29
    EXISTS, 101, 168
    NOT EXISTS, 42, 73
    OR, 166
  FLOOR, 138, 140, 181
  GROUP BY, 31, 43, 49, 76, 77, 138, 140
    HAVING, 99
  GROUP_CONCAT, 62
  IF, 38, 78
  INCLUDE, 62
  Label
    rdfs:label, 125
    SERVICE, 125
    serviceParam, 125
  LANG, 87
  MAX, 71
  MINUS, 56, 87, 100, 101, 115
  NOT IN, 78
  OPTIONAL, 78, 108, 168
    FILTER, 89
  ORDER BY, 140
    ASC, 89
    DESC, 77, 89
    двойная сортировка, 106, 108
  p:[ps:]
    country, 134
    instance of, 85
    not only preferred rank, 169
    official language, 169
    operating system, 110
  Preferred value, 72, 83
  psn:
    нормализация площади, 164
  rdfs:label, 43
  SELECT
    WITH, 62
    вложенный, 36, 50, 57, 75, 95, 177
  SERVICE, 43, 136, 137, 140, 143
  STR, 71, 140
    REPLACE, 71
  SUM, 30, 49, 50, 71
  UNION, 73, 74, 80, 95, 102, 115, 131, 138, 140, 181
  Unknown value, 166
  VALUES, 58, 74, 77, 136, 143, 179
  xsd:integer, 71
  YEAR, 105, 140, 166, 181
  Анонимная переменная, 179
  Закон сохранения Викиданных, 131
  Квалификатор
    of, 138
  Квантификатор
    +, 179–181
  Поиск подклассов
    wdt:P31/wdt:P279*, 36, 165, 167
  Свойство
    country, 128
    country of registry, 131
    geoshape, 92
    main subject, 178
    operator, 128
    subclass of, 80

Wikidata Query Service, 16, 17
  Execute query, 13
  Help/Result Views, 106
  Image grid, 169
  LineChart, 75
  Timeline, 106

Библиотека
Pywikibot, 149–156, 158, 159
Визуализация данных
Rawgraphs, 34, 41
График
BarChart, 62, 64, 109, 139, 141
BubbleChart, 31, 77, 89, 118, 121, 123, 124
Graph, 39, 91, 143
Histogram, 45
ImageGrid, 165
LineChart, 37, 75, 76
Map, 78, 92, 102, 119, 122
Sunburst diagram, 34, 41
Timeline, 106
hide, 105, 106
Tree, 108
Информатика
База знаний
DBPedia, 17
Граф знаний, 20
Листинг, 15
Пермиссивная лицензия, 116
Расширение имени файла, 117
Семантический поиск, 17
Хештег, 126
Коэффициент Джини, 82, 127, 146
Обработка данных
OpenRefine, 149
QuickStatements, 149
Программирование
Библиотека
Matplotlib, 45
pandas, 45
Среда разработки
Colab, 42
Toolforge, 149
Язык
Python, 42