Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет геосистем и технологий»
(СГУГиТ)

А. А. Колесников, П. Ю. Бугаков

БАЗЫ ПРОСТРАНСТВЕННЫХ ДАННЫХ

Утверждено редакционно-издательским советом университета в качестве
учебно-методического пособия для обучающихся по направлению подготовки
05.03.03 Картография и геоинформатика

Новосибирск
СГУГиТ
2022

УДК 528.9
К603

Рецензенты: кандидат технических наук, доцент, СГУГиТ *Я. Г. Пошивайло*

кандидат технических наук, зав. кафедрой картографии и геоинформатики, СПбГУ *Е. А. Паниди*

Колесников, А. А.

К603 Базы пространственных данных : учебно-методическое пособие / А. А. Колесников, П. Ю. Бугаков. – Новосибирск : СГУГиТ, 2022. – 87 с. – Текст : непосредственный.

ISBN 978-5-907513-38-9

Учебно-методическое пособие подготовлено кандидатами технических наук, доцентами А. А. Колесниковым и П. Ю. Бугаковым на кафедре картографии и геоинформатики.

Пособие содержит теоретические основы и примеры использования баз пространственных данных в виде описания лабораторных работ, посвященных изучению способов интеграции данных различных форматов и построению запросов к ним в геоинформационной системе QGIS и системе управлениями базами данных PostgreSQL. Учебно-методическое пособие охватывает темы: «Базы данных», «Базы пространственных данных», «SQL-запросы», «Пространственный анализ», «Анализ и оптимизация SQL-запросов», изучаемые в рамках дисциплины «Базы пространственных данных».

Учебно-методическое пособие предназначено для обучающихся по направлению подготовки 05.03.03 «Картография и геоинформатика (уровень бакалавриата).

Рекомендовано к изданию кафедрой картографии и геоинформатики, Ученым советом Института геодезии и менеджмента СГУГиТ.

Печатается по решению редакционно-издательского совета СГУГиТ

УДК 528.9

ISBN 978-5-907513-38-9 © СГУГиТ, 2022

ОГЛАВЛЕНИЕ

Введение 5

1. Теоретические основы баз данных 6

2. Модели представления данных 8
   2.1. Основные модели данных 8
   2.2. Системы управления базами данных 12
   2.3. Основные функции СУБД 14

3. Языки запросов к данным 17
   3.1. Языки определения и манипулирования данными 17
   3.2. Типы SQL-запросов 17
   3.3. SQL-команды для управления структурой данных 18
   3.4. SQL-команды для работы с данными 18
   3.5. SQL-команды для управления правами доступа 21
   3.6. SQL-команды для операций с пространственными объектами 21

4. Лабораторная работа № 1. Интеграция данных различных типов в базу пространственных данных геоинформационной системы 24
   4.1. Открытие таблиц в ГИС QGIS 24
   4.2. Импорт данных в ГИС из Excel 26
   4.3. Импорт данных в ГИС из файлов *.csv 26
   4.4. Импорт данных в ГИС из SQLite 29
   4.5. Интеграция данных с картой 29
   4.6. Запросы на выборку данных 30

5. Лабораторная работа № 2. Анализ и визуализация данных в базе пространственных данных геоинформационной системы 35
   5.1. Составление запросов с агрегирующей функцией 35
   5.2. Построение и выполнение запросов с расчетом количественных характеристик 37
   5.3. Построение и выполнение пространственных запросов 40
   5.4. Построение картограммы 45

5.5. Построение картодиаграммы 47
5.6. Индивидуальные задания 49
6. Лабораторная работа № 3. Интеграция данных различных типов и составление аналитических запросов в базе пространственных данных реляционного типа 55
6.1. Создание базы данных. 56
6.2. Загрузка векторных данных в PostgreSQL 57
6.3. Импорт внешних таблиц в базу данных 63
6.4. Оценка производительности запроса и создание индексов 71
6.5. Индивидуальные задания 77
Заключение 85
Библиографический список 86

ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для обучающихся по направлению подготовки бакалавров 05.03.03 Картография и геоинформатика» изучающих дисциплину «Базы пространственных данных. Пособие содержит теоретический материал и описания лабораторных работ, последовательно рассматривающих процесс создания и использования баз пространственных данных, начиная от проектирования структуры базы данных до анализа и оптимизация формируемых SQL-запросов. В настоящем издании процессы создания и использования баз пространственных данных, построения запросов и пространственного анализа рассматриваются на примере географической информационной системы (ГИС) QGIS и системы управления базами данных (СУБД) PostgreSQL.

Описание лабораторных работ включает в себя цель, задачи, формулировку заданий, требования к содержанию отчетов и вопросы для подготовки к защите работы.

В издании приведен список рекомендуемой литературы, позволяющей обучающимся самостоятельно изучить дополнительный материал по базам пространственных данных, пространственным запросам и геоинформационному анализу.

После выполнения каждой лабораторной работы обучающийся должен предоставить отчет о проделанной работе в письменном или электронном виде, а также выполнить дополнительное задание, выдаваемое преподавателем, которое способствует расширению функциональности операций с базой данных, созданию новых запросов к данным, в том числе пространственным.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ БАЗ ДАННЫХ

Базовыми понятиями в теории баз данных являются данные и информация. Под информацией понимаются любые сведения о каком-либо событии, процессе, объекте, сформированные на основе определенного набора данных. Данные – это факты и значения определенных показателей, являющиеся источником для дальнейшего извлечения информации, представленные в определенном виде, позволяющем структурировать их сбор, хранение и дальнейшую обработку человеком или компьютерной системой. Для информационных технологий данные – это поддающееся многократной интерпретации представление информации в формализованном виде, пригодном для передачи, связи или обработки (ISO/IEC 2382:2015). Для удобства использования данных часто прибегают к созданию определенных структур, называющихся базами данных [1].

Можно выделить общие положения, характерные для любых баз данных:

− хранение данных происходит по определенным правилам, описанным в виде концепции или схемы данных;

− управление данными (добавление новых, изменение и удаление существующих, извлечение) выполняется по определенным наборам алгоритмов и механизмов;

− база данных создается для удовлетворения информационных потребностей в определенной предметной области.

Для реализации всех необходимых функций по использованию баз данных применяют специализированное программное обеспечение, называемое системами управления базами данных.

Система управления базами данных (СУБД) – совокупность языковых и программных средств, независимых от тематической области базы данных, предназначенных для создания и ведения баз данных, организации и контроля совместного использования данных многими пользователями, проверки ограничений и учета правил, накладываемых используемой моделью данных.

Основными компонентами СУБД являются:

− ядро, отвечающее за управление процессами, памятью, файловой системой, передачей данных по сети;

− диспетчер данных, отвечающий за реализацию механизмов транзакций и использование параметров ранее выполнявшихся запросов;

− диспетчер запросов, выполняющий синтаксический анализ текста поступающих запросов, оптимизирующий и осуществляющий план их выполнения;

− набор системных инструментов, отвечающий за выполнение служебных операций (резервное копирование, восстановление, мониторинг, ведение журналов большинства операций, выполняемых пользователями и самой СУБД) [2].

2. МОДЕЛИ ПРЕДСТАВЛЕНИЯ ДАННЫХ

2.1. Основные модели данных

Каждая база данных (БД) и СУБД строится на основе некоторой явной или неявной модели данных. Все СУБД, построенные на одной и той же модели данных, относят к одному типу.

Базовые требования к модели:

− модель должна быть достаточно универсальной, позволять работать с данными различной структуры, сложности и предметной области;

− модель должна допускать автоматическую обработку данных, то есть ее механизмы должны быть реализуемы программными средствами;

− модель должна быть наглядной и интуитивно понятной для обычных пользователей. Поскольку задача описания структуры данных средствами выбранной модели возлагается на разработчика (человека), то чем сложнее модель, тем труднее избежать ошибок при проектировании [3].

Также, в зависимости от модели, определяются требования к транзакционной системе СУБД, обеспечивающие наиболее надежную и предсказуемую работу.

Как правило, используется одно из двух требований: ACID (для реляционной модели) и BASE (для большинства остальных моделей).

ACID определяет следующие требования:

− атомарность (Atomicity) обеспечивает невозможность частичной фиксации транзакций. То есть либо будут выполнены все входящие в нее операции, либо не будет выполнено ни одной;

− согласованность (Consistency) отвечает за то, чтобы каждая успешная транзакция фиксировала только допустимые результаты, причем это относится к проверке не только корректности типов данных и доменов, но и каких-либо пользовательских правил;

− изолированность (Isolation) говорит о том, что в процессе выполнения транзакции другие, параллельные ей транзакции, не должны оказывать влияния на результат выполнения;

− прочность (Durability) отвечает за то, чтобы изменения, сделанные успешно завершенной транзакцией, не будут отменены из-за какого-либо программного или аппаратного сбоя [4, 5].

Ниже перечислены основные разновидности моделей представления данных, используемых в современных СУБД и реализующих функции хранения и обработки пространственных данных.

Документо-ориентированная модель (*document-based*). Документо-ориентированные СУБД служат для хранения иерархических структур данных (рис. 1.1). В данном контексте под документом понимается самостоятельная запись данных, содержащая все необходимое для его понимания значения, подобно документам, используемым в реальном мире. Такая модель находит свое применение в системах управления контентом, базах данных персоналий, поисковых системах и т. п.

Рис. 1.1. Документо-ориентированная модель

Графовая модель (*graph*). Эффективность работы с данными, обладающими большим количеством внутренних связей между элементами, является на сегодняшний день очень важным фактором для решения задач анализа, требующих обработки сложноструктурированных данных. Как одно из возможных решений сегодня используются графовые СУБД, ориентированные на явное отражение всего множества связей между элементами данных (рис. 1.2).

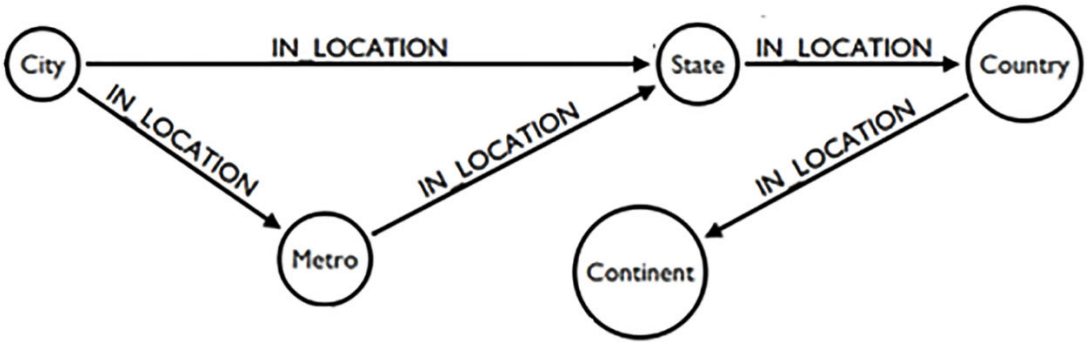


Рис. 1.2. Графовая модель

Реляционная модель (*relation*). В реляционной модели (рис. 1.3) данные представлены в виде таблиц, состоящих из строк и столбцов. Каждая строка таблицы описывает информацию об одном конкретном объекте, столбцы содержат свойства этого объекта. Таблица в целом представляет собой набор объектов, сгруппированных по какому-либо общему принципу. Взаимоотношения между объектами задаются с помощью связей между таблицами. Реляционная модель на сегодняшний день наиболее распространена. Она достаточно универсальна и проста в проектировании, а логико-математический аппарат хорошо исследован.

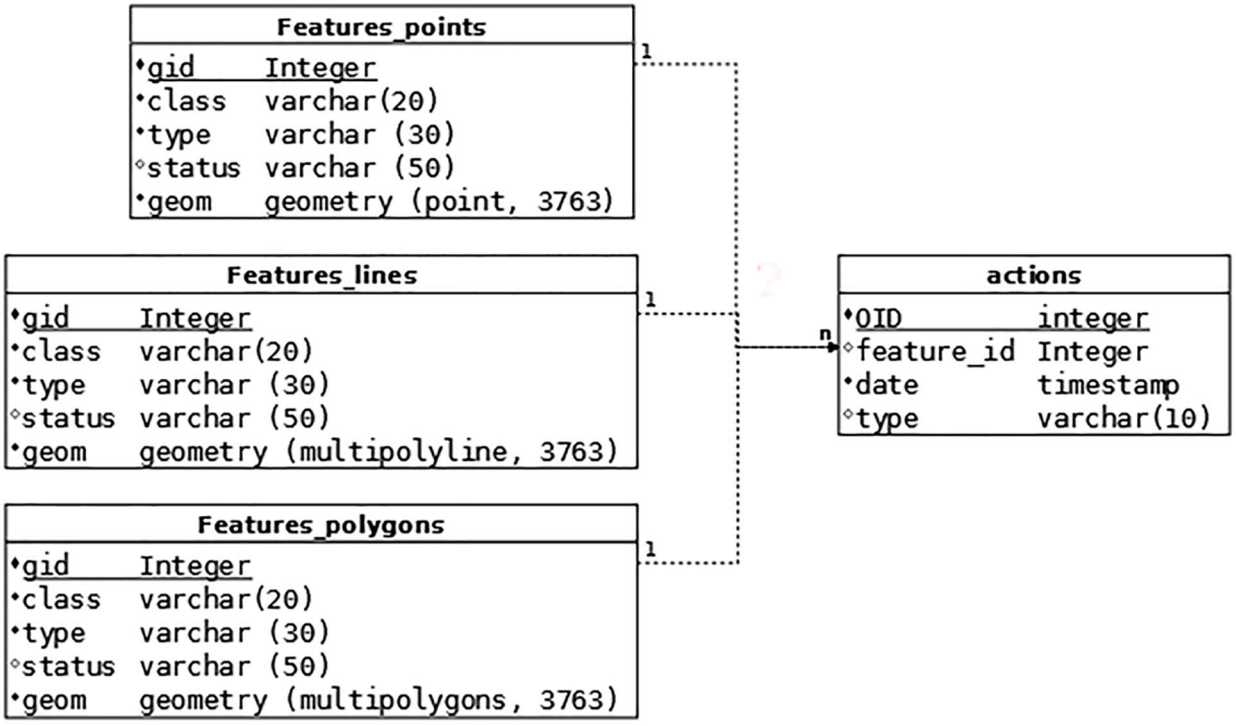


Рис. 1.3. Реляционная модель

Модель «ключ – значение» (*key – value*). Хранилище, построенное на основе модели «ключ – значение» является одним из наиболее простых вариантов структурированного хранения данных. Такая база данных хранит данные как совокупность пар «ключ – значение», в которых ключ служит уникальным идентификатором (рис. 1.4). Как ключи, так и значения могут представлять собой что угодно: от простых (число или несколько текстовых символов) до сложных составных объектов (растровые или векторные файлы карт). Базы данных с использованием этой модели данных хорошо поддерживают принципы распределенного хранения и позволяют эффективно использовать горизонтальное масштабирование, недостижимое при использовании других типов баз данных. Такие хранилища используются для хранения изображений, создания специализированных файловых систем, в качестве кэшей объектов, а также в системах, спроектированных с учетом дальнейшей масштабируемости.

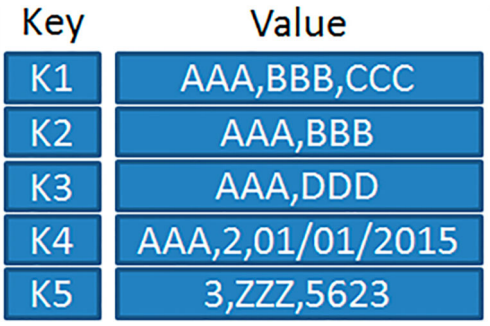


Рис. 1.4. Модель «ключ – значение»

Колоночная модель (column, wide-column). Основной принцип колоночных СУБД – хранение данных не по строкам, как это осуществляется в традиционных реляционных моделях, а по колонкам (рис. 1.5). Это означает, что с точки зрения SQL-клиента пользователя данные представлены как обычно в виде таблиц, но физически эти таблицы являются совокупностью колонок, каждая из которых представляет собой таблицу из одного поля и хранится отдельно. При колоночном хранении данных появляется возможность сильно компрессировать их, так как в одной колонке таблицы данные, как правило, однотипные, в отличие от строк. Колоночные СУБД призваны решить проблему неэффективной работы реляционных СУБД

в аналитических системах и системах, где превалируют операции чтения. Они позволяют на более дешевом и маломощном оборудовании получить прирост скорости выполнения запросов в 5–10 раз, при этом, благодаря компрессии, данные будут занимать на диске в несколько раз меньше места, чем в случае с реляционными СУБД.

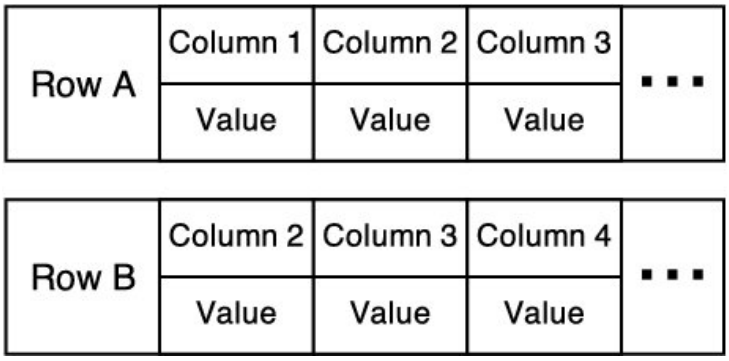


Рис. 1.5. Колоночная модель

Гибридная модель (*multimodel*). В некоторых современных СУБД реализовано смешение различных моделей представления данных (мультимодельные СУБД). Пример такого смешения – графово-реляционная модель. В ней использовано некоторое сходство между реляционной и графовой идеологиями. Узлы графовой модели соответствуют объектам, таблицы – атрибутам этих объектов. Отсюда вытекает возможность введения наследования при определении таблиц – таблица-потомок содержит те же столбцы, что и родительская, и, кроме того, – дополнительные, определенные при наследовании [6, 7].

2.2. Системы управления базами данных

СУБД – это программное обеспечение, с помощью которого пользователи могут создавать и управлять базами данных, осуществлять к ним контролируемый доступ и выполнять администрирование всего комплекса программных модулей, входящих в его состав.

По степени универсальности различаются два класса СУБД – системы общего назначения и специализированные системы.

СУБД общего назначения не ориентированы на какую-либо конкретную предметную область или на информационные потребности узкой группы пользователей. СУБД общего назначения обладает достаточно большим количеством инструментов для настройки работы конкретной БД в условиях решения определенной задачи.

В некоторых ситуациях СУБД общего назначения не позволяют добиться требуемых проектных и эксплуатационных характеристик (производительность, занимаемый объем памяти и прочее). Тем не менее создание специализированных СУБД – весьма трудоемкий процесс, и для того, чтобы его реализовать, нужны очень весомые причины.

В процессе функционирования СУБД постоянно взаимодействует с базами данных, находящимися под ее управлением, и реализует доступ к данным сторонних прикладных программных продуктов на основе реализованных интерфейсов и указанных прав доступа.

Для того чтобы СУБД успешно справлялась со своими задачами, она должна обладать следующими возможностями:

− включать язык определения данных, с помощью которого задается база данных, ее структура, типы данных и средства задания ограничений (домены) для хранимой информации. Все вышеперечисленное сейчас подразумевается и в многопользовательском варианте, то есть формирование множеств баз данных, пользователей, различные уровни безопасности данных и многое другое;

− позволять вставлять, удалять, обновлять и извлекать информацию из базы данных посредством языка управления данными;

− поддерживать различные архитектуры компьютеров и операционных систем. При этом тип платформы на работу пользователя при доступе к данным влияния обычно не оказывает;

− обеспечивать развитые средства администрирования БД.

С точки зрения безопасности типовая СУБД обладает следующими подсистемами:

1) подсистема обеспечения безопасности, предотвращающая несанкционированный доступ к информации, содержащейся в базах данных;

2) подсистема поддержки целостности базы данных, обеспечивающая непротиворечивое состояние хранимых данных;

3) подсистема управления параллельной работой транзакций, контролирующая процессы их совместного доступа к базе данных;

4) подсистема работы с резервными копиями, позволяющая восстановить базу данных до предыдущего непротиворечивого состояния, нарушенного в результате аппаратного или программного сбоя.

2.3. Основные функции СУБД

Управление данными в оперативной и долговременной памяти. Данная функция предоставляет пользователям возможность выполнения самых основных операций, которые осуществляются с данными на том или ином носителе, – сохранение, извлечение и обновление информации. Она включает в себя управление необходимыми структурами памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для ускорения доступа к данным.

Управление транзакциями. Транзакция – это последовательность операций над базой данных (доступ или изменение содержимого), рассматриваемых СУБД как единое целое. Особенность этой последовательности в том, что если хотя бы одна из операций не выполняется, то происходит откат всех операций, выполненных до этого. Примерами простых транзакций может служить добавление, обновление или удаление в базе данных сведений о некоем объекте (изменение строки). Сложная же транзакция образуется в том случае, когда в базу данных требуется внести сразу несколько изменений (особенно если эти изменения затрагивают несколько таблиц). Инициализация транзакции может быть вызвана как вручную пользователем, так и прикладной программой.

Резервное копирование и восстановление баз данных. Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида аппаратных сбоев: мягкие, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания), и жесткие, характеризуемые потерей информации на носите-

лях внешней памяти. Поддержание надежности хранения данных, как правило, основывается на избыточности хранения данных, причем та часть данных, которая используется для восстановления, должна храниться особо надежно. Наиболее распространенным методом реализации такого способа является ведение журнала изменений БД.

Поддержка языков запросов. Для работы с базами данных используются специальные декларативные языки, называемые языками запросов. В современных реляционных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Им является язык SQL (Structured Query Language – язык структурированных запросов). Язык SQL позволяет определять схему реляционной БД и манипулировать данными. В СУБД, функционирующих на основе других моделей данных, используются альтернативные языки запросов (например, Cypher для графовых БД или языки запросов на базе JSON для документо-ориентированных БД). Также нужно упомянуть, что распространенность SQL настолько высока, что большинство NoSQL СУБД имеют собственные или сторонние трансляторы из SQL в «нативный» язык запросов.

Управление параллельным доступом. Одна из основных целей создания и использования СУБД заключается в том, чтобы множество пользователей могло осуществлять параллельный доступ к совместно обрабатываемым данным. Параллельный доступ сравнительно просто организовать, если все пользователи выполняют только чтение данных, поскольку в этом случае они не могут помешать друг другу. Однако, когда двое или больше пользователей одновременно получают доступ к базе данных, конфликт с нежелательными последствиями легко может возникнуть, например, если хотя бы один из них попытается обновить данные. СУБД должна гарантировать, что при одновременном доступе к базе данных многих пользователей подобных конфликтов не произойдет. В современных СУБД эти механизмы реализуются на основе принципов изоляции транзакций.

Контроль доступа к данным. СУБД должна иметь механизмы, гарантирующие возможность организации доступа к базе данных только санкционированных пользователей и защищающие ее от любого несанкциониро-

ванного доступа или изменения. В современных СУБД поддерживается один из двух широко распространенных подходов к вопросу обеспечения безопасности данных: избирательный подход или обязательный подход. В большинстве современных систем предусматривается избирательный подход, при котором некий пользователь обладает различными правами при работе с разными объектами. Значительно реже применяется альтернативный, обязательный подход, где каждому объекту данных присваивается некоторый классификационный уровень, а каждый пользователь обладает некоторым уровнем допуска. Сегодня СУБД позволяют очень гибко настраивать уровни доступа, вплоть до отдельных ячеек данных в таблицах.

Поддержка целостности данных. Термин «целостность» используется для описания корректности и непротиворечивости, хранимых в БД. Реализация поддержки целостности данных предполагает, что СУБД должна уметь обрабатывать определенный свод правил, которые нельзя нарушать при работе с данными, и обладать инструментами контроля за тем, чтобы им соответствовали сами данные и операции по изменению.

3. ЯЗЫКИ ЗАПРОСОВ К ДАННЫМ

3.1. Языки определения и манипулирования данными

В СУБД поддерживается несколько специализированных по своим функциям подъязыков. Их можно разбить на две категории:

− язык определения данных БД;

− язык манипулирования данными.

Язык определения данных – описательный язык, с помощью которого описывается предметная область: именуются объекты, определяются их свойства и связи между ними. Он используется главным образом для определения логической структуры БД. Схема базы данных, выраженная в терминах специального языка определения данных, состоит из набора определений. Язык определения данных используется как для определения новой схемы, так и для модификации уже существующей. Результатом выполнения операторов языка определения данных является набор структур (таблиц), хранимый в системном каталоге, содержащий метаданные. Перед доступом к реальным данным СУБД обычно обращается к системному каталогу.

Язык манипулирования данными содержит набор операторов манипулирования, позволяющий добавлять новые данные, удалять, модифицировать или выбирать существующие.

3.2. Типы SQL-запросов

SQL-запросы чаще всего делятся на следующие три категории:

− запросы, предназначенные для работы со структурой данных (Data Definition Language – DDL), то есть для создания, описания и модификации структуры базы данных и ее элементов;

− запросы, используемые непосредственно в работе с данными (Data Manipulation Language – DML), с помощью которых можно добавлять, обновлять и удалять данные;

− запросы, применяемые для предоставления или отмены прав доступа к БД (Data Control Language – DCL).

3.3. SQL-команды для управления структурой данных

К этой категории относятся команды, работающие со структурой БД. К ним относятся CREATE – «создать» (например, CREATE TABLE (создать таблицу), CREATE USER (создать пользователя), ALTER – «модифицировать» (этот запрос используется при внесении изменений в саму БД или в ее часть), DROP – «удалить» (также относится к БД и ее частям).

Например, синтаксис SQL-команды для создания новой таблицы (CREATE TABLE):

CREATE TABLE <table_name1> (
<col_name1><col_type1>,
<col_name2><col_type2>,
<col_name3><col_type3>
PRIMARY KEY(<col_name1>),
FOREIGN KEY(<col_name2>)
REFERENCES <table_name2> (<col_name2>));

При создании таблиц обычно сразу задаются ограничения целостности данных для определенных столбцов в таблице. Чаще всего применяются следующие ограничения:

− NOT NULL – ячейка таблицы не может иметь значение NULL;

− PRIMARY KEY(col_name1, col_name2, …) – первичный ключ;

− FOREIGN KEY (col_name_1, …, col_name_n) REFERENCES table_name(col_name_1, …, col_name_n) – внешний ключ. SQL-синтаксис позволяет задавать более одного первичного ключа. В таком случае формируется составной первичный ключ.

3.4. SQL-команды для работы с данными

К наиболее востребованным командам работы с данными относятся: SELECT (выборка данных), INSERT (вставка новых данных), UPDATE (обновление данных), DELETE (удаление данных), MERGE (слияние данных).

Общая структура запроса на выборку (SELECT) данных выглядит следующим образом (табл. 3.1):

SELECT < col_name1, col_name2, ...>
FROM <table_name1, table_name2, ...>
WHERE <condition1> AND/OR/NOT <condition2> AND/OR/NOT ...
GROUP BY < col_name1, col_name2, ...>
HAVING <gropu_condition1> AND/OR/NOT ...
ORDER BY < col_name1, col_name2, ...>
LIMIT <int_digit>;

Таблица 3.1

Элементы SQL-запроса

Ключевое слово	Параметры	Обязательность
SELECT	Список столбцов или * для выбора всех столбцов	Да
FROM	Список таблиц	Да
WHERE	Набор условий, объединенных логическими операторами	Нет
GROUP BY	Список столбцов для группировки результатов выборки с помощью агрегирующих функций	Нет
HAVING	Набор условий, объединенных логическими операторами для сгруппированных данных	Нет (используется только совместно с GROUP BY)
ORDER BY	Список столбцов, по которым сортируются результаты запроса	Нет
LIMIT	Ограничение количества записей, выводимых в результате запроса	Нет

Команда INSERT INTO <table_name> в SQL отвечает за добавление данных в таблицу:

INSERT INTO <table_name> (<col_name1>, <col_name2>, <col_name3>, ...) VALUES (<value1>, <value2>, <value3>, ...);

При добавлении данных в каждый столбец таблицы можно не указывать названия столбцов.

INSERT INTO <table_name> VALUES (<value1>, <value2>, <value3>, ...);

UPDATE – это SQL-команда для обновления данных в таблице:

UPDATE <table_name> SET <col_name1> = <value1>,
<col_name2> = <value2>, ... WHERE <condition>;

SQL-команда DELETE FROM используется для удаления данных из таблицы:

DELETE FROM <table_name> WHERE <condition>;

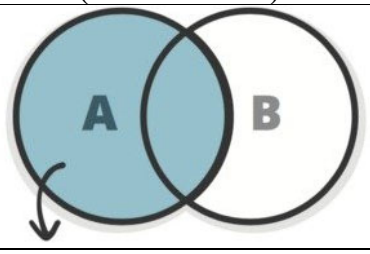
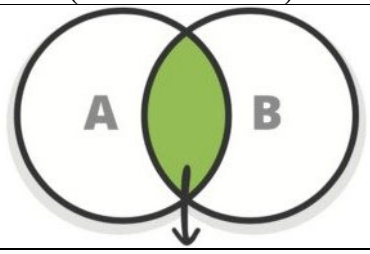
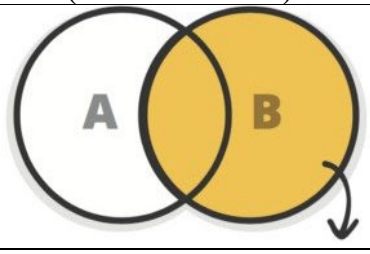
Также для связанных запросов (с помощью общих атрибутов) к нескольким таблицам используется оператор JOIN. Общий синтаксис запроса приведен далее:

SELECT <col_name1>, <col_name2>, ...
FROM <table_name1>
JOIN <table_name2>
ON <table_name1.col_name> = <table2.col_name>;

В табл. 3.2 ниже показаны различные способы объединения в SQL.

Таблица 3.2

Схемы запросов JOIN

Левостороннее объединение (LEFT JOIN).	Внутреннее объединение (INNER JOIN).	Правостороннее объединение (RIGHT JOIN).
		
Результатом будут все строки из таблицы *A* и те строки из таблицы *B*, которые имеют те же значения, что и в таблице *A* для выбранных столбцов	Результатом будут только те строки из таблицы *A* и *B*, которые имеют одинаковые значения для выбранных столбцов	Результатом будут все строки из таблицы *B* и те строки из таблицы *A*, которые имеют те же значения, что и в таблице *B* для выбранных столбцов

3.5. SQL-команды для управления правами доступа

К командам, работающим с правами доступа, относятся GRANT – разрешение пользователю на проведение определенных операций с БД или данными, REVOKE – отзыв выданного разрешения, DENY – установка запрета, имеющего приоритет над разрешением.

Общий синтаксис команды GRANT:

− privilege_name – это список команд, разрешение на выполнение которых выдается пользователю (например, SELECT, INSERT);

− object_name – это имя объекта базы данных, на который выдаются разрешения (например, таблица или хранимая процедура);

− user_name – это имя пользователя, которому выдаются разрешения (может быть указано несколько пользователей одновременно, через запятую);

− role_name – указывается вместо конкретных пользователей и означает группу пользователей объединенных по какому-то критерию.

GRANT privilege_name
ON object_name
TO {user_name |PUBLIC |role_name}
[WITH GRANT OPTION];

Общий синтаксис команды для удаления разрешений:

REVOKE privilege_name
ON object_name
FROM {user_name |PUBLIC |role_name}

3.6. SQL-команды для операций с пространственными объектами

Для получения характеристик (например, количество узлов, площадь, периметр и т. д.) и выполнения операций с пространственными объектами СУБД предоставляет дополнительные команды, которые могут быть использованы в стандартных SQL-запросах.

Расширения СУБД для работы с пространственными данными добавляют дополнительные типы данных: точки, линии, полигоны, мультиобъекты, поверхности, геопривязанные растры.

Рассмотрим синтаксис наиболее популярных команд, используемых в SQL-запросах, при работе с пространственными данными.

Проверка пересечения (ST_Intersects), содержания (ST_Contains) или входимости (ST_Within) двух наборов пространственных объектов:

ST_Intersection (или ST_Contains, или ST_Within) (поле геометрии набора объектов №1, поле геометрии набора объектов №2);

Вычисление длины линейных объектов:

ST_Length(поле геометрии);

Вычисление периметра полигональных объектов:

ST_Perimeter(поле геометрии);

Вычисление площади объекта:

ST_Area(поле геометрии);

Вычисление центроида объекта:

ST_Centroid(поле геометрии);

Вычисление расстояния между ближайшими точками объектов:

ST_Distance(поле геометрии набора объектов №1, поле геометрии набора объектов №2);

Сшивка пространственных объектов:

ST_Union(поле геометрии набора объектов №1, поле геометрии набора объектов №2);

Построение буферной зоны для одного или более пространственных объектов:

ST_Buffer(поле геометрии, радиус буферной зоны в единицах измерения системы координат таблицы данных, количество сегментов буферной зоны);

При использовании эти операторы задействуются непосредственно в тексте запроса. Например, далее представлен текст запроса для построения минимально-описывающих полигонов вокруг каждого из наборов объектов таблицы:

```
SELECT line, ST_ConvexHull(ST_Union(geom)) AS hull
FROM s
GROUP BY line;
```

4. ЛАБОРАТОРНАЯ РАБОТА № 1. ИНТЕГРАЦИЯ ДАННЫХ РАЗЛИЧНЫХ ТИПОВ В БАЗУ ПРОСТРАНСТВЕННЫХ ДАННЫХ ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ

Время выполнения – 2 часа.

Цель работы: усвоить основные принципы интеграции данных различных форматов, основные типы запросов к данным, освоить принципы связывания таблиц и фильтрации данных инструментами геоинформационной системы.

Задачи работы

1. Изучить:
– распространенные форматы данных для геоинформационных систем;
– алгоритм выполнения запросов к данным.
2. Научиться:
– работать с основными форматами данных геоинформационных систем;
– выполнять связывание данных из разных таблиц;
– выполнять пространственные запросы.

Перечень обеспечивающих средств

Для выполнения работы необходимо иметь методические указания по выполнению работы, а также программное обеспечение QGIS (версия 3.10 и выше).

4.1. Открытие таблиц в ГИС QGIS

Набор данных для выполнения заданий представляет собой пространственные данные о странах и городах мира, а также различную статистическую информацию. Исходные данные представлены в форматах, которыми может оперировать ГИС QGIS.

Исходные данные о границах стран мира в виде площадных объектов представлены в стандартном формате хранения данных ГИС MapInfo – TAB.

Для открытия карты с административно-территориальным делением стран мира (Countries) нужно в меню *Слой* выбрать пункты *Добавить слой – Векторный слой*. В открывшемся диалоге указать файл Countries.TAB и нажать кнопку *Добавить*. В результате будет открыто окно карты с площадными объектами (рис. 4.1). Семантическая информация об объектах представлена в таблице (правой кнопкой мыши нажать по слою и выбрать пункт меню *Открыть таблицу атрибутов*) (рис. 4.2).

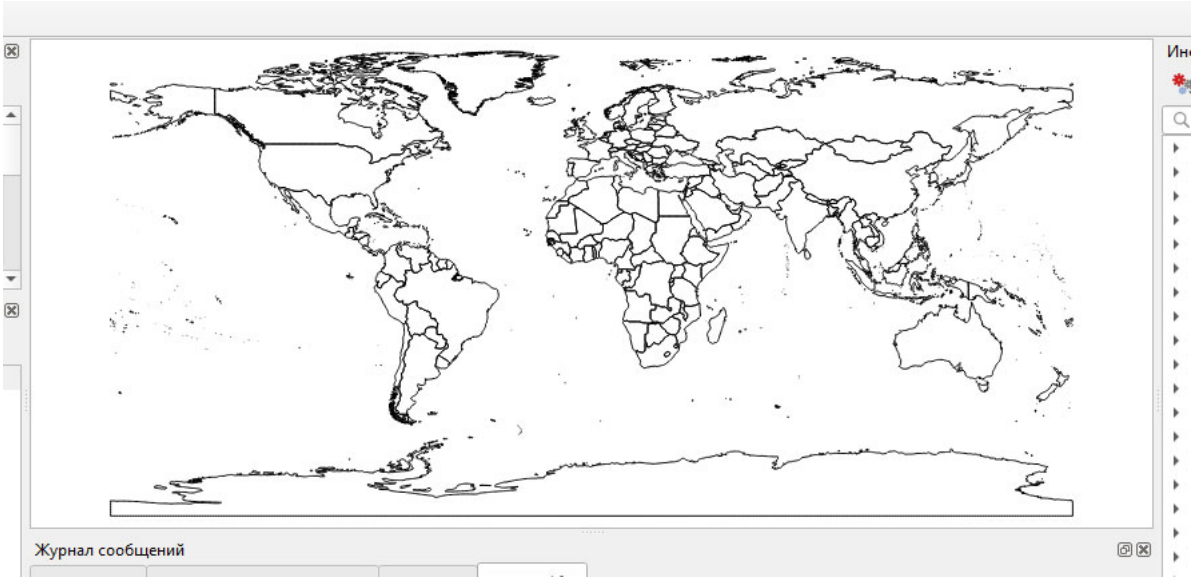


Рис. 4.1. Исходная политико-административная карта мира

Countries - Всего объектов: 246, Отфильтровано: 246, Выделено: 0

	FIPS	ISO2	ISO3	UN	NAME	AREA	POP2005	REGION	SUBREGION	LON
1	AC	AG	ATG	28	Antigua and Ba...	44	83039	19	29	-61,783
2	AG	DZ	DZA	12	Algeria	238174	32854159	2	15	2,632
3	AJ	AZ	AZE	31	Azerbaijan	8260	8352021	142	145	47,395
4	AL	AL	ALB	8	Albania	2740	3153731	150	39	20,068
5	AM	AM	ARM	51	Armenia	2820	3017661	142	145	44,563
6	AO	AO	AGO	24	Angola	124670	16095214	2	17	17,544
7	AQ	AS	ASM	16	American Samoa	20	64051	9	61	-170,730
8	AR	AR	ARG	32	Argentina	273669	38747148	19	5	-65,167
9	AS	AU	AUS	36	Australia	768230	20310208	9	53	136,189
10	BA	BH	BHR	48	Bahrain	71	724788	142	145	50,562
11	BB	BB	BRB	52	Barbados	43	291933	19	29	-59,559
12	BD	BM	BMU	60	Bermuda	5	64174	19	21	-64,709
13	BF	BS	BHS	44	Bahamas	1001	323295	19	29	-78,014
14	BG	BD	BGD	50	Bangladesh	13017	15328112	142	34	89,941
15	BH	BZ	BLZ	84	Belize	2281	275546	19	13	-88,602
16	BK	BA	BIH	70	Bosnia and Her...	5120	3915238	150	39	17,786

BHR

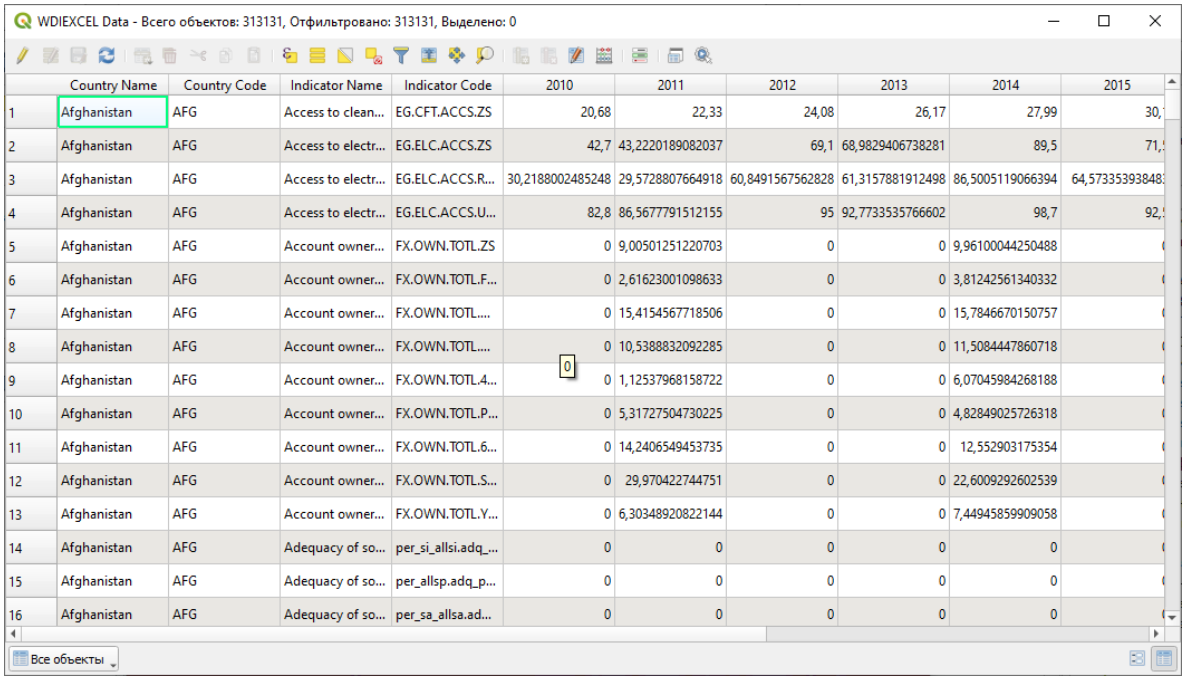
Все объекты

Рис. 4.2. Семантические данные политико-административной карты мира

4.2. Импорт данных в ГИС из Excel

Одним из распространенных источников статистической информации являются файлы Microsoft Excel. Их добавление в виде таблиц с данными также осуществляется через меню *Добавить слой – Векторный слой* и указания типа файла *Файлы Excel* (*.xls, *.xlsx). Импорт выполняется на примере файла *WDI.xlsx* (лист *Data*), содержащего ежегодные значения различных социально-экономических индикаторов стран мира.

При открытии файлов Excel выводится диалог, в котором можно указать один или несколько листов с интересующими данными. В результате будет получена таблица (рис. 4.3).



	Country Name	Country Code	Indicator Name	Indicator Code	2010	2011	2012	2013	2014	2015
1	Afghanistan	AFG	Access to clean...	EG.CFT.ACCS.ZS	20,68	22,33	24,08	26,17	27,99	30,
2	Afghanistan	AFG	Access to electr...	EG.ELC.ACCS.ZS	42,7	43,2220189082037	69,1	68,9829406738281	89,5	71,
3	Afghanistan	AFG	Access to electr...	EG.ELC.ACCS.R...	30,2188002485248	29,5728807664918	60,8491567562828	61,3157881912498	86,5005119066394	64,573353938
4	Afghanistan	AFG	Access to electr...	EG.ELC.ACCS.U...	82,8	86,5677791512155	95	92,7733535766602	98,7	92,
5	Afghanistan	AFG	Account owner...	FX.OWN.TOTL.ZS	0	9,00501251220703	0	0	9,96100044250488	
6	Afghanistan	AFG	Account owner...	FX.OWN.TOTL.F...	0	2,61623001098633	0	0	3,81242561340332	
7	Afghanistan	AFG	Account owner...	FX.OWN.TOTL....	0	15,4154567718506	0	0	15,7846670150757	
8	Afghanistan	AFG	Account owner...	FX.OWN.TOTL....	0	10,5388832092285	0	0	11,5084447860718	
9	Afghanistan	AFG	Account owner...	FX.OWN.TOTL.4...	0	1,12537968158722	0	0	6,07045984268188	
10	Afghanistan	AFG	Account owner...	FX.OWN.TOTL.P...	0	5,31727504730225	0	0	4,82849025726318	
11	Afghanistan	AFG	Account owner...	FX.OWN.TOTL.6...	0	14,2406549453735	0	0	12,552903175354	
12	Afghanistan	AFG	Account owner...	FX.OWN.TOTL.S...	0	29,970422744751	0	0	22,6009292602539	
13	Afghanistan	AFG	Account owner...	FX.OWN.TOTL.Y...	0	6,30348920822144	0	0	7,44945859909058	
14	Afghanistan	AFG	Adequacy of so...	per_si_allsi.adq_...	0	0	0	0	0	
15	Afghanistan	AFG	Adequacy of so...	per_allsp.adq_p...	0	0	0	0	0	
16	Afghanistan	AFG	Adequacy of so...	per_sa_allsa.ad...	0	0	0	0	0	

Рис. 4.3. Просмотр импортированных данных из файла формата Excel

4.3. Импорт данных в ГИС из файлов *.csv

Формат *.csv сейчас является самым распространенным способом представления статистических данных. Для примера загрузим в QGIS csv-файл Countries_of_the_world, содержащий краткую сводную информацию по странам мира на текущий год. Для этого в меню *Слой* нужно выбрать пункты меню *Добавить слой – Добавить слой их текста с разделителями*. После выбора файла будет выдан диалог, позволяющий указать раздели-

тель полей (если он отличается от стандартной запятой), кодировку, названия и типы данных колонок (рис. 4.4). Результат добавления таблицы в формате *.csv представлен на рис. 4.5.

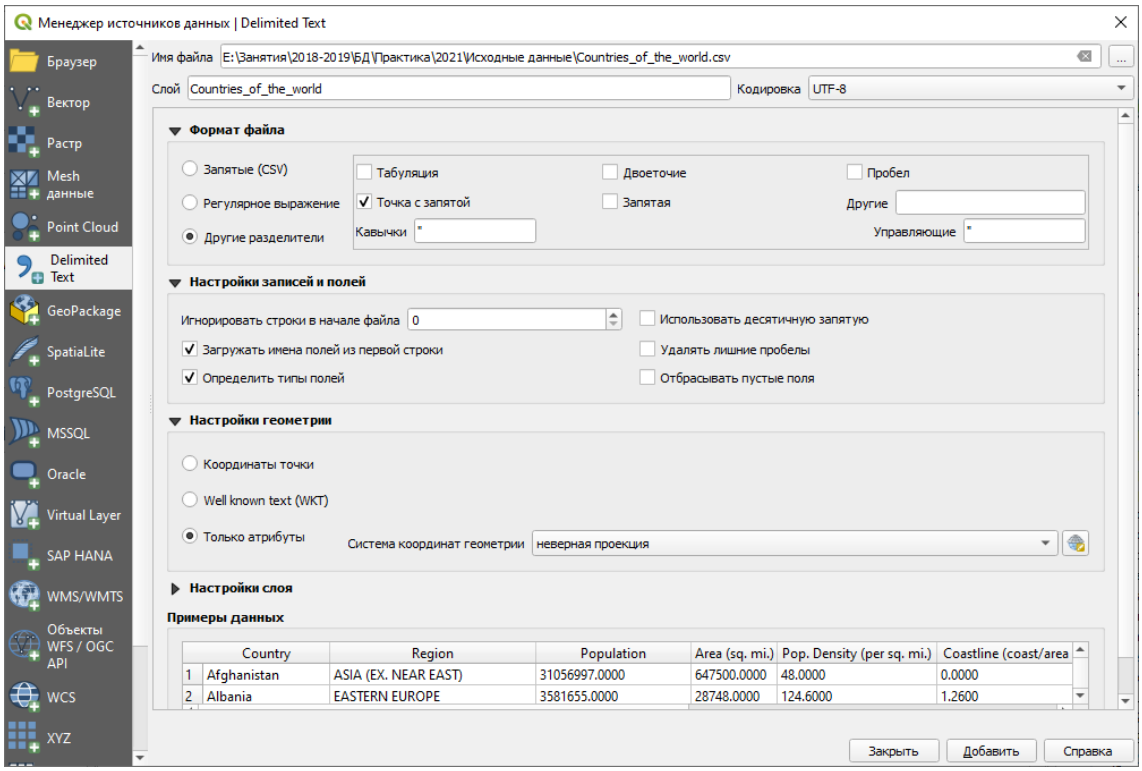
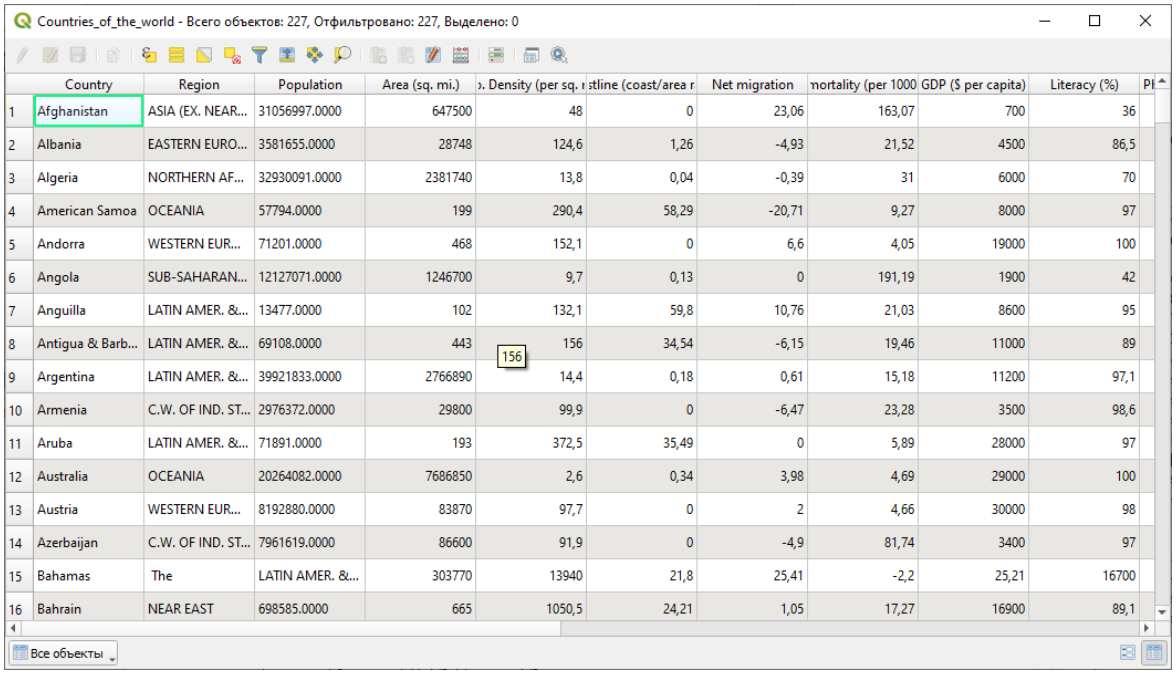


Рис. 4.4. Настройка параметров импорта csv-файла



	Country	Region	Population	Area (sq. mi.)	Density (per sq.	tline (coast/area r	Net migration	ortality (per 1000	GDP ($ per capita)	Literacy (%)	Pl
1	Afghanistan	ASIA (EX. NEAR...	31056997.0000	647500	48	0	23,06	163,07	700	36	
2	Albania	EASTERN EURO...	3581655.0000	28748	124,6	1,26	-4,93	21,52	4500	86,5	
3	Algeria	NORTHERN AF...	32930091.0000	2381740	13,8	0,04	-0,39	31	6000	70	
4	American Samoa	OCEANIA	57794.0000	199	290,4	58,29	-20,71	9,27	8000	97	
5	Andorra	WESTERN EUR...	71201.0000	468	152,1	0	6,6	4,05	19000	100	
6	Angola	SUB-SAHARAN...	12127071.0000	1246700	9,7	0,13	0	191,19	1900	42	
7	Anguilla	LATIN AMER. &...	13477.0000	102	132,1	59,8	10,76	21,03	8600	95	
8	Antigua & Barb...	LATIN AMER. &...	69108.0000	443	156	34,54	-6,15	19,46	11000	89	
9	Argentina	LATIN AMER. &...	39921833.0000	2766890	14,4	0,18	0,61	15,18	11200	97,1	
10	Armenia	C.W. OF IND. ST...	2976372.0000	29800	99,9	0	-6,47	23,28	3500	98,6	
11	Aruba	LATIN AMER. &...	71891.0000	193	372,5	35,49	0	5,89	28000	97	
12	Australia	OCEANIA	20264082.0000	7686850	2,6	0,34	3,98	4,69	29000	100	
13	Austria	WESTERN EUR...	8192880.0000	83870	97,7	0	2	4,66	30000	98	
14	Azerbaijan	C.W. OF IND. ST...	7961619.0000	86600	91,9	0	-4,9	81,74	3400	97	
15	Bahamas	The	LATIN AMER. &...	303770	13940	21,8	25,41	-2,2	25,21	16700	
16	Bahrain	NEAR EAST	698585.0000	665	1050,5	24,21	1,05	17,27	16900	89,1	

Рис. 4.5. Просмотр импортированных данных из csv-файла

Текстовые файлы могут содержать не только семантические данные, но и пространственные. В случае csv-файлов это, как правило, пары координат точечных объектов. QGIS позволяет создавать точечные объекты в импортированных таблицах на основе значений координат из определенных колонок. Рассмотрим порядок действий на примере файла, содержащего координаты и семантическую информацию наиболее крупных городов мира. Выберем для открытия csv-файл Cities.

В диалоге импорта файла с данными о городах нужно установить следующие параметры: разделитель – запятая, загружать имена полей из первой строки, координаты *X* и *Y* объектов из полей (с помощью выпадающих списков), содержащих значения широты и долготы, система координат – «WGS 84» (рис. 4.6).

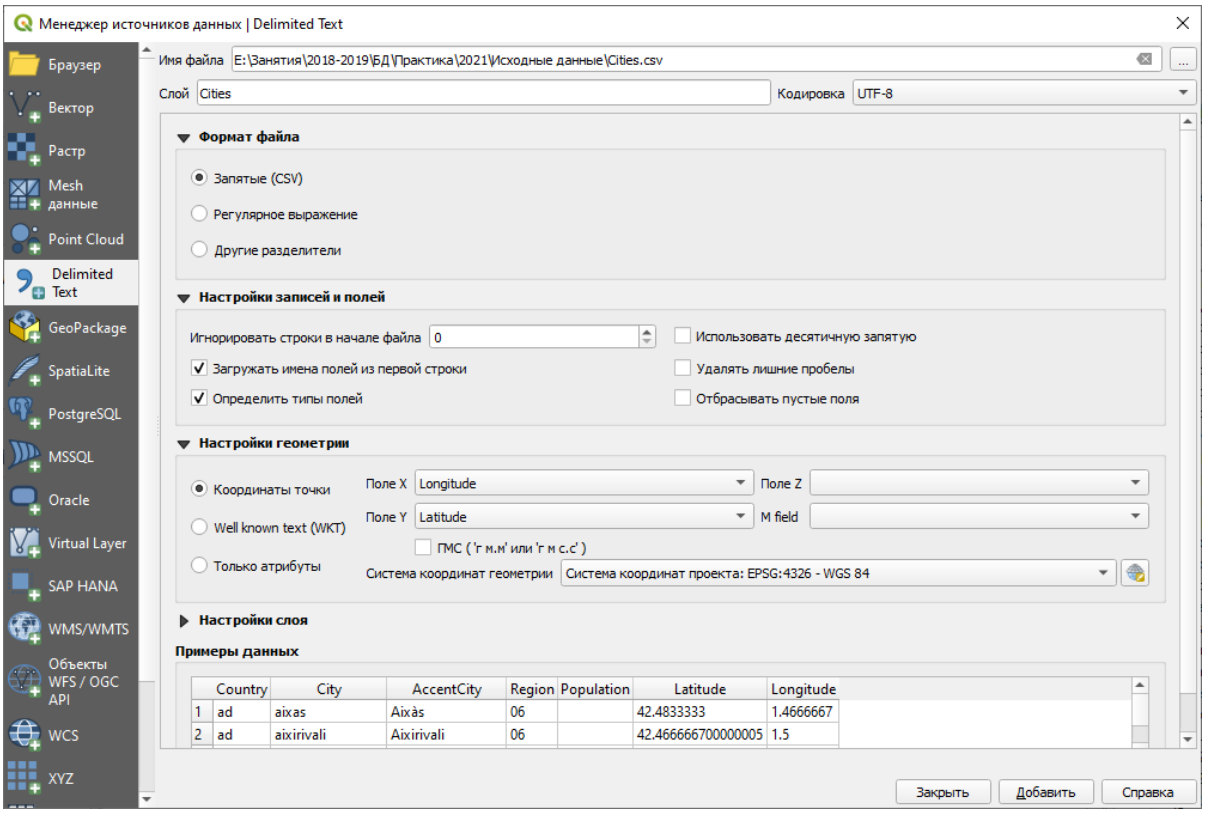


Рис. 4.6. Настройка параметров импорта csv-файла
с данными о городах мира

4.4. Импорт данных в ГИС из SQLite

ГИС QGIS позволяет работать с наиболее распространенными файловыми базами данных и сетевыми СУБД, в том числе поддерживающими хранение и анализ пространственных данных. Для примера добавим информацию из файла открытого формата файловых баз данных – SQLite, содержащего статистические данные о демографии стран. В этом диалоге рекомендуется установить параметр *LIST_ALL_TABLE* в значение *Да*, чтобы получить список возможных к загрузке таблиц. В этом списке нужно выделить только таблицу с названием *Data*, остальные являются служебными (рис. 4.7).

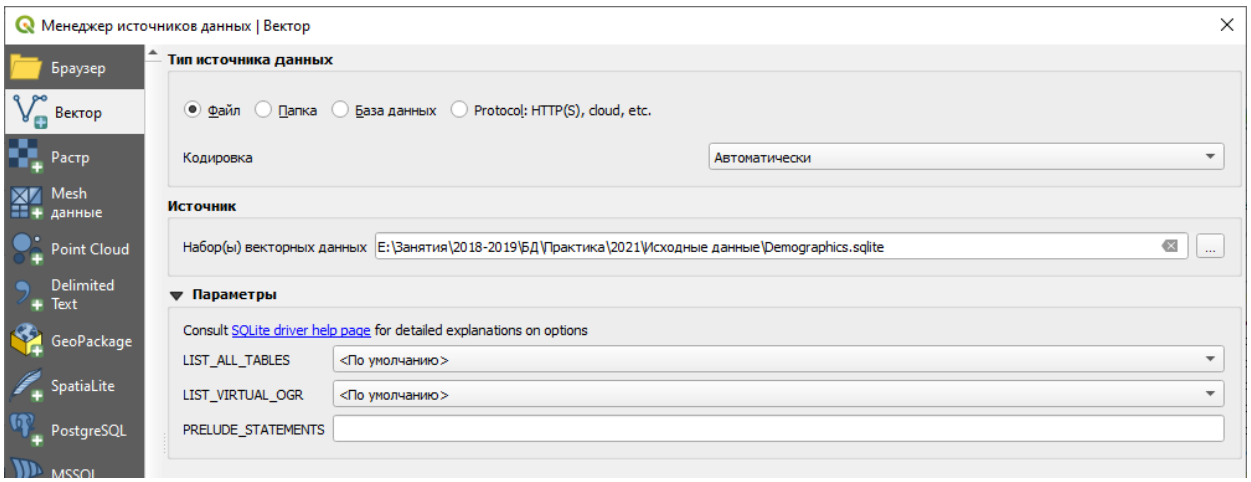


Рис. 4.7. Диалог настройки параметров импорта данных
из файла формата SQLite

4.5. Интеграция данных с картой

Поскольку пространственные данные в виде контуров стран мира и статистические данные из импортированных таблиц представляют собой различные источники, то практически всегда возникает необходимость объединения данных. Для осуществления этой операции необходимым является наличие в обеих таблицах колонок с одинаковыми данными; в случае представленных файлов такими колонками будут названия стран или их ISO-коды.

Для объединения таблиц по условию в QGIS используется механизм связей. Параметры связывания таблиц устанавливаются в свойствах слоя,

раздел *Связи*. На рисунке ниже представлен пример запроса, объединяющего таблицы *Countries* и *Demographics* с условием идентичности трехсимвольных кодов стран (рис. 4.8).

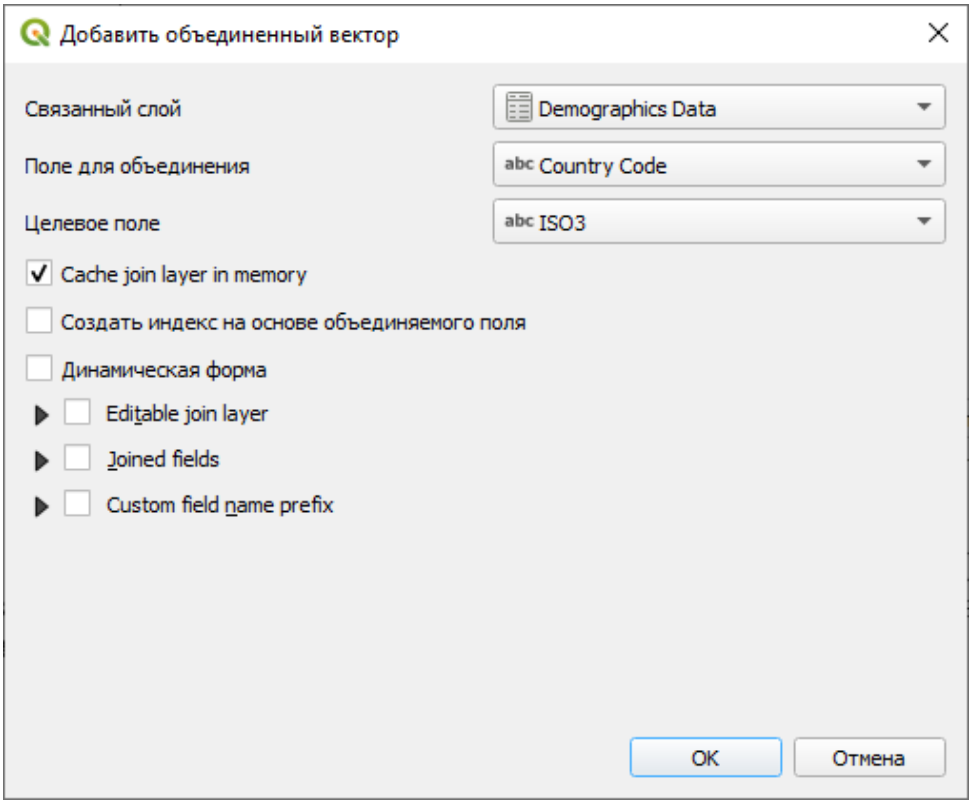


Рис. 4.8. Параметры связывания таблиц

В результате выполнения запроса на объединение таблиц будет выведена таблица, содержащая колонки из обеих таблиц исходных данных.

4.6. Запросы на выборку данных

Поскольку исходная таблица с городами содержит слишком много записей, то демонстрацию работы с запросами на выборку данных выполним на примере отбора населенных пунктов, лежащих на территории Российской Федерации с указанным (ненулевым) населением. Затем результаты этой выборки нужно будет выгрузить в отдельный файл.

Составим запрос на выборку всех населенных пунктов с ненулевым населением. Конструктор запросов вызывается кнопкой *Выделить объекты удо-*

влетворяющие условию поиска, находящейся в окне таблицы атрибутов слоя. Для отбора используется поле *Population (Население)*, показанное на рис. 4.9. Для последующего сравнения производительности СУБД и ГИС нужно засечь время выполнения запроса (в секундах) и внести в табл. 4.1.

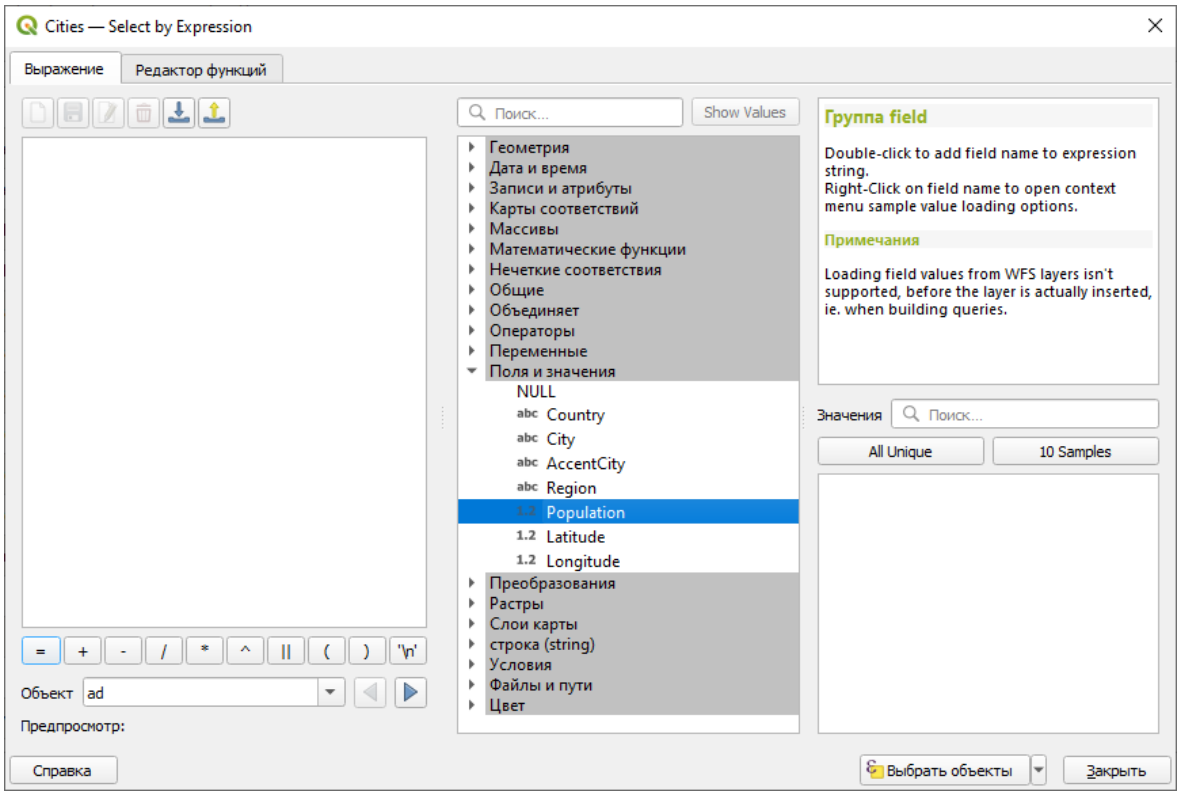


Рис. 4.9. Диалог отбора объектов по критерию

Таблица 4.1

Статистика времени выполнения запроса

Используемое программное обеспечение	Время выполнения запроса на поиск городов с ненулевым населением, с
QGIS	

Созданную выборку нужно сохранить в отдельный файл (контекстное меню, вызываемое щелчком правой кнопки по слою, выбрать *Экспорт – Сохранить выделенные объекты*) (рис. 4.10).

Сохраняем результаты запросы в новом shape-файле *Cities_not_null* (рис. 4.11).

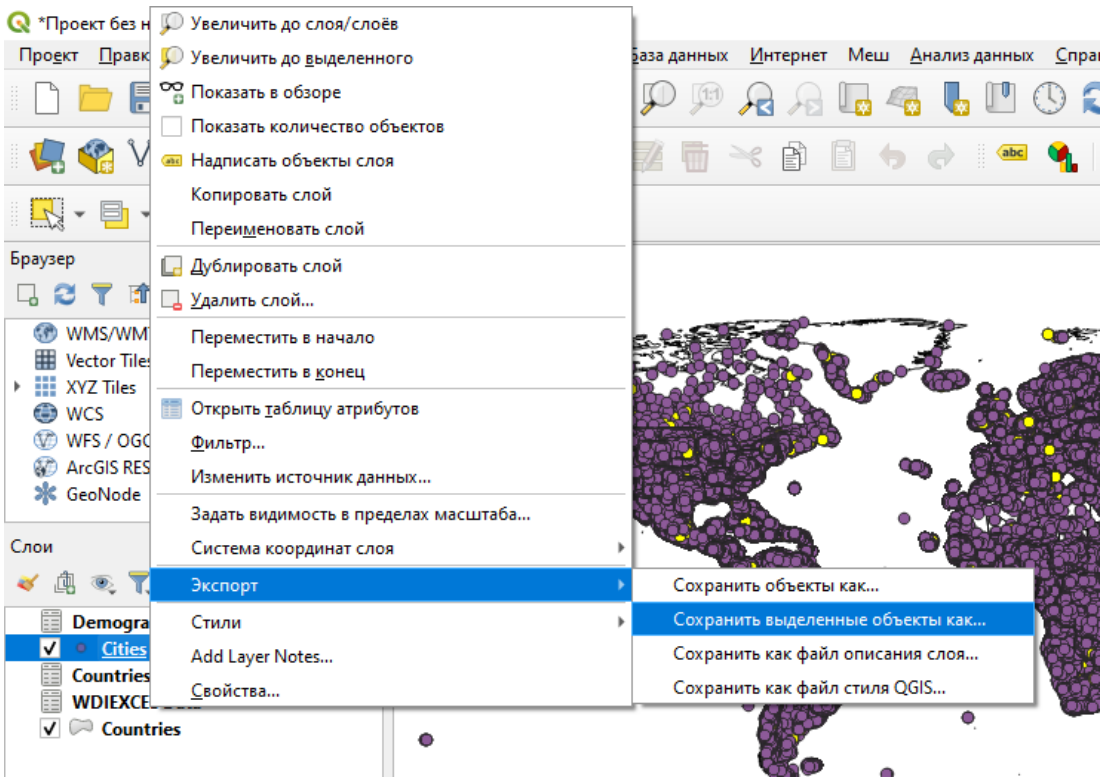


Рис. 4.10. Команда сохранения результатов выборки в отдельный файл

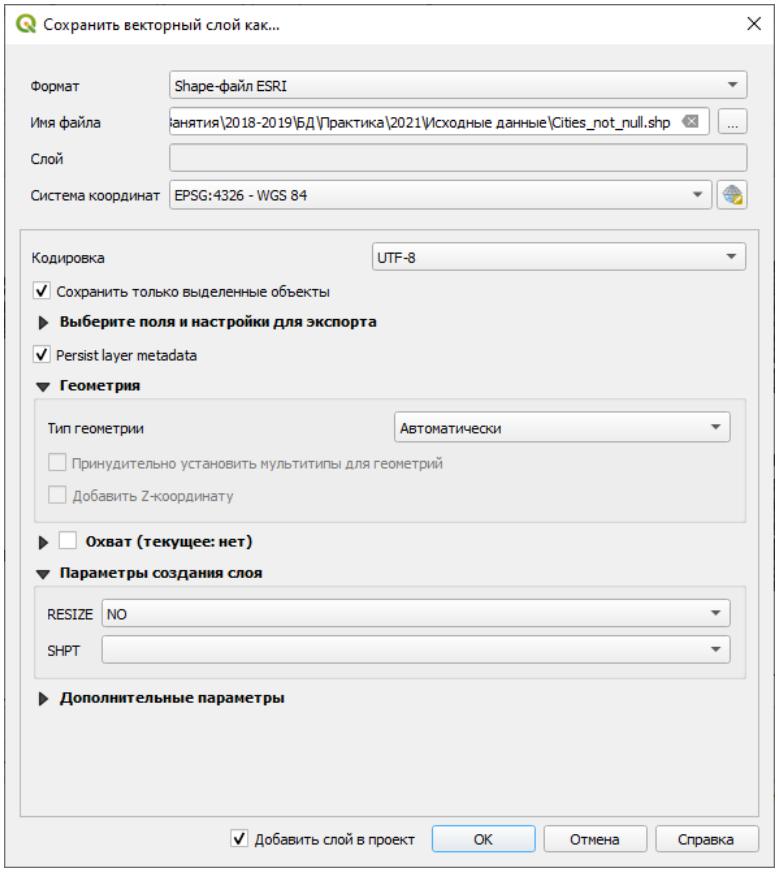


Рис. 4.11. Диалог указания параметров нового файла

Далее нужно закрыть исходную таблицу с городами и добавить созданную на предыдущем шаге (если это не было указано сразу в диалоге).

Запрос для выборки только тех городов, которые попадают на территорию Российской Федерации, формируется на основе пространственного отношения. Пространственный запрос находит объекты, полностью попадающие в указанные объекты одного из слоев (условие *are within*). Для осуществления выбора только внутри заданного полигона (из слоя *Countries*) нужно его предварительно выделить (вручную либо через запрос по названию объекта) (рис. 4.12). Также для дальнейшего сравнения нужно засечь время выполнения запроса на выбор в границах полигона и внести результаты в табл. 4.2. Результаты выполнения запроса представлены ниже на рис. 4.13.

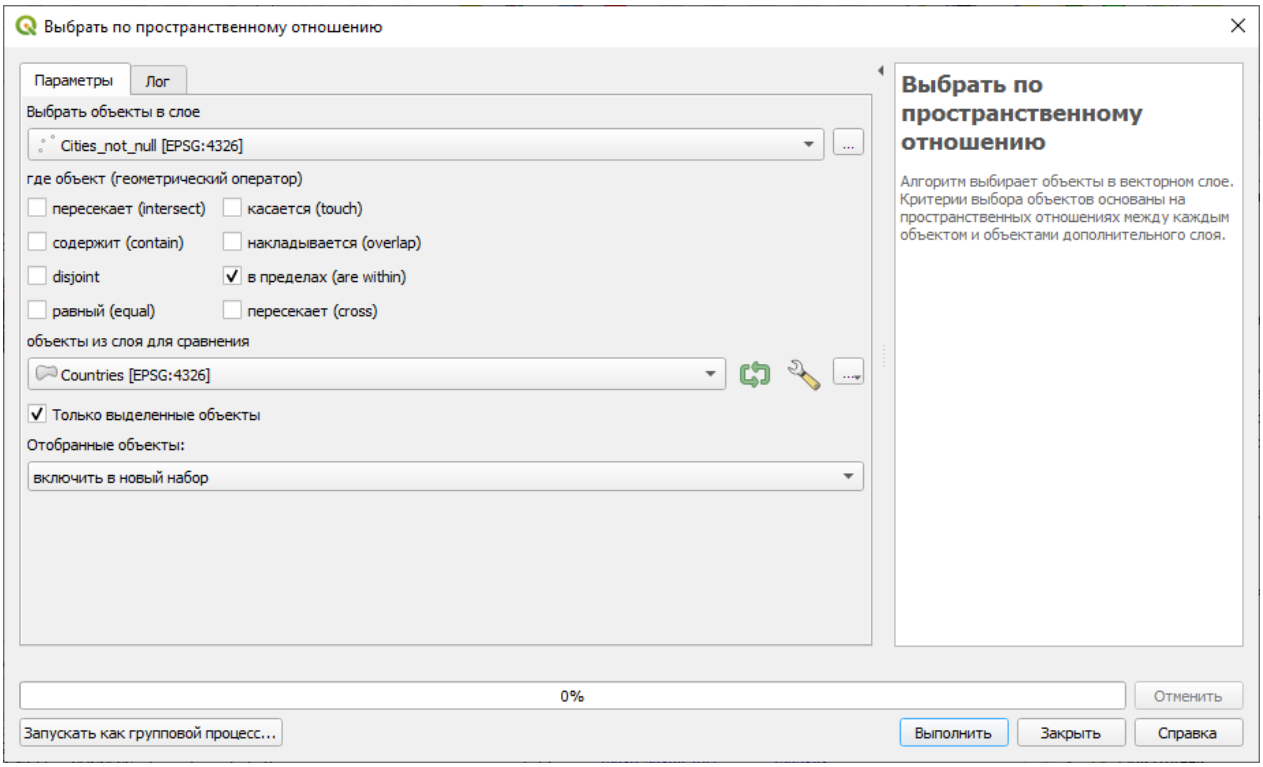


Рис. 4.12. Пространственный запрос
на поиск точек внутри заданного полигона

Таблица 4.2

Статистика времени выполнения запроса

Используемое программное обеспечение	Время выполнения запроса на поиск городов внутри указанной страны, с
QGIS	

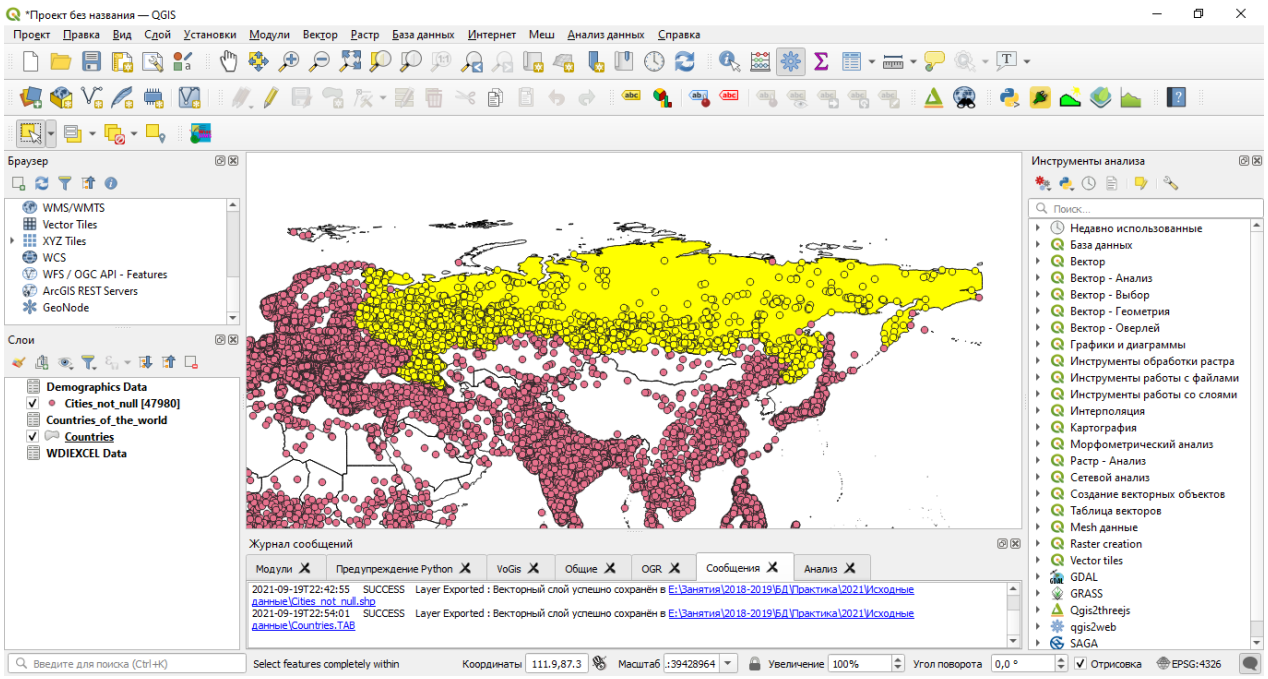


Рис. 4.13. Результат выборки

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Постановка задачи.
4. Описание выполненной работы со скриншотами и заполненными таблицами.

Вопросы и задания для защиты работы

1. Какая модель данных преимущественно используется в геоинформационных системах и почему?
2. В чем заключаются функции СУБД?
3. Опишите организацию данных в геоинформационной системе QGIS.
4. Опишите основные типы данных, которые позволяет импортировать геоинформационная система QGIS.
5. Перечислите порядок действий для создания точечных объектов на основе импортированных данных.
6. Перечислите порядок действий при интеграции внешних данных с таблицами QGIS.
7. Опишите принципы выполнения атрибутивных и пространственных запросов в QGIS.

5. ЛАБОРАТОРНАЯ РАБОТА № 2. АНАЛИЗ И ВИЗУАЛИЗАЦИЯ ДАННЫХ В БАЗЕ ПРОСТРАНСТВЕННЫХ ДАННЫХ ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ

Время выполнения – 4 часа.

Цель работы: усвоить понятие агрегирующих функций как одного из базовых понятий СУБД, способы визуализации данных в геоинформационных системах, принципы выполнения SQL-запросов, в том числе пространственных, к нескольким таблицам.

Задачи работы

1. Изучить:

– аналитические функции геоинформационных систем;

– способы визуализации статистических данных в геоинформационных системах.

2. Научиться:

– выполнять агрегирующие операции к данным в геоинформационных системах;

– строить тематические карты в геоинформационных системах;

– выполнять пространственные запросы к нескольким таблицам.

Перечень обеспечивающих средств

Для выполнения работы необходимо иметь методические указания по выполнению работы, а также программное обеспечение QGIS (версия 3.10 и выше).

5.1. Составление запросов с агрегирующей функцией

В QGIS можно выполнять запросы к данным (в том числе и из файлов), составляя текстовые запросы на языке SQL либо используя графический интерфейс. Первый вариант находится в меню *Базы данных – Менеджер*

БД, далее кнопка *Окно SQL*. Для запуска второго варианта нужно проделать те же действия, что и для первого варианта, и потом нажать кнопку *Конструктор SQL-запросов*.

Агрегирующие функции расположены в категории *Обобщение* и позволяют проводить базовые статистические вычисления над данными числовых типов. Для примера оценим средний уровень грамотности по регионам на основе данных из таблицы *Countries_of_the_world*.

Для выполнения этой задачи используется агрегирующая функция *Avg* и группировка (поле *Group by*) по колонке *Region*. Обратите внимание на то, что для корректного выполнения запроса требуется использовать одинарные кавычки в названиях таблиц и колонок вместо двойных (рис. 5.1).

Текст SQL-запроса:

SELECT 'Countries_of_the_world'.'Region',
AVG('Countries_of_the_world'.'Literacy (%)')
FROM 'Countries_of_the_world'
GROUP BY 'Countries_of_the_world'.'Region';.

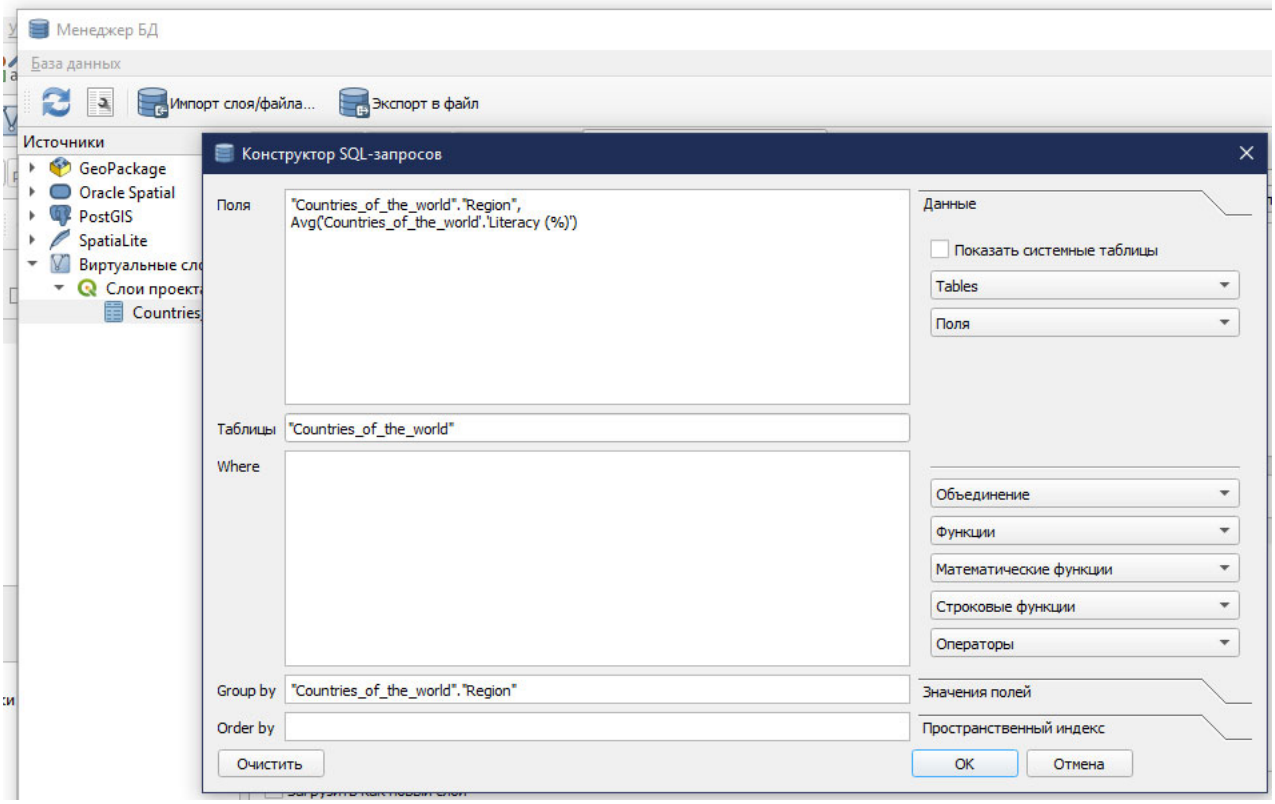


Рис. 5.1. SQL-запрос с агрегирующей функцией

В результате выполнения запроса получаем таблицу с названиями регионов и соответствующим им значениям показателей. Названия регионов и соответствующие им значения нужно внести в табл. 5.1.

Таблица 5.1

Результаты выполнения запроса расчета среднего уровня грамотности

Наименование региона	Средний уровень грамотности (%)

Далее составьте еще один запрос с использованием агрегирующей функции на примере подсчета количества стран, относящихся к определенному региону и отсортируйте полученный список по названию региона. Для вычисления количества используется функция *Count*, а для сортировки нужно указать необходимую колонку в поле *Order by*. В отчет нужно вставить текстовое выражение полученного запроса и заполнить табл. 5.2 в соответствие с полученными результатами.

Таблица 5.2

Результаты выполнения запроса вычисления количества стран по регионам

Наименование региона	Количество стран в регионе

5.2. Построение и выполнение запросов с расчетом количественных характеристик

Синтаксис SQL запросов допускает использование стандартных математических операций для расчета значений. В качестве примера рассчитаем плотность жителей на квадратный километр по странам путем деления количество жителей на вычисленную площадь полигона; данные используются из слоя *Countries*. Для расчета площади на основе метрики объектов используется стандартная функция *ST_Area* (список *Функции*).

Аргументом этой функции должна быть геометрия объектов (в QGIS это отдельное поле `*geometry*`).

Текст SQL-запроса:

SELECT 'Countries'.'NAME', 'Countries'.'POP2005' /
ST_Area('Countries'.geometry') FROM 'Countries'

В результате деления получается дробное число, что противоречит логике показателя, и для отображения результата в виде целого числа нужно дополнить запрос функцией *Round* с указанием нулевого значения в качестве параметра, показывающего количество знаков после запятой (рис. 5.2).

Текст SQL-запроса:

SELECT 'Countries'.'NAME', Round('Countries'.'POP2005' /
ST_Area('Countries'.geometry'), 0) FROM 'Countries'

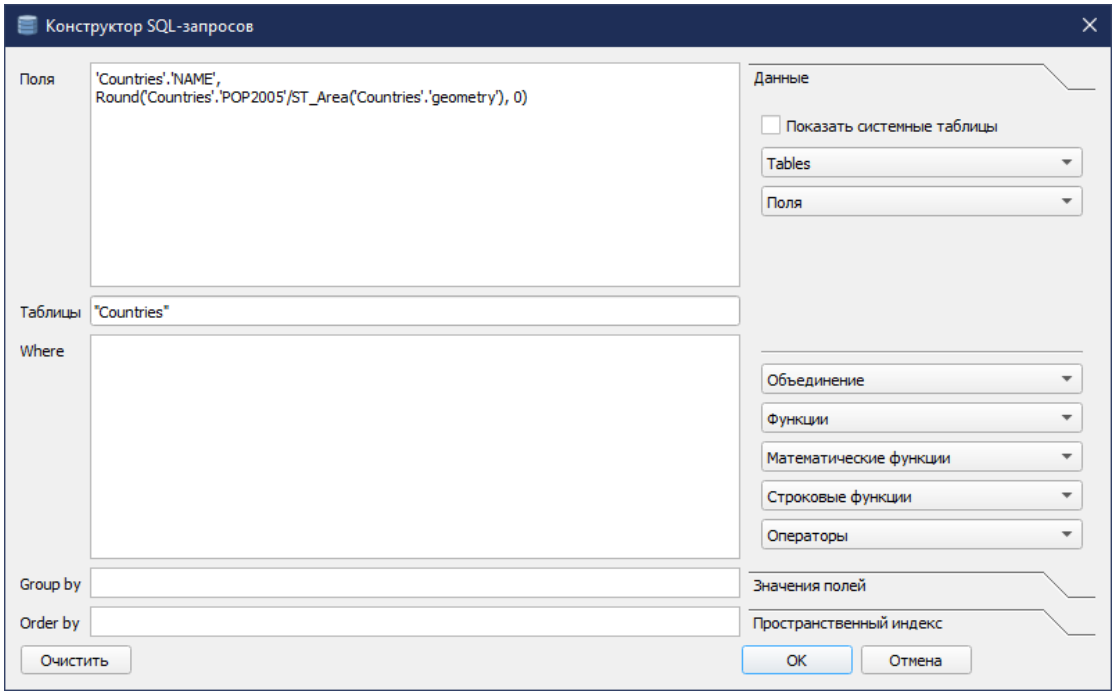


Рис. 5.2. Запрос с расчетом количественных пространственных характеристик и округлением результатов

Еще одним недостатком текущего варианта запроса является то, что в выводе, в заголовке столбца, указан полный текст запроса. Для задания

произвольного заголовка колонки в результатах запроса в QGIS используются псевдонимы (aliases) с тем же синтаксисом, что и в стандарте SQL, то есть после колонки или вычисляемого значения в блоке *SELECT* нужно добавить ключевое слово AS и в одинарных кавычках указать необходимое значение псевдонима. Для корректного представления результатов запроса с точки зрения человека нужно установить псевдоним для колонки с результатами расчета в виде значения *Density*.

Текст SQL-запроса:

SELECT 'Countries'.'NAME', Round('Countries'.'POP2005' /
ST_Area('Countries'.geometry'), 0) As 'Density' FROM 'Countries'

В результате выполнения запроса должен получиться следующий результат (рис. 5.3).

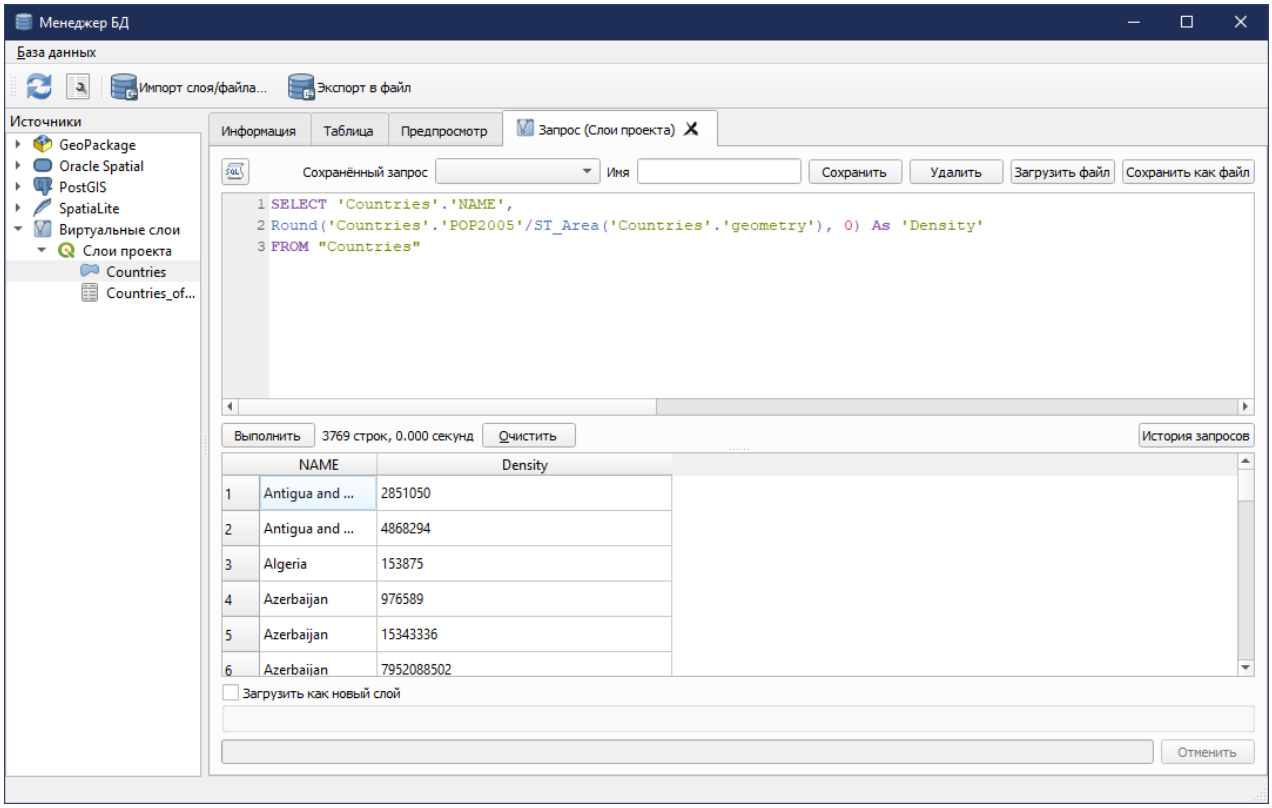


Рис. 5.3. Результаты запроса с расчетом количественных характеристик и группировкой результатов

5.3. Построение и выполнение пространственных запросов

Одно из основных отличий геоинформационных систем – наличие функций пространственного анализа и возможность выполнять пространственные запросы.

Например, выполним поиск всех городов, расположенных в 50 км от Москвы. В QGIS, при использовании стандартных инструментов, он выполняется в три шага.

Поскольку для геометрических вычислений QGIS использует единицы измерения слоя, то перед выполнением запросов слой с городами, полученный в первой работе, нужно сохранить с проекционной системой координат (например, *World Mercator*, *EPSG*: *3395*). Для этого используется пункт контекстного меню *Экспорт – Сохранить объекты как* (рис. 5.4). Для последующих запросов нужно будет использовать новый слой.

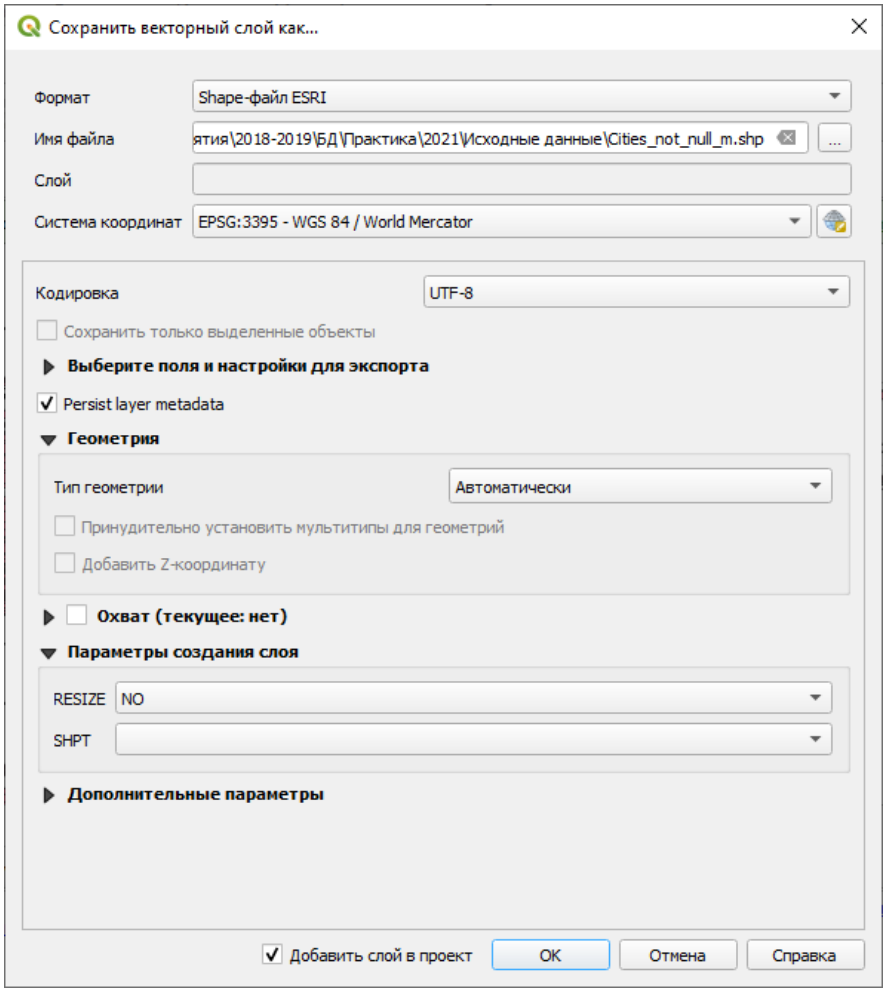


Рис. 5.4. Настройки изменения проекции слоя

Первым шагом будет поиск нужного города по названию. Для этого нужно выполнить команду *Выбрать объекты по выражению* или при открытой таблице атрибутов слоя нажать на кнопку *Выделить объекты, удовлетворяющие условию* (рис. 5.5).

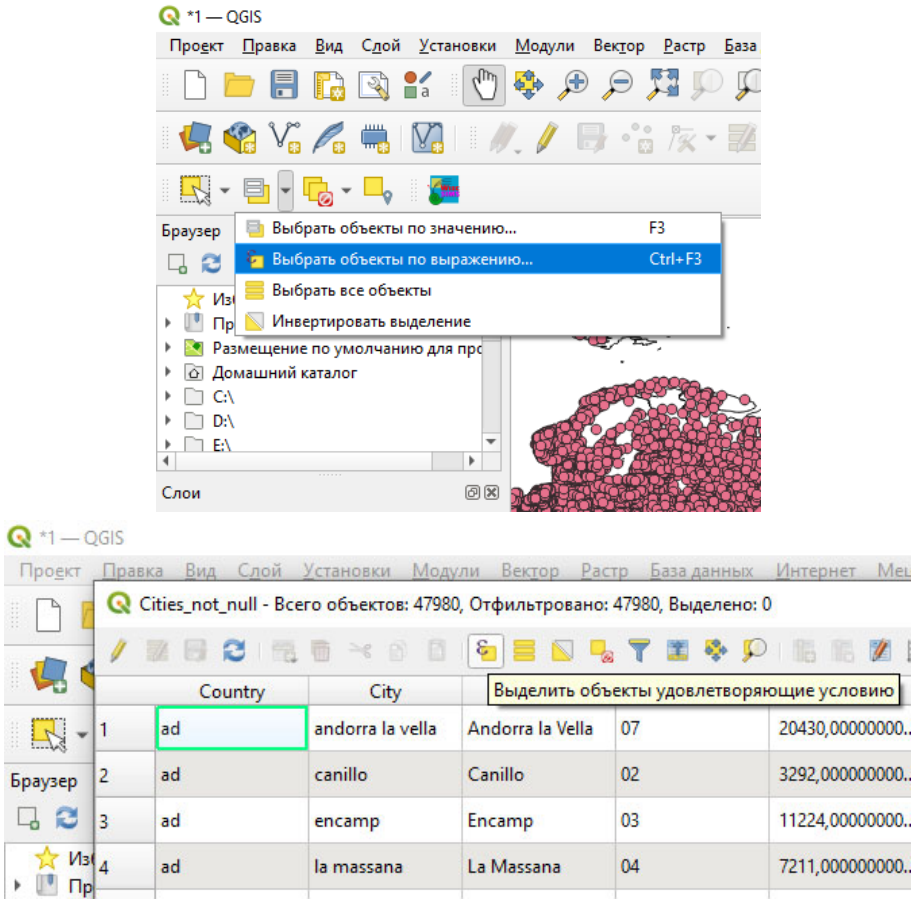


Рис. 5.5. Инструменты выбора объектов на основе значений атрибутов

Городов, имеющих такое название, два, поэтому, кроме указания названия, также нужно добавить условие принадлежности к определенному региону (рис. 5.6).

После того как визуально проверена корректность результатов поиска (для этого можно воспользоваться кнопкой *Увеличить до результатов поиска* в окне составления запроса), можно переходить к следующему шагу.

Далее, используя результаты предыдущего запроса, построим буферную зону с радиусом в 50 км относительно выбранного объекта. Для этого используется инструмент *Буфер*, находящийся в меню *Вектор – Геообработка* (рис. 5.7).

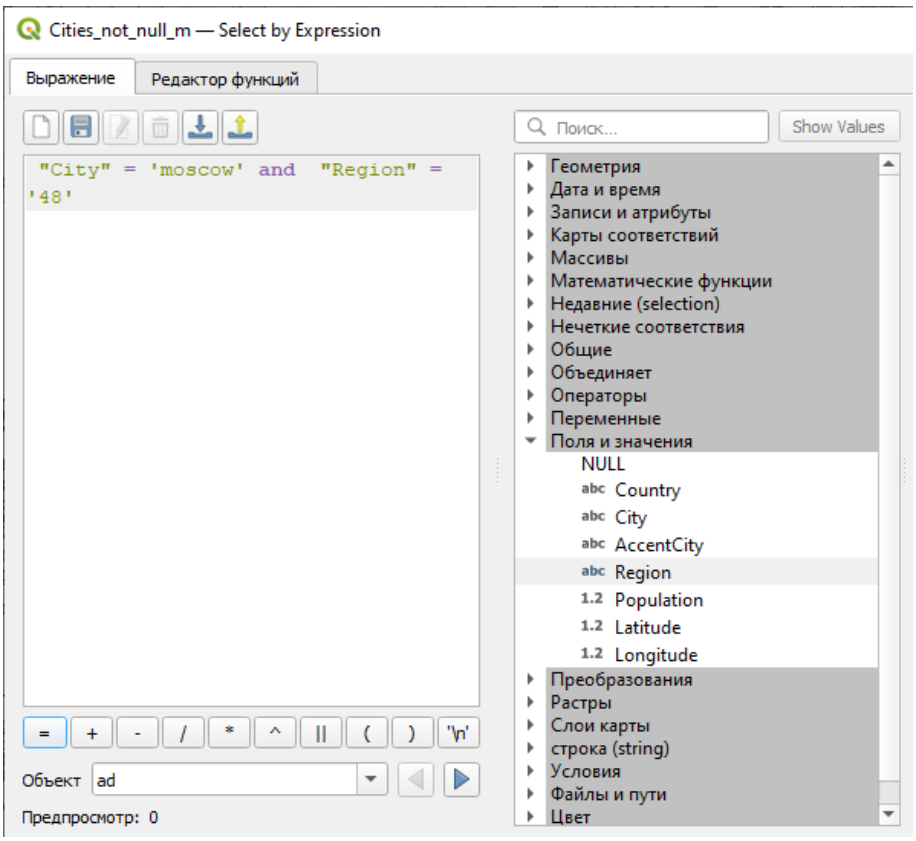


Рис. 5.6. SQL-запрос на выборку опорного объекта

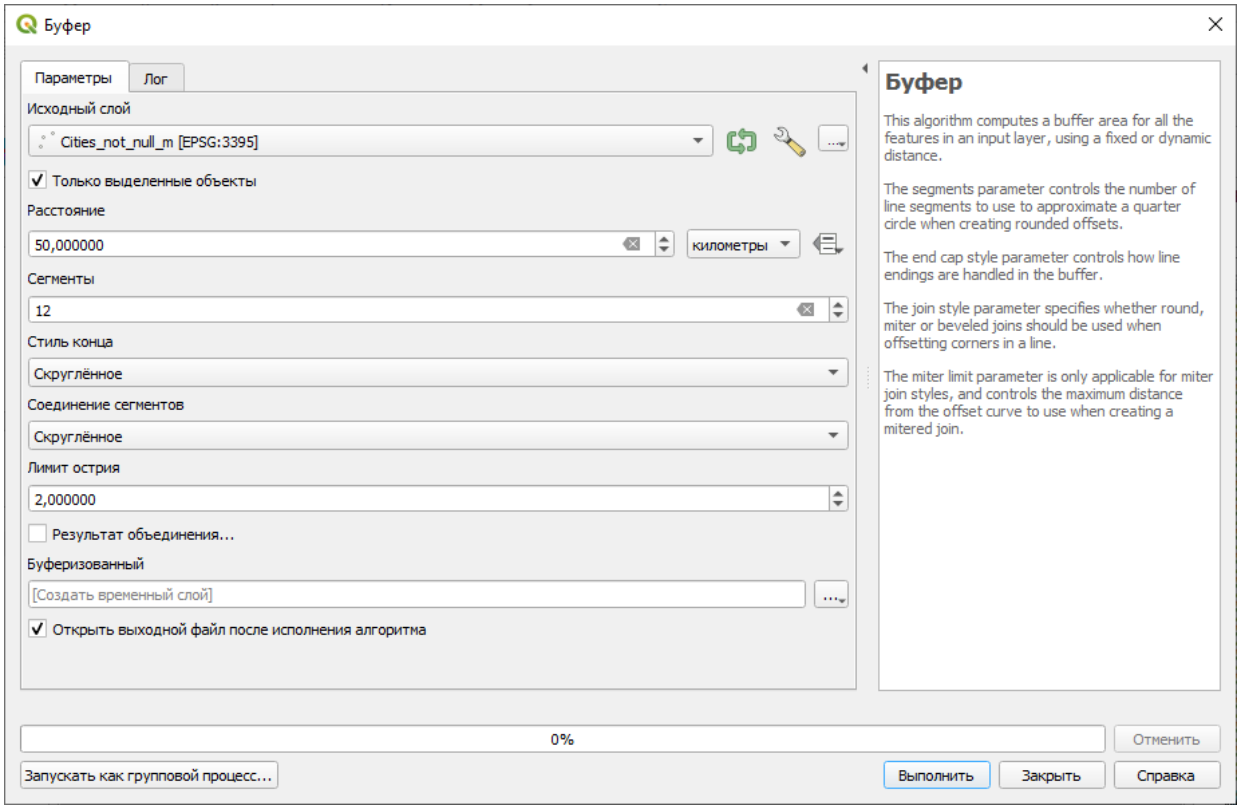


Рис. 5.7. Параметры инструмента построения буферной зоны

Теперь можно выполнить выбор городов, попавших в построенную буферную зону. Для этого нужно запустить инструмент *Выбрать по пространственному отношению*, находящийся в меню *Вектор – Выбор*, и использовать в настройках тип пространственного отношения – *в пределах* (*are within*) (рис. 5.8). В результате будут выделены города, находящиеся внутри 50-километровой буферной зоны (рис. 5.9).

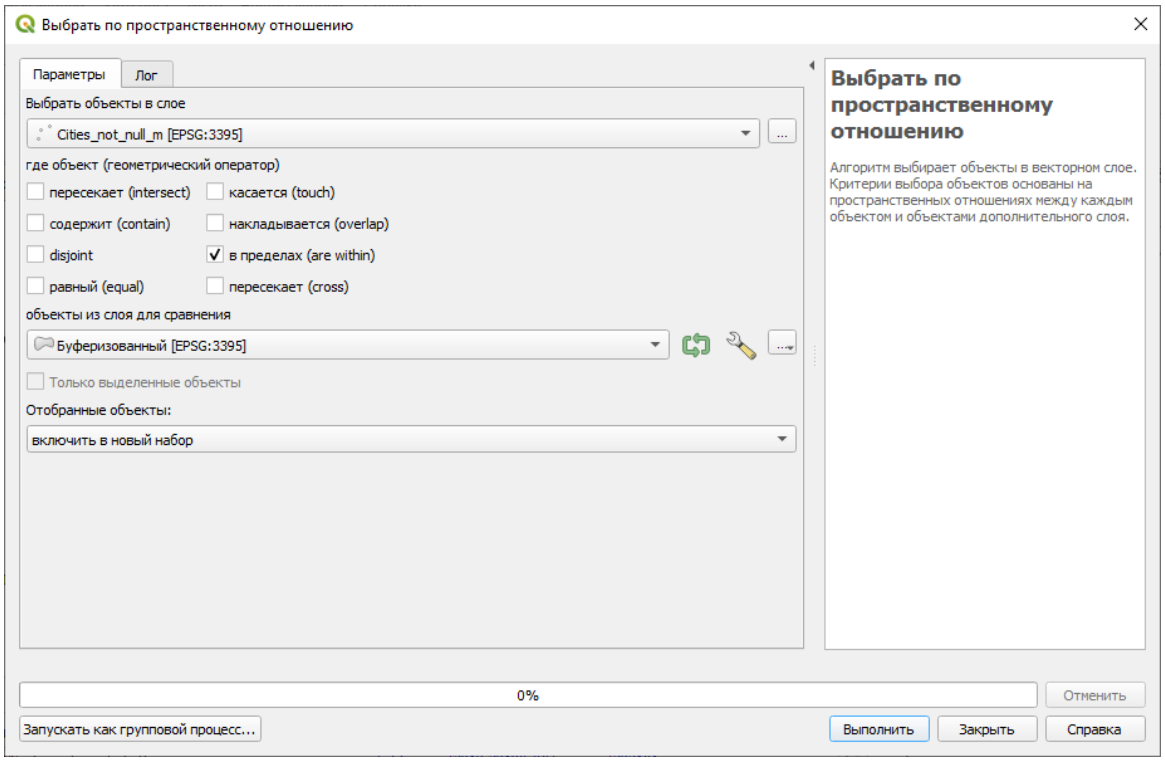


Рис. 5.8. Параметры выбора объектов

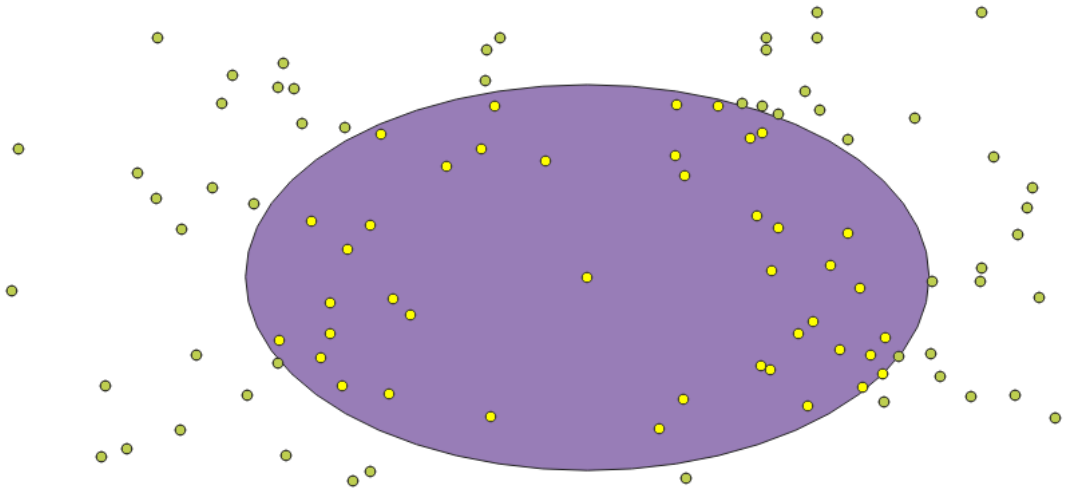


Рис. 5.9. Результаты запроса

Альтернативным вариантом решить эту задачу поиска объектов на указанном расстоянии можно с помощью команды *ST_Distance* (выпадающий список *Функции*), которая используется совместно с вложенным запросом (рис. 5.10).

Текст SQL-запроса:

SELECT 'Cities_not_null_m'.'City',
ST_Distance((SELECT 'Cities_not_null_m'.'geometry'
FROM 'Cities_not_null_m'
WHERE 'Cities_not_null_m'.'City' = 'Moscow' and 'Cities_not_null_m'.'Region' = '48'), 'Cities_not_null_m'.'geometry', 50000) AS 'Distance' FROM 'Cities_not_null_m'

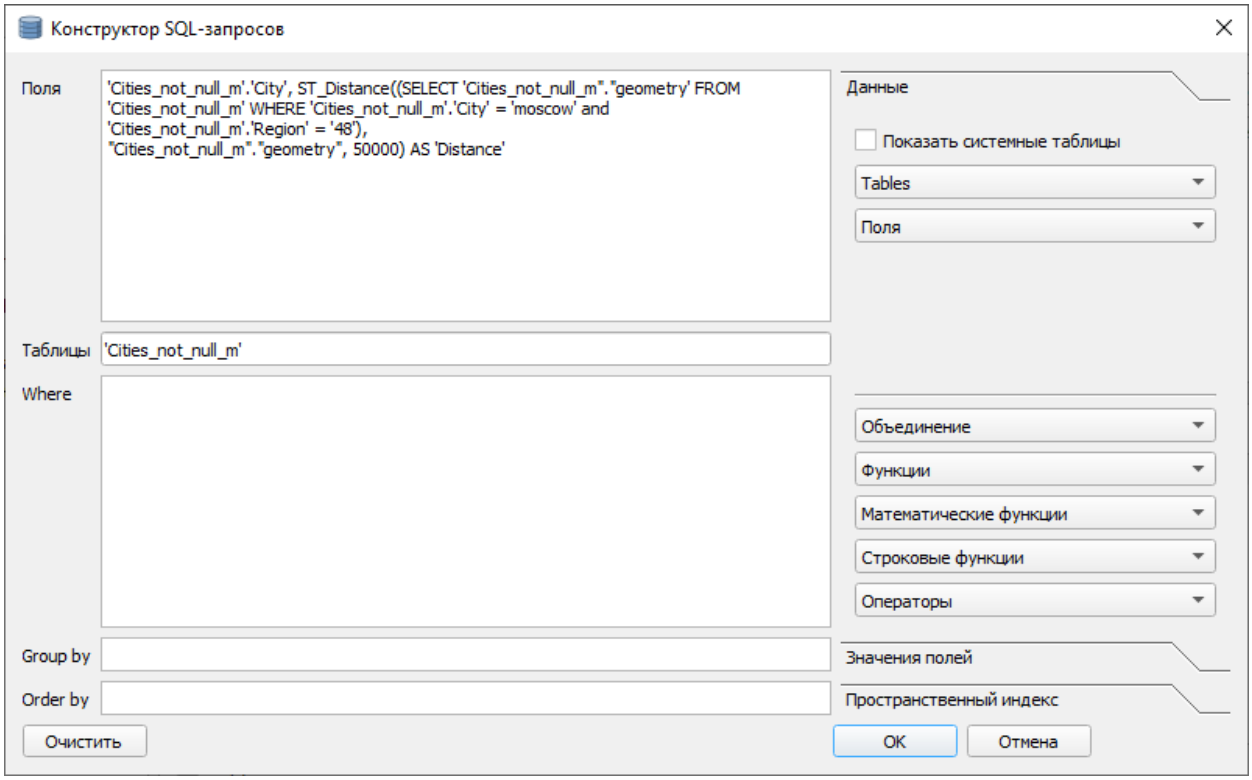


Рис. 5.10. Использование функции *ST_Distance* в SQL-запросе

5.4. Построение картограммы

Осуществим построение картограммы на основе данных о регионах, к которым принадлежат страны.

Поскольку в исходной таблице с контурами стран представлена не вся необходимая информация, то ее необходимо связать с информацией из таблицы *Countries_of_the_world*. Ключевыми полями для связи будут названия стран. Для того чтобы настроить параметры связи между таблицами, нужно открыть *Свойства основного слоя (Countries)* и перейти в раздел *Связи*. В этом разделе нужно нажать на кнопку *Добавить новую связь* и указать присоединяемую таблицу (*Countries_of_the_world*) и ключевые поля *Country* и *NAME* (рис. 5.11).

Теперь у стран мира добавилось поле с названиями регионов мира, и по этому полю можно построить тематическую карту регионов.

Для настройки тематической карты нужно перейти на вкладку *Оформление* в свойствах слоя. Выбрать *Символизацию по уникальным значениям* и указать, что значения будут использованы из присоединенного поля *Countries_of_the_world_Region* (рис. 5.12).

Результат создания тематической карты по регионам мира представлен на рис. 5.13 ниже.

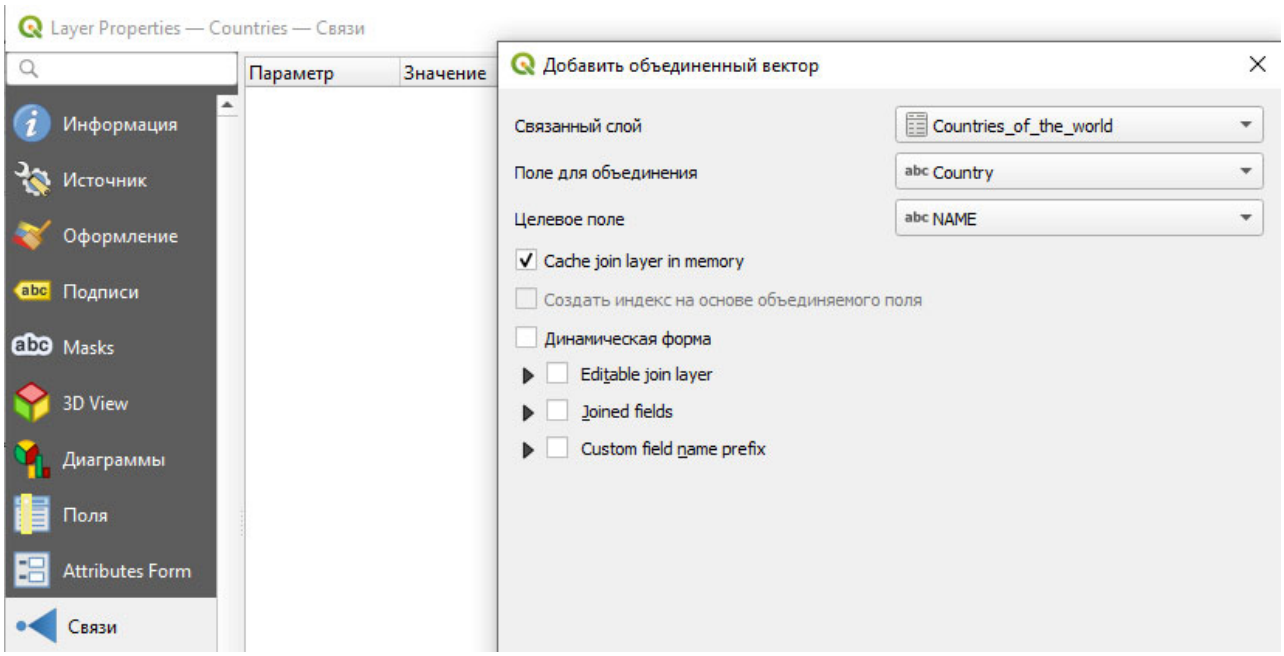


Рис. 5.11. Меню настройки связей таблиц

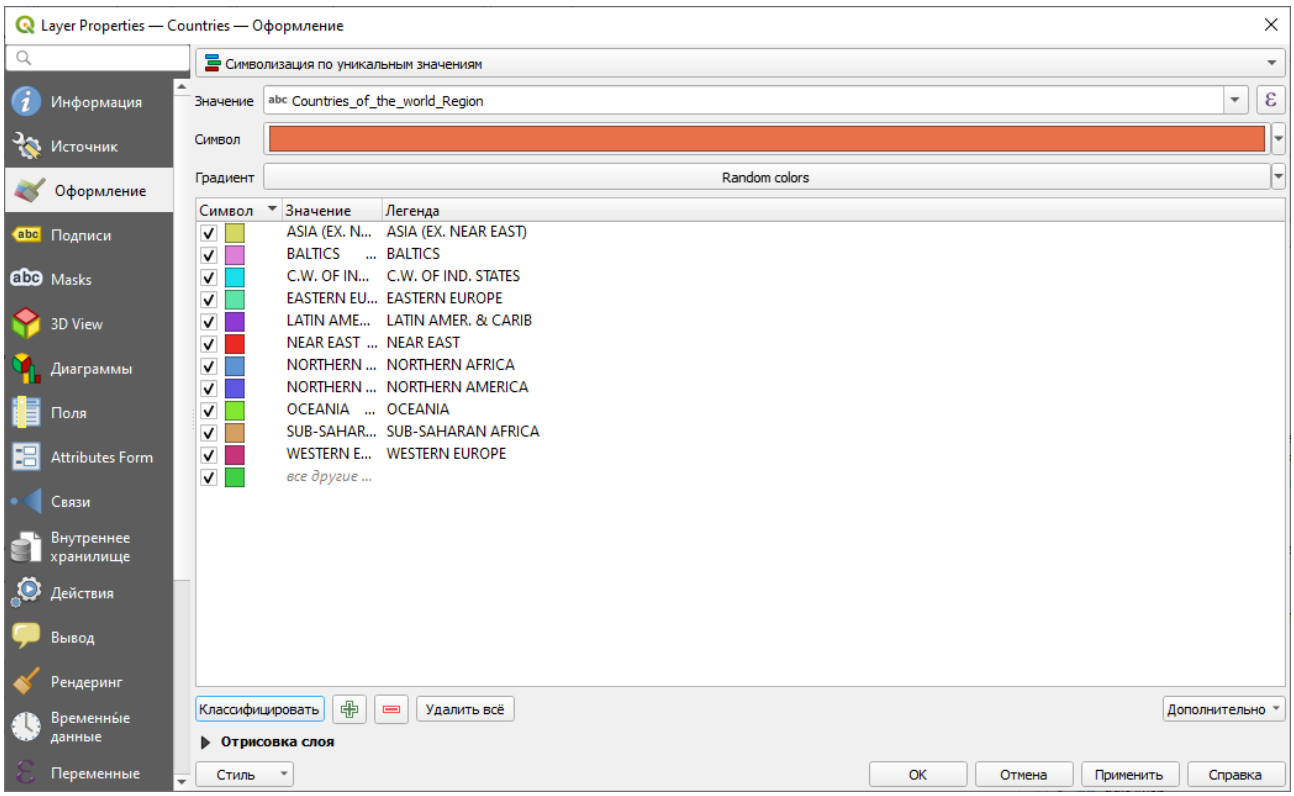


Рис. 5.12. Настройки внешнего вида тематической карты

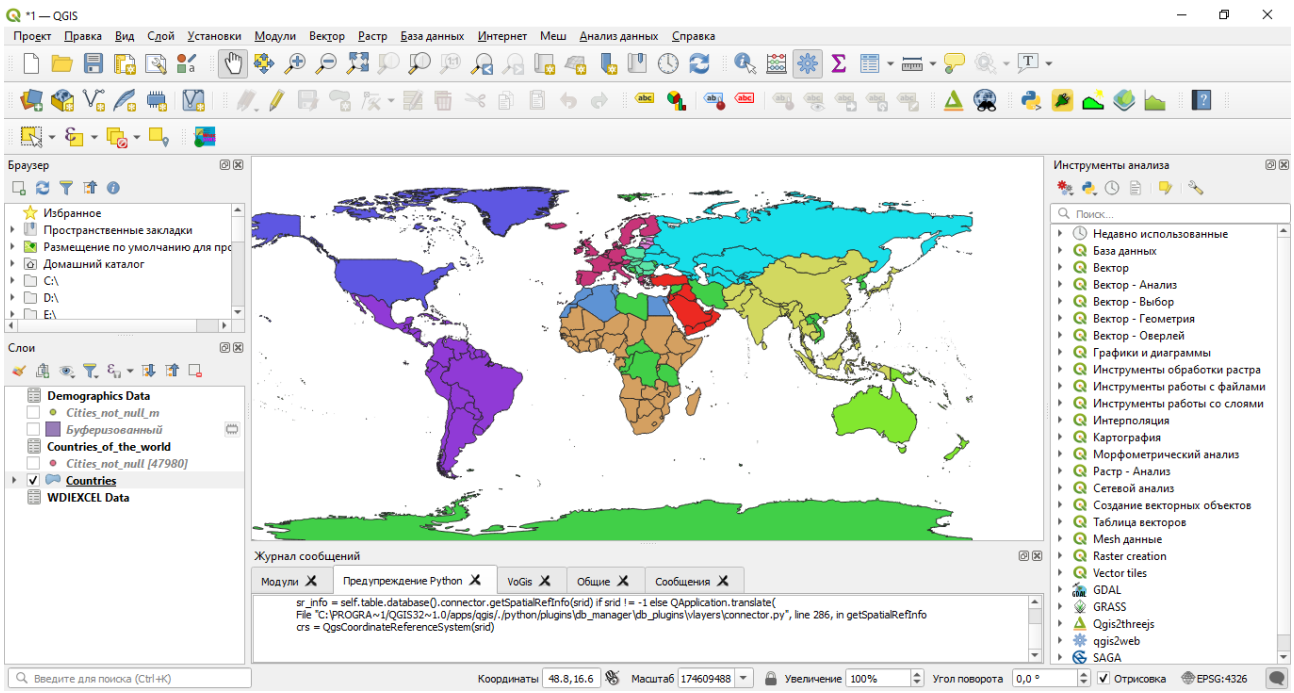


Рис. 5.13. Результат построения тематической карты

5.5. Построение картодиаграммы

Для примера построения картодиаграмм используются данные за 5 лет о численности населения по странам мира из таблицы *Demographics*.

Так же, как и в предыдущем примере, выполняется объединение таблиц с данными, только в этом случае не по названию страны, а по кодам стран (*ISO3* и *Country_Code*). Данный вариант позволит минимизировать возможные ошибки при связывании, в отличие от предыдущего варианта, где вероятность различного написания названий стран гораздо выше (рис. 5.14).

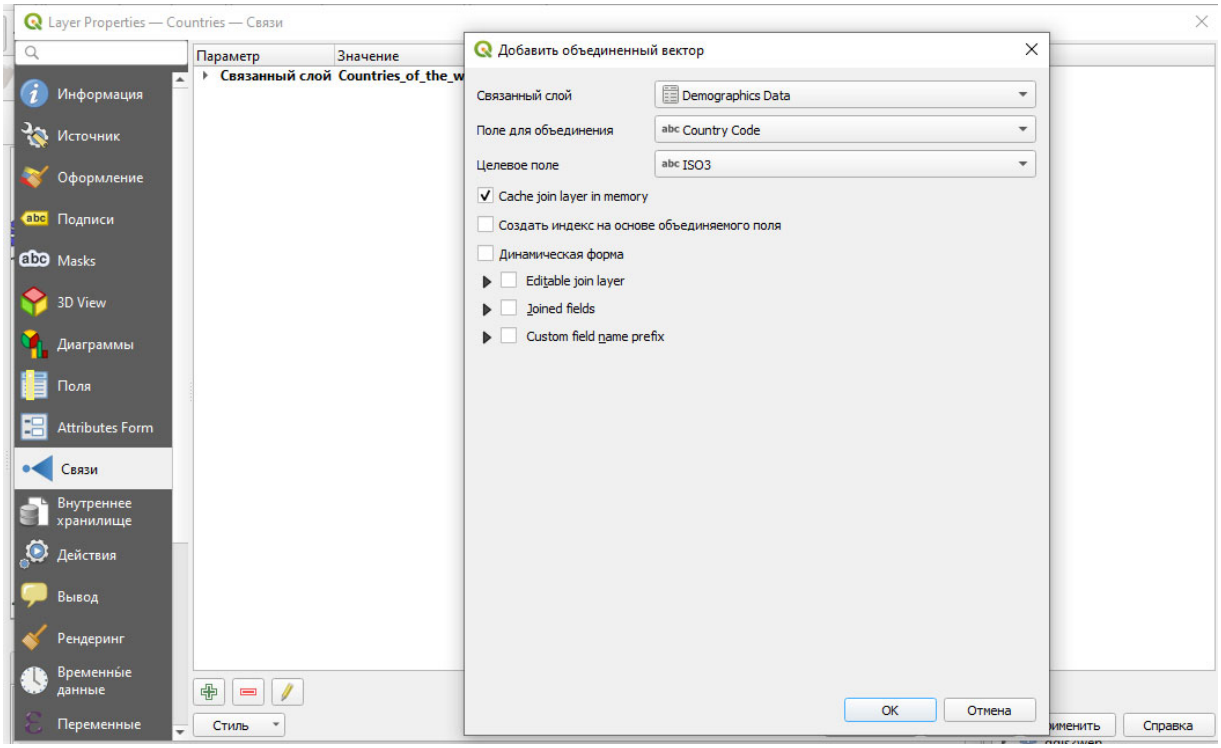


Рис. 5.14. Диалог связывания таблиц

Для построения столбчатой диаграммы (раздел *Диаграммы* в свойствах слоя) используются данные с 2009 по 2013 гг. В настройках выбирается тип диаграммы – *Гистограмма*, указываются 5 атрибутов с данными, на вкладке *Рендеринг* нужно обнулить отступы (*Bar Spacing*) между столбцами, на вкладке *Размер* подобрать оптимальное соотношение максимального значения и высоты столбца (рис. 5.15).

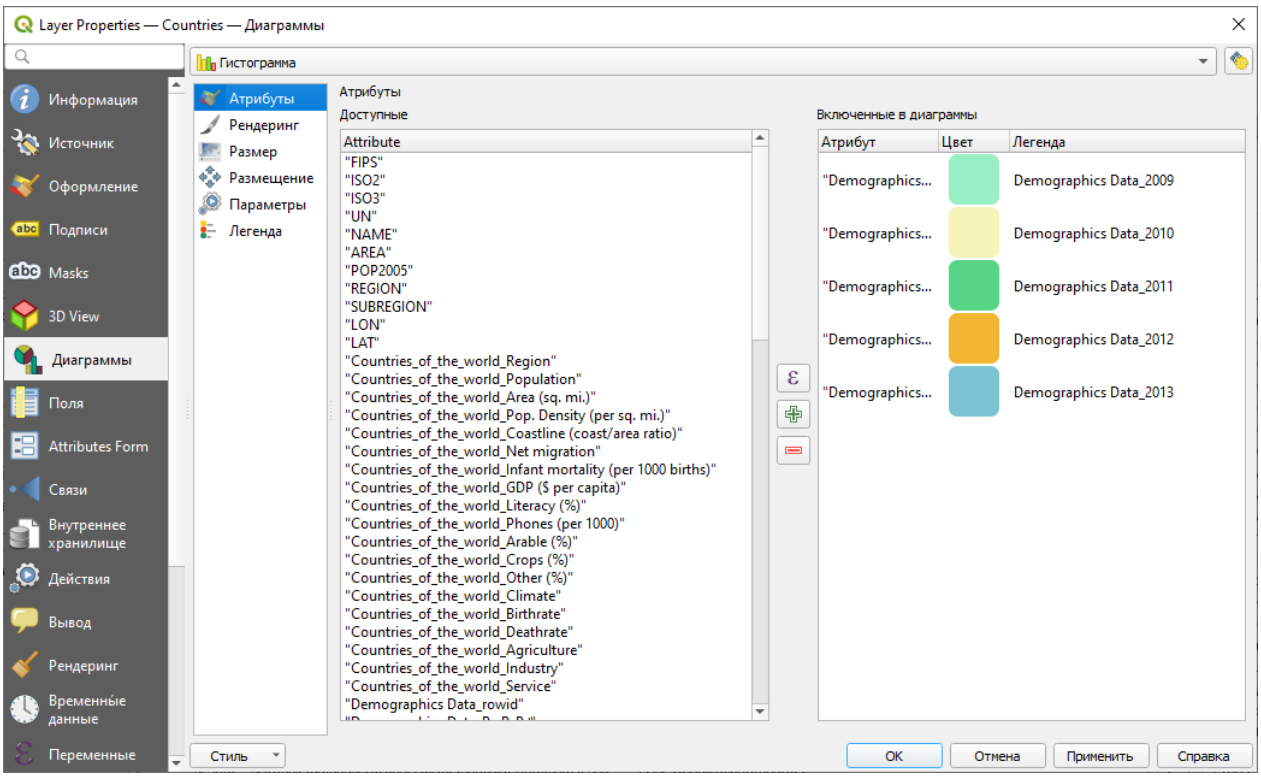


Рис. 5.15. Диалог настройки параметров диаграммы

Результат построения картодиаграмм представлен на рис. 5.16.

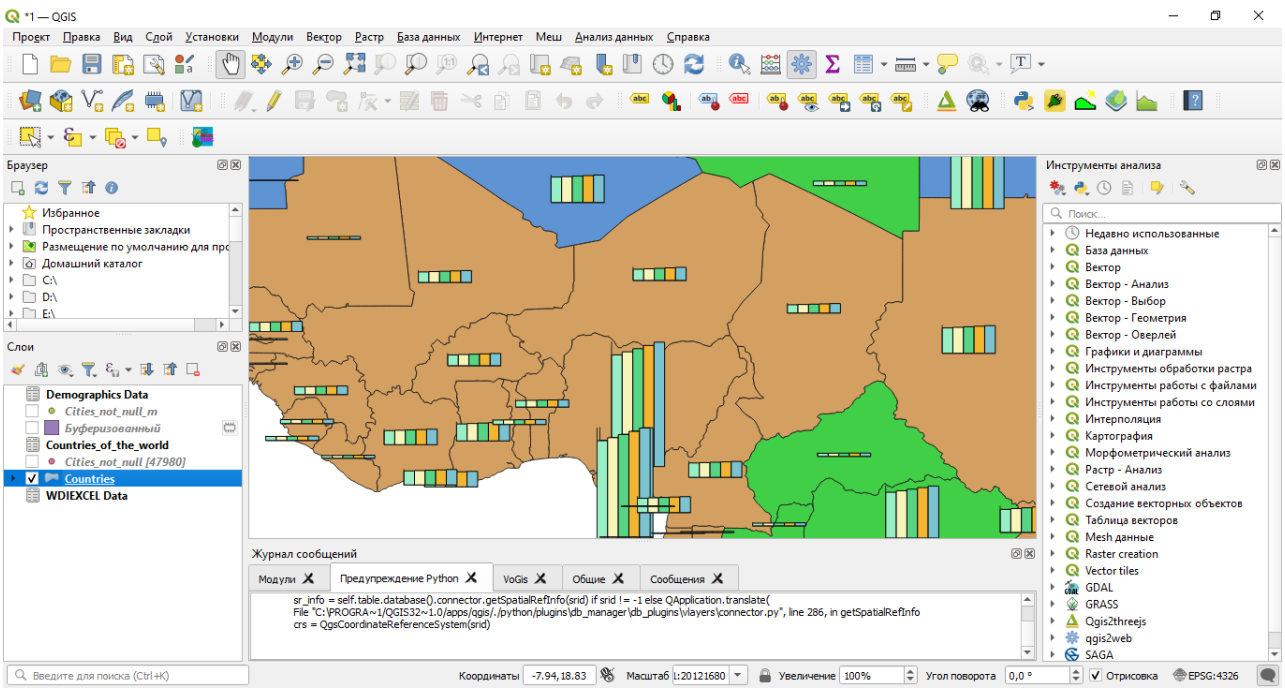


Рис. 5.16. Результат построения тематической карты

Для более наглядной визуализации можно включить подписи названий стран. Для этого в разделе *Подписи* свойств слоя нужно выбрать пункт *Простая подпись* и указать атрибут с данными – *Name*. Параметры шрифта и размещения текста подбираются самостоятельно для обеспечения наилучшей читаемости. Результат построения тематической карты с картодиаграммами и подписями представлен на рис. 5.17.

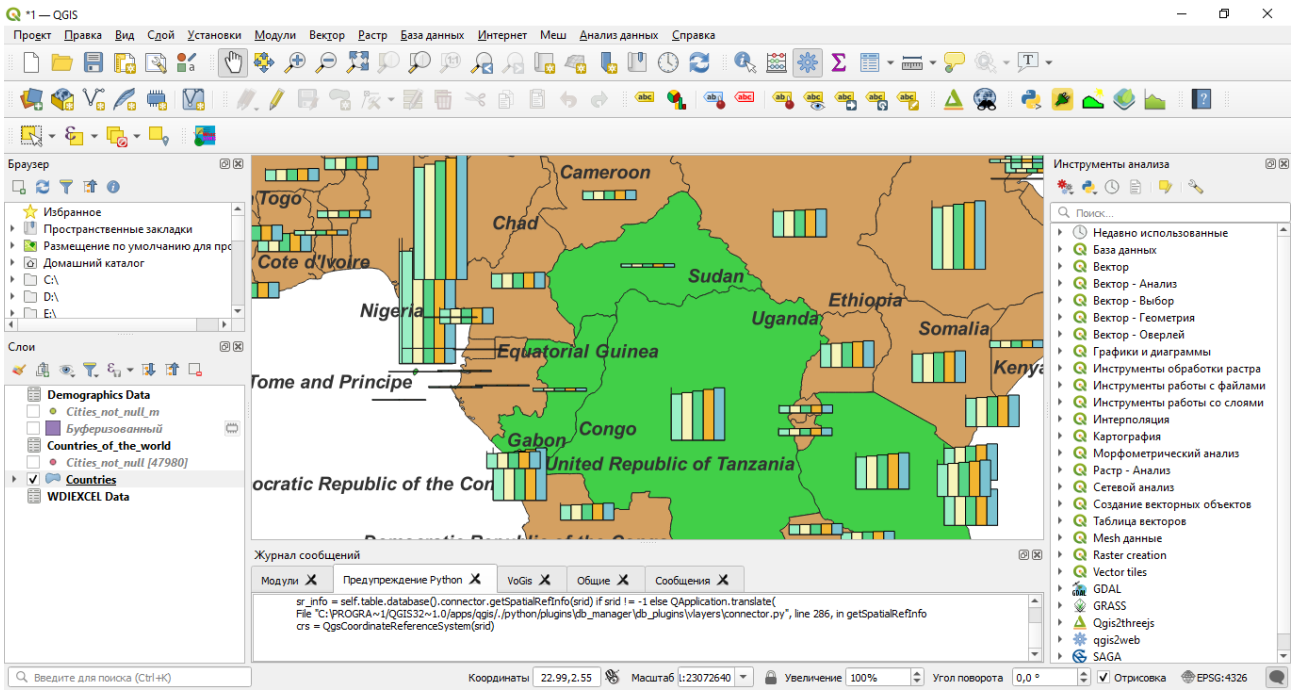


Рис. 5.17. Результат настройки внешнего вида слоев карты

5.6. Индивидуальные задания

Запрос № 1. Составить список стран региона с указанием количества населения за указанный год, упорядоченный по возрастанию. Регион и год взять на основе номера варианта (табл. 5.3). В отчет нужно вставить скриншот (один или более при необходимости) с составленным запросом и таблицу с результатами его работы.

Запрос № 2. Найти максимальную численность населения для определенной страны на указанном интервале лет (обе границы интервала включаются). В отчет нужно вставить скриншот (один или более при необходимости) с составленным запросом и итоговое значение. Варианты указаны в табл. 5.4.

Таблица 5.3

Варианты для запроса № 1

Номер варианта	Регион	Год
1	Asia (ex. Near East)	2002
2	Baltics	2010
3	C.W. Of Ind. States	2009
4	Eastern Europe	2008
5	Latin Amer. & Carib	2002
6	Near East	2005
7	Northern Africa	2004
8	Northern America	2003
9	Oceania	2008
10	Sub-Saharan Africa	2006
11	Western Europe	2007
12	Asia (ex. Near East)	2007
13	Baltics	2010
14	C.W. of Ind. States	2012
15	Eastern Europe	2011
16	Latin Amer. & Carib	2005
17	Near East	2003
18	Northern Africa	2004
19	Northern America	2006
20	Oceania	2007
21	Sub-Saharan Africa	2002
22	Western Europe	2010
23	Asia (Ex. Near East)	2010
24	Baltics	2005
25	C.W. of Ind. States	2007
26	Eastern Europe	2003
27	Latin Amer. & Carib	2006
28	Near East	2011
29	Northern Africa	2012
30	Northern America	2009

Таблица 5.4

Варианты для запроса № 2

Номер варианта	Страна	Интервал лет
1	Albania	2002–2004
2	Algeria	2008–2010
3	Argentina	2007–2009
4	Armenia	2006–2008
5	Austria	2002–2004
6	Bahrain	2005–2007
7	Belgium	2004–2006
8	Bolivia	2003–2005
9	Brazil	2008–2010
10	Bulgaria	2006–2008
11	Canada	2007–2009
12	Colombia	2007–2009
13	Croatia	2010–2012
14	Denmark	2010–2012
15	Estonia	2009–2011
16	Finland	2005–2007
17	Georgia	2003–2005
18	Guatemala	2004–2006
19	Hungary	2006–2008
20	Indonesia	2007–2009
21	Kenya	2002–2004
22	Liechtenstein	2010–2012
23	Mexico	2008–2010
24	Nepal	2005–2007
25	Norway	2007–2009
26	Pakistan	2003–2005
27	Philippines	2006–2008
28	Portugal	2009–2011
29	Qatar	2010–2012
30	Romania	2007–2009

Запрос № 3. Составить таблицу, выводящую процентное соотношение каждой из возрастных групп и процентное соотношение мужского и женского населения для указанной страны за указанный год. В отчет нужно вставить скриншот (один или более при необходимости) с составленным запросом и таблицу с результатами его работы. Варианты указаны в табл. 5.5.

Таблица 5.5

Варианты для запроса № 3

Номер варианта	Страна	Год
1	Portugal	2010
2	Albania	2012
3	Estonia	2012
4	Pakistan	2016
5	Liechtenstein	2015
6	Croatia	2014
7	Brazil	2020
8	Kenya	2011
9	Austria	2012
10	Finland	2013
11	Indonesia	2014
12	Armenia	2015
13	Bahrain	2020
14	Denmark	2011
15	Romania	2012
16	Belgium	2013
17	Bulgaria	2014
18	Argentina	2016
19	Guatemala	2018
20	Mexico	2012
21	Hungary	2011
22	Canada	2015
23	Colombia	2014
24	Nepal	2012
25	Bolivia	2010
26	Qatar	2011
27	Algeria	2016
28	Norway	2017
29	Georgia	2018
30	Philippines	2019

Запрос № 4. Рассчитать разницу между населением указанной страны и суммарным населением тех городов, которые относятся к этой стране. Для увеличения скорости работы рекомендуется выделить нужную страну и сохранить ее в отдельный слой (через пункт *Экспорт – Сохранить выделенные объекты как* контекстного меню слоя). В отчет нужно вставить скриншот (один или более при необходимости) с составленным запросом и итоговое значение. Варианты указаны в табл. 5.6.

Таблица 5.6

Варианты для запроса № 4

Номер варианта	Страна
1	Bolivia
2	Portugal
3	Georgia
4	Finland
5	Liechtenstein
6	Kenya
7	Qatar
8	Nepal
9	Argentina
10	Belgium
11	Bulgaria
12	Guatemala
13	Indonesia
14	Brazil
15	Algeria
16	Philippines
17	Canada
18	Norway
19	Bahrain
20	Austria
21	Mexico
22	Armenia
23	Colombia
24	Croatia
25	Denmark
26	Pakistan
27	Hungary
28	Estonia
29	Romania
30	Albania

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Постановка задачи.
4. Описание выполненной работы.
5. Описание выполнения индивидуальных заданий.

Вопросы и задания для защиты работы

1. Что такое агрегирующая функция в SQL-запросах?
2. Опишите способы настройки представления результатов запроса с помощью функций диалога «SQL-запрос».
3. Перечислите основные этапы выполнения пространственных запросов к нескольким таблицам.
4. Перечислите основные этапы построения тематических карт в геоинформационной системе QGIS.
5. Опишите основные параметры настройки тематической карты с картодиаграммами.

6. ЛАБОРАТОРНАЯ РАБОТА № 3. ИНТЕГРАЦИЯ ДАННЫХ РАЗЛИЧНЫХ ТИПОВ И СОСТАВЛЕНИЕ АНАЛИТИЧЕСКИХ ЗАПРОСОВ В БАЗЕ ПРОСТРАНСТВЕННЫХ ДАННЫХ РЕЛЯЦИОННОГО ТИПА

Время выполнения – 4 часа.

Цель работы: изучить основные этапы импорта табличных и пространственных данных в реляционную СУБД, усвоить понятие агрегирующих функций языка SQL, принципы составления, выполнения и получения данных для оптимизации SQL-запросов, в том числе пространственных, к нескольким таблицам, создания представлений и их использования в геоинформационных системах.

Задачи работы

1. Изучить:

– способы импорта данных в реляционные СУБД;

– принципы составления и оценки SQL-запросов;

– принципы создания и использования представлений в реляционных СУБД.

2. Научиться:

– выполнять агрегирующие операции к данным в реляционных СУБД с помощью SQL-запросов;

– оценивать производительность запросов и ускорять их с помощью индексов;

– выполнять пространственные запросы к нескольким таблицам.

Перечень обеспечивающих средств

Для выполнения работы необходимо иметь методические указания по выполнению работы, а также программное обеспечение PostgreSQL (версия 13 и выше), PostGIS (версия 3 или выше), pgAdmin (версия 4 или выше).

6.1. Создание базы данных

Запустить интерфейс к СУБД pgAdmin 4. Развернуть список серверов (пункт *Servers*). Найти сервер с именем *PostgreSQL 13* и подключиться к нему, щелкнув два раза кнопкой мыши или выбрав из контекстного меню пункт *Connect Server* (рис. 6.1). Пароль пользователя – тот, который был задан при установке (по умолчанию *postgres*).

Развернуть список *Databases* и открыть у него контекстное меню для создания базы данных (рис. 6.2).

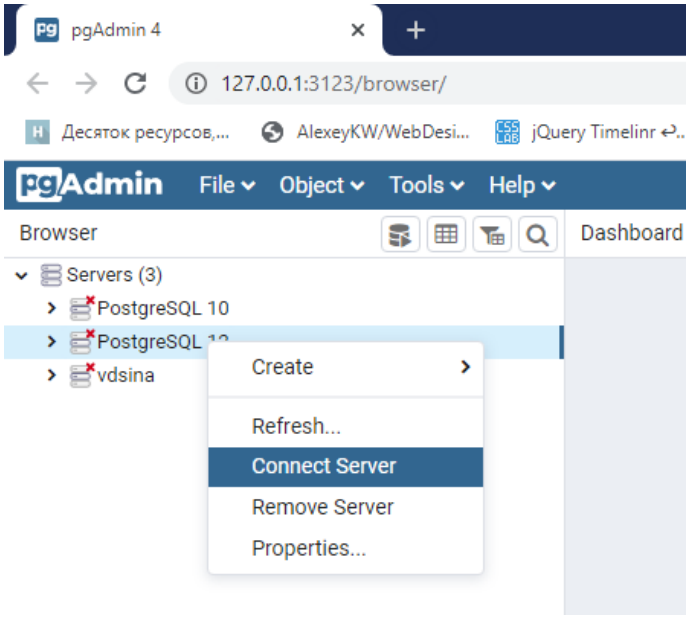


Рис. 6.1. Контекстное меню сервера баз данных

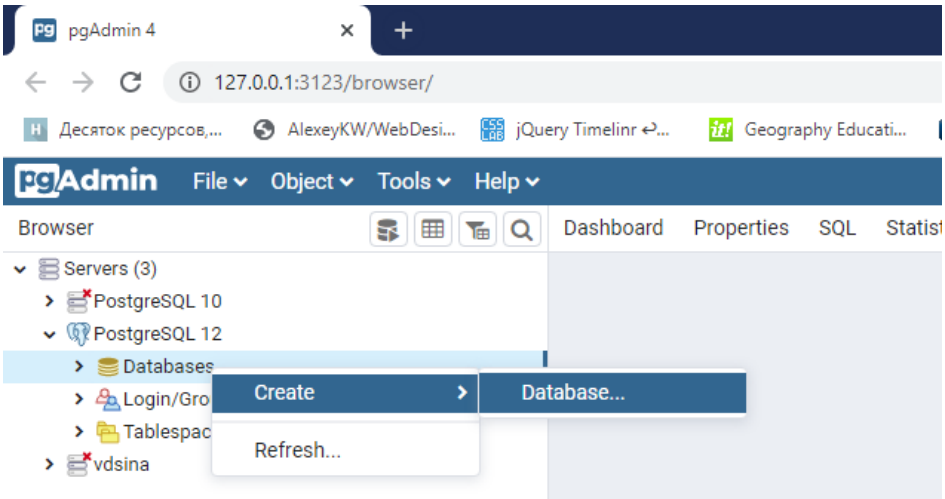


Рис. 6.2. Контекстное меню создания базы данных

Выбрать *Create* и далее *Database*. В открывшемся диалоге ввести в качестве названия базы данных свою фамилию транслитом и нажать на кнопку *Save* (рис. 6.3). В результате должна создаться новая база данных.

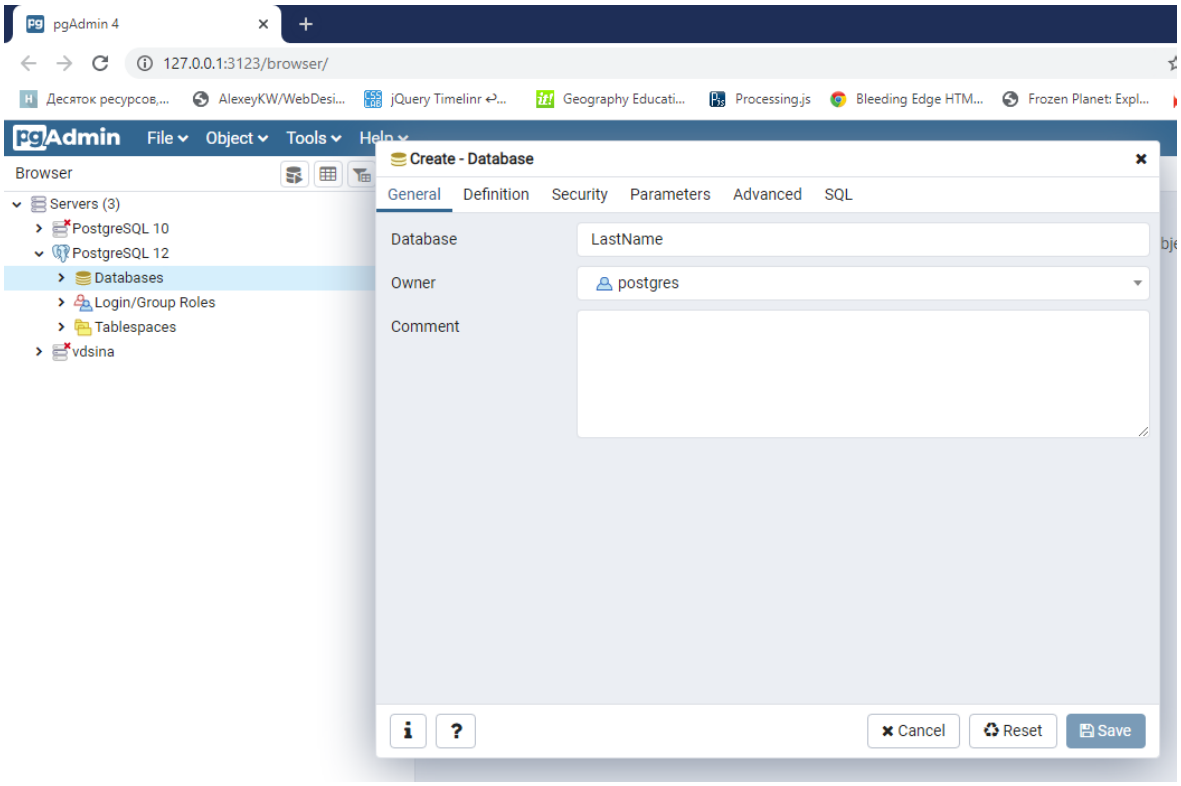


Рис. 6.3. Параметры создаваемой базы данных

6.2. Загрузка векторных данных в PostgreSQL

Для активации возможности работы с геоданными в созданной базе данных требуется предварительно активировать расширение *PostGIS*. Для этого нужно открыть вкладку для написания запросов *Query Tool* в меню *Tools* (рис. 6.4).

Затем вписать команды:

CREATE EXTENSION postgis;

CREATE EXTENSION postgis_topology; и нажать на кнопку *Execute/Refresh* .

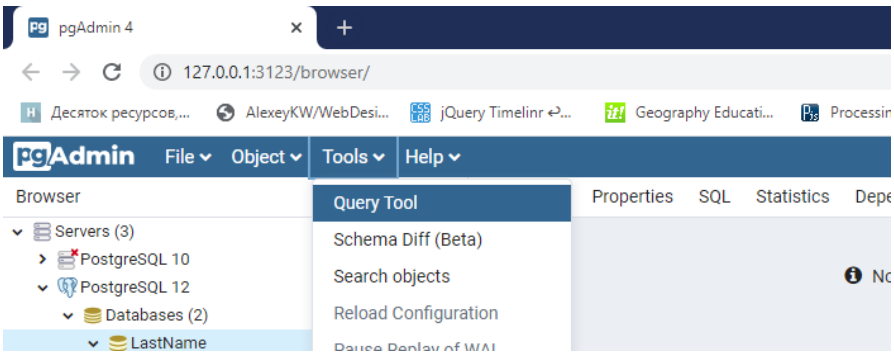


Рис. 6.4. Меню инструментов pgAdmin

В случае успешного выполнения будет выдано сообщение *Query returned successfully in…* В этом случае можно выполнить загрузку векторных данных по странам мира (*Countries.TAB*) из предыдущей работы в базу данных, используя утилиту *PostGIS Shapefile and DBF Loader*. Для получения SHP можно воспользоваться QGIS. Необходимо вызвать команду главного меню QGIS *Слой – Добавить слой – Добавить векторный слой*. Затем открыть контекстное меню слоя и выбрать пункт *Экспорт – Сохранить объекты как…* (рис. 6.5).

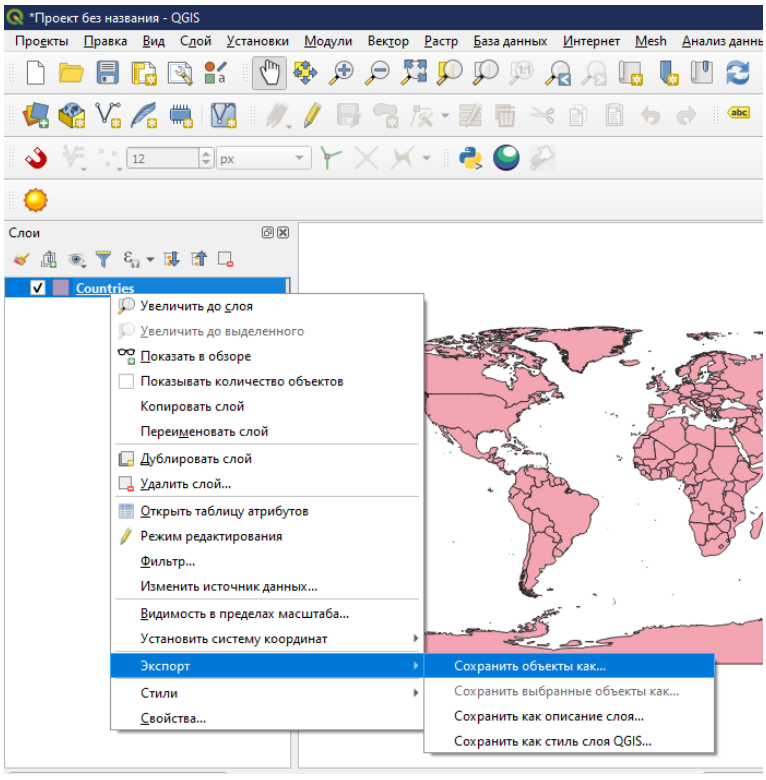


Рис. 6.5. Меню экспорта слоев

Указать формат сохранения *Shape-файл*, расположение файла (для последующего импорта в PostgreSQL в пути не должно быть кириллических символов) и кодировку *UTF-8* (рис. 6.6).

Далее нужно запустить программу *PostGIS Shapefile and DBF Loader*, которая устанавливается вместе с PostGIS. В открывшемся диалоговом окне нужно прописать параметры соединения в диалоге по кнопке *View connection detail*. Имя базы данных указывается то же самое, что было выбрано в начале работы (рис. 6.7).

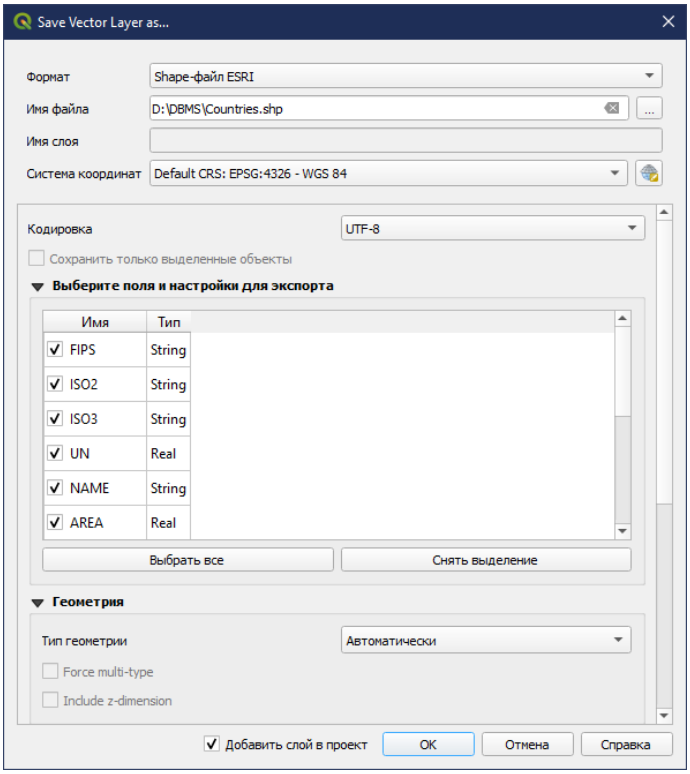


Рис. 6.6. Параметры экспорта векторного слоя

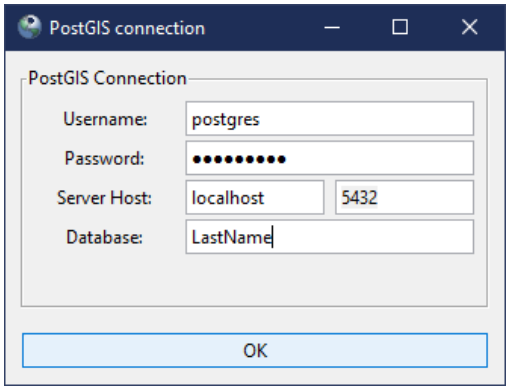


Рис. 6.7. Параметры подключения к СУБД PostgreSQL

Затем по нажатию кнопки *Add file* указать сконвертированный shp-файл и нажать кнопку *Import* (рис. 6.8).

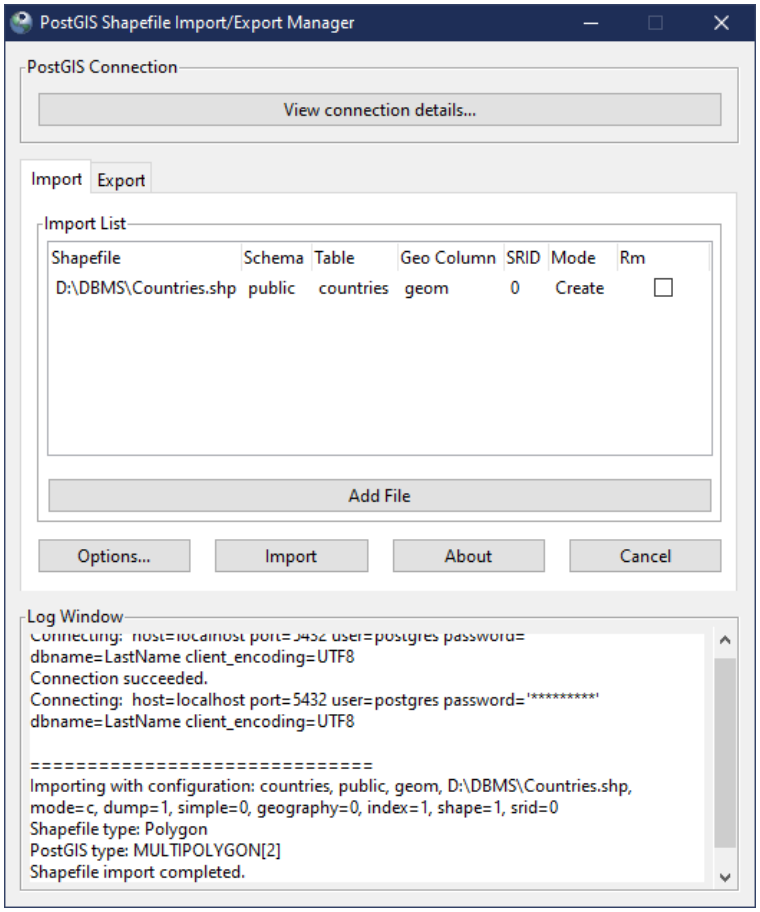


Рис. 6.8. Параметры загрузки SHP-файла в базу данных PostGIS

Для проверки корректности импорта созданную утилитой таблицу нужно открыть в QGIS. Для этого выбирается пункт главного меню *Слой – Добавить слой – Добавить слой PostGIS*. Затем создается новое соединение, для этого необходимо нажать кнопку *Создать*.

Пункты диалога заполняются в соответствии со скриншотом ниже (рис. 6.9), изменяется только имя базы данных.

После заполнения рекомендуется нажать на кнопку *Проверка соединения* и убедиться, что все параметры прописаны верно. Далее нужно нажать на кнопку *Подключиться* и дождаться появления списка схем данных. Затем развернуть схему данных и выбрать в списке таблицы с контурами стран (рис. 6.10).

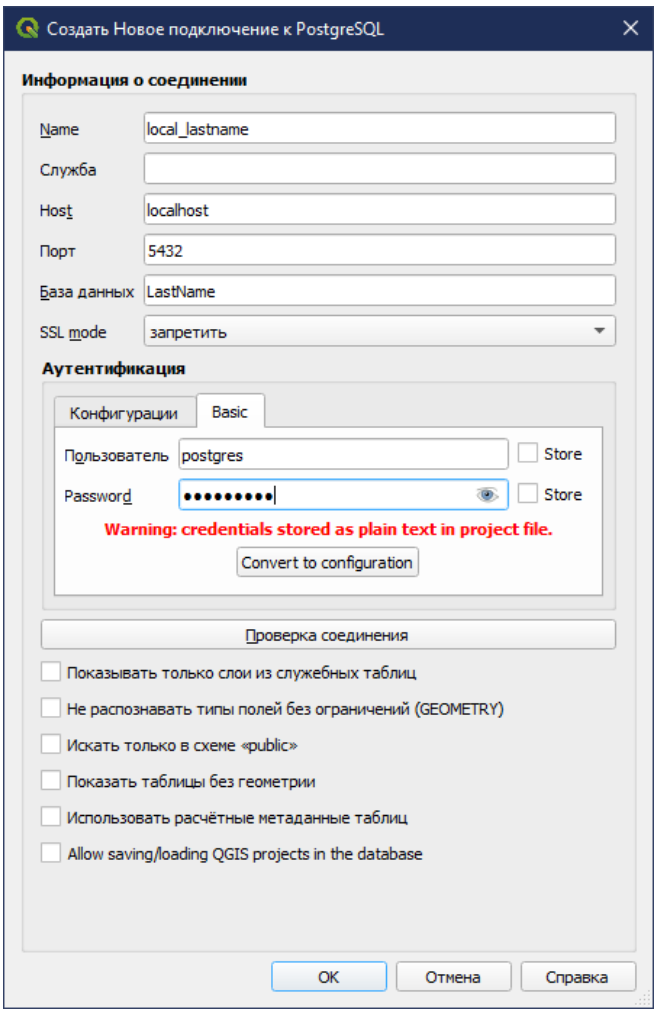


Рис. 6.9. Параметры соединения с сервером PostgreSQL

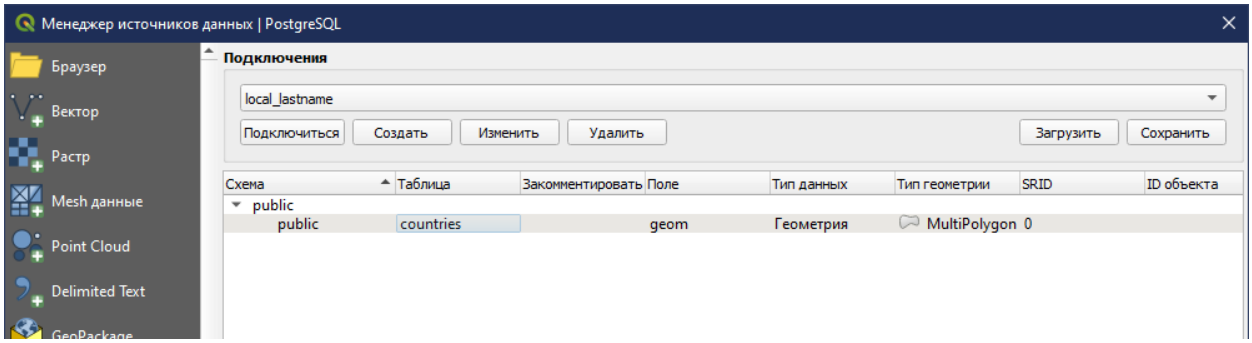
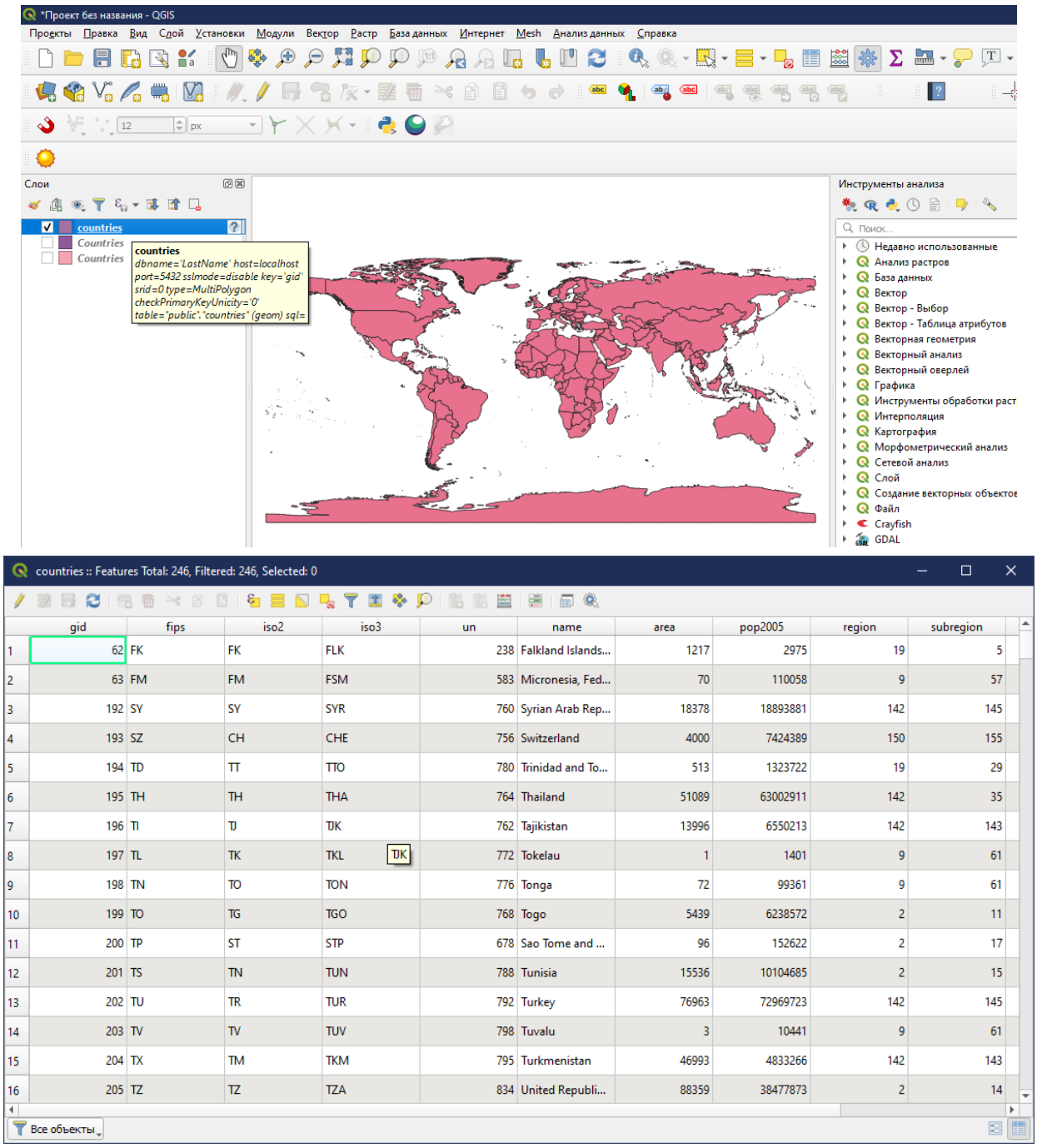


Рис. 6.10. Выбор слоя PostGIS для отображения

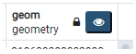
Убедиться, что геометрия и семантика отображаются корректно (рис. 6.11).



	gid	fips	iso2	iso3	un	name	area	pop2005	region	subregion
1	62	FK	FK	FLK	238	Falkland Islands...	1217	2975	19	5
2	63	FM	FM	FSM	583	Micronesia, Fed...	70	110058	9	57
3	192	SY	SY	SYR	760	Syrian Arab Rep...	18378	18893881	142	145
4	193	SZ	CH	CHE	756	Switzerland	4000	7424389	150	155
5	194	TD	TT	TTO	780	Trinidad and To...	513	1323722	19	29
6	195	TH	TH	THA	764	Thailand	51089	63002911	142	35
7	196	TI	TJ	TJK	762	Tajikistan	13996	6550213	142	143
8	197	TL	TK	TKL	772	Tokelau	1	1401	9	61
9	198	TN	TO	TON	776	Tonga	72	99361	9	61
10	199	TO	TG	TGO	768	Togo	5439	6238572	2	11
11	200	TP	ST	STP	678	Sao Tome and ...	96	152622	2	17
12	201	TS	TN	TUN	788	Tunisia	15536	10104685	2	15
13	202	TU	TR	TUR	792	Turkey	76963	72969723	142	145
14	203	TV	TV	TUV	798	Tuvalu	3	10441	9	61
15	204	TX	TM	TKM	795	Turkmenistan	46993	4833266	142	143
16	205	TZ	TZ	TZA	834	United Republi...	88359	38477873	2	14

Рис. 6.11. Отображение слоя PostGIS в QGIS

Альтернативным вариантом просмотра является добавленный в 12-й версии PostgreSQL *Geometry Viewer*. Для этого сначала нужно отобразить данные из таблицы *Countries*: развернуть поочередно списки *Имя_базы_данных – Schemas – Public – Tables*, щелкнуть правой кнопкой по таблице *Countries* и выбрать пункт *View/Edit Data – All Rows*. После этого в основном окне

будет выведен запрос и его результаты. Для отображения контуров стран нужно пролистать вправо до колонки с координатами объектов (*geom*) и в ее заголовке нажать на значок просмотра 

(рис. 6.12).

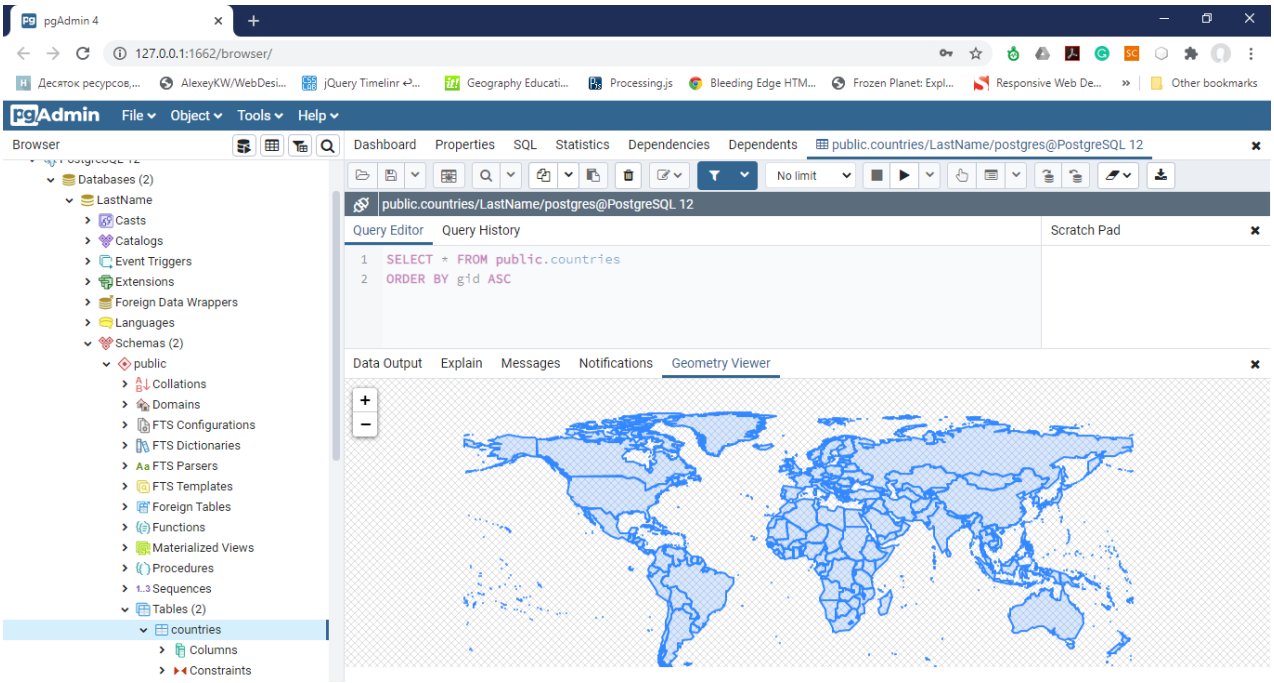


Рис. 6.12. Отображение пространственных данных в pgAdmin

6.3. Импорт внешних таблиц в базу данных

Следующим шагом нужно добавить информацию из Excel-файла *WDI.xlsx*. Для этого основной лист с данными нужно сохранить в формате CSV (при сохранении также требуется, чтобы в пути не было кириллических символов). Перед выполнением операций импорта в сохраненном ранее CSV-файле нужно заменить запятую на точку, которая будет являться разделителем целой и дробной частей в значениях показателей (рекомендуется вместо стандартного блокнота использовать Notepad++ или аналогичные программы).

После сохранения нужно создать структуру таблицы для импорта данных. Для этого в контекстном меню пункта *Tables* выбирается пункт *Create* и затем *Table* (рис. 6.13).

Название таблицы рекомендуется использовать такое же, как и у исходного Excel-файла. Названия колонок (не рекомендуется использовать

пробелы и спецсимволы) и типы данных на вкладке *Columns* указываются в соответствии со структурой исходного Excel-файла. Также для колонок с указанием года (с 1960 по 2017) нужно в начале добавить букву *y* (year), чтобы впоследствии упростить структуру запроса (иначе эти колонки в запросах нужно было бы обрамлять кавычками). После завершения создания структуры таблицы нужно нажать на кнопку *Save (*рис. 6.14*)*.

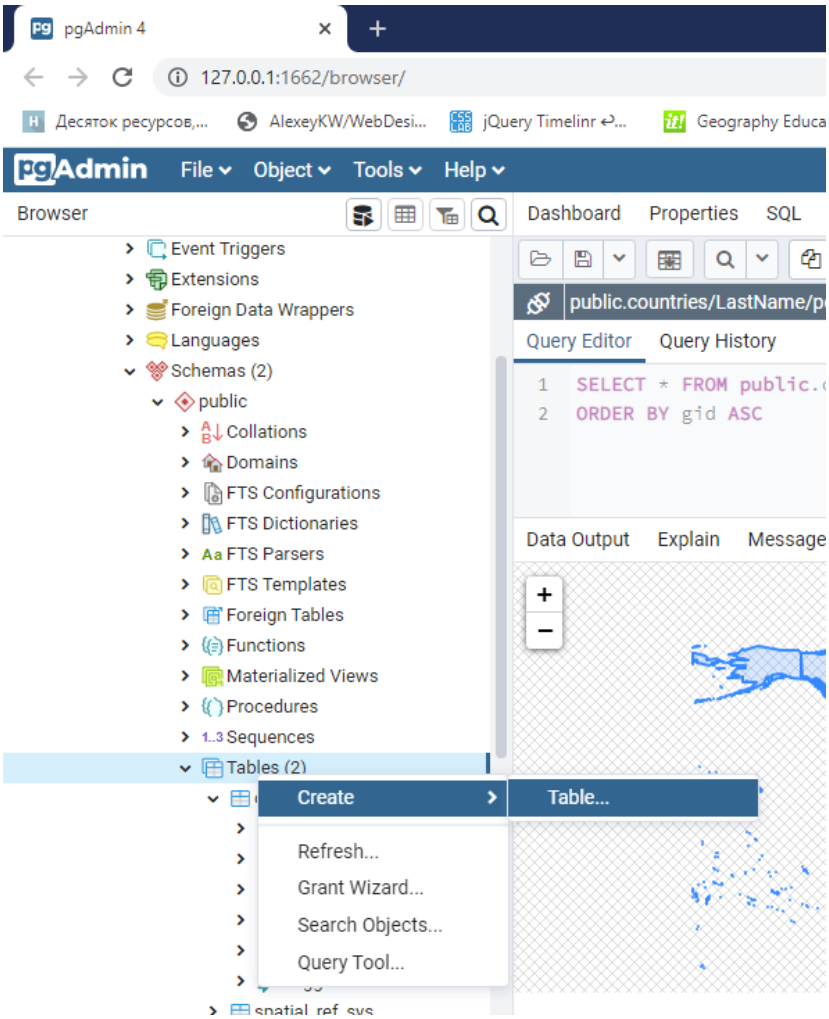


Рис. 6.13. Создание новой таблицы в pgAdmin

Для импорта данных из CSV-файла используется команда *COPY* с указанием имени таблицы, куда будут записаны данные, имени файла-источника, разделителя колонок и наличия заголовка в файле. Команда указывается в новом окне запроса *Tools – Query Tool*. Также в указании пути вместо одинарного используется двойной слэш (рис. 6.15).

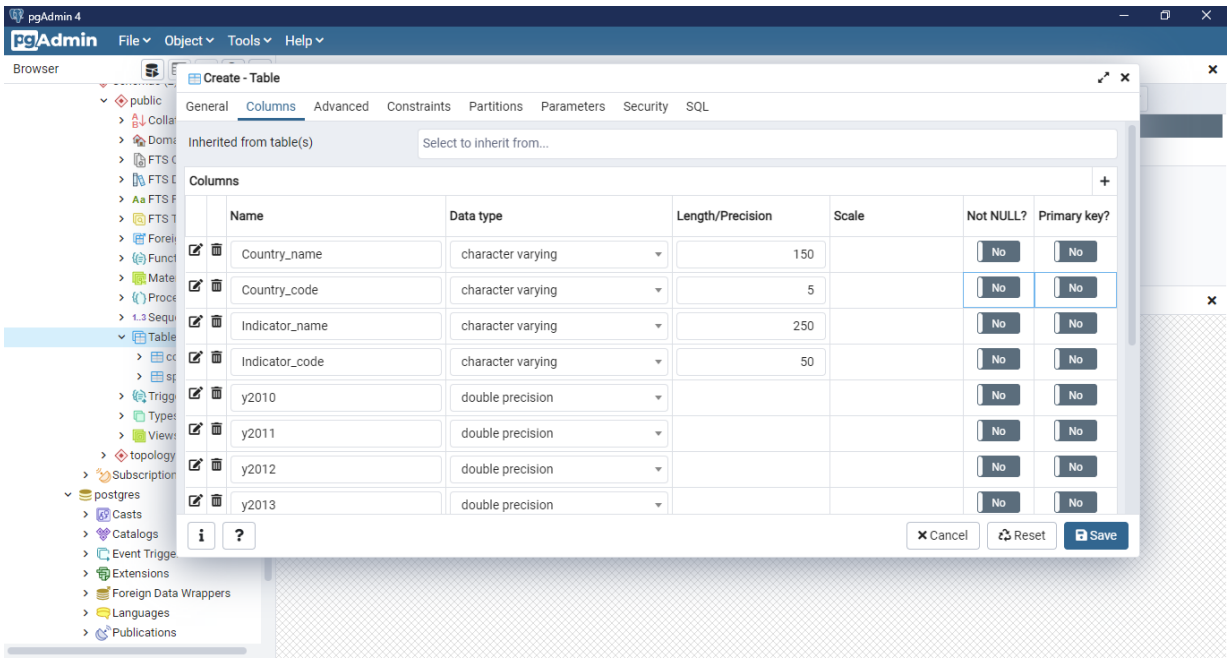


Рис. 6.14. Настройка колонок для создаваемой таблицы

Общая структура команды:

COPY "World_indicators" FROM '\\path\\to\\csv\\file' WITH CSV HEADER DELIMITER ';';

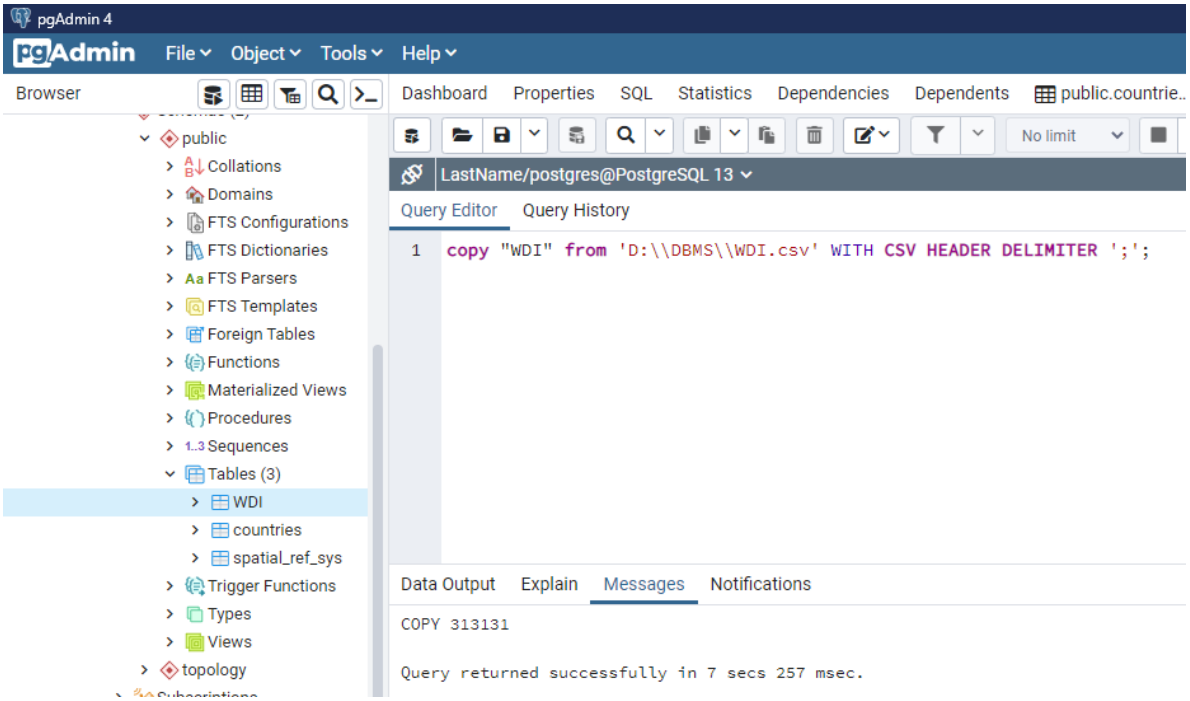


Рис. 6.15. Запрос для импорта CSV-файла в таблицу базы данных

После завершения импорта нужно проверить правильность записанных данных. В интерфейсе pgAdmin это можно сделать через контекстное меню *View / Edit Data* либо составив запрос *SELECT*.

Те же операции нужно сделать для CSV-файла со статистикой по странам мира (*Countries_of_the_world.csv*). Из названий столбцов нужно убрать спецсимволы, а сами названия можно сократить.

Важно обращать внимание на содержимое столбцов CSV-файла и выставлять правильные типы данных у соответствующих колонок таблицы базы данных (рис. 6.16).

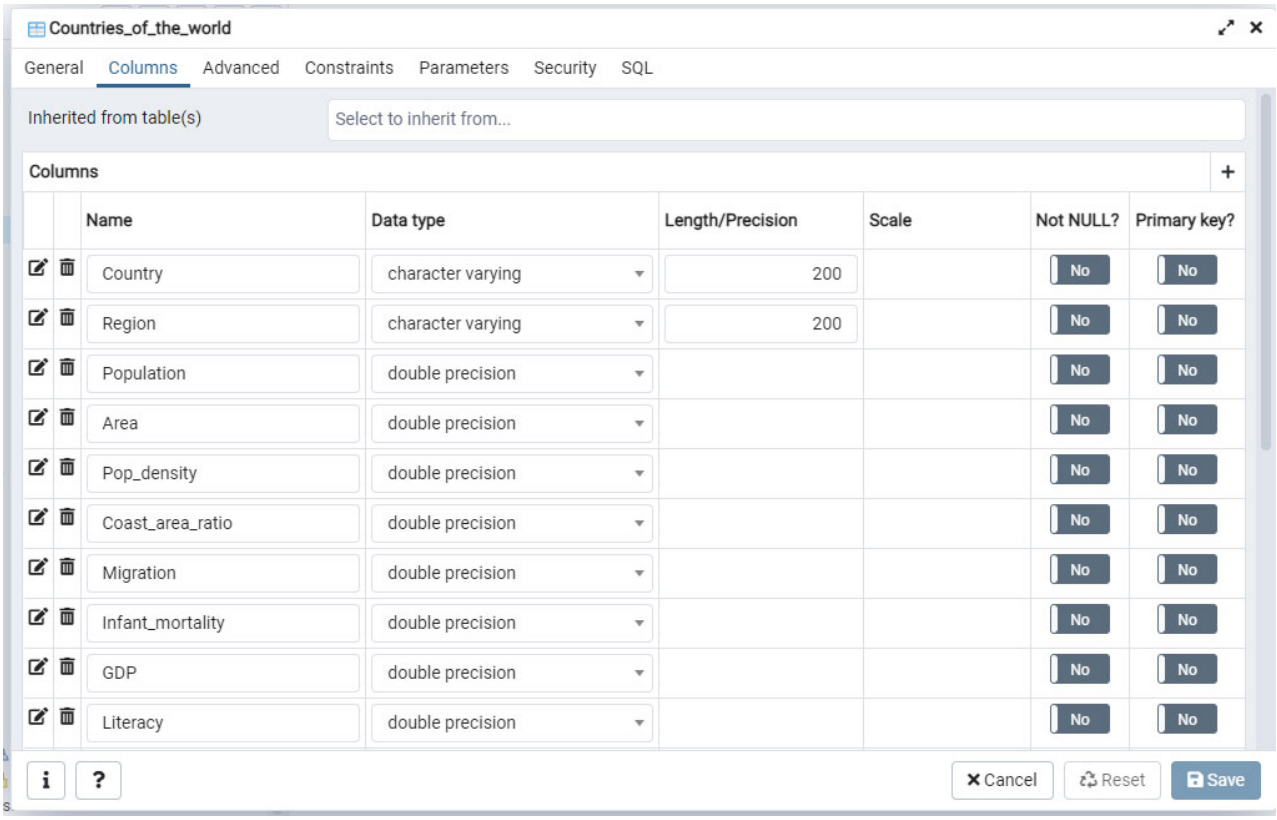


Рис. 6.16. Настройка колонок для таблицы Countries_of_the_world

Запрос на импорт данных аналогичен предыдущему, только разделителем колонок файла в данном случае является запятая; то есть нужно изменить параметр *DELIMETER* либо совсем его убрать (рис. 6.17).

После завершения импорта нужно проверить правильность записанных данных о странах мира (рис. 6.18).

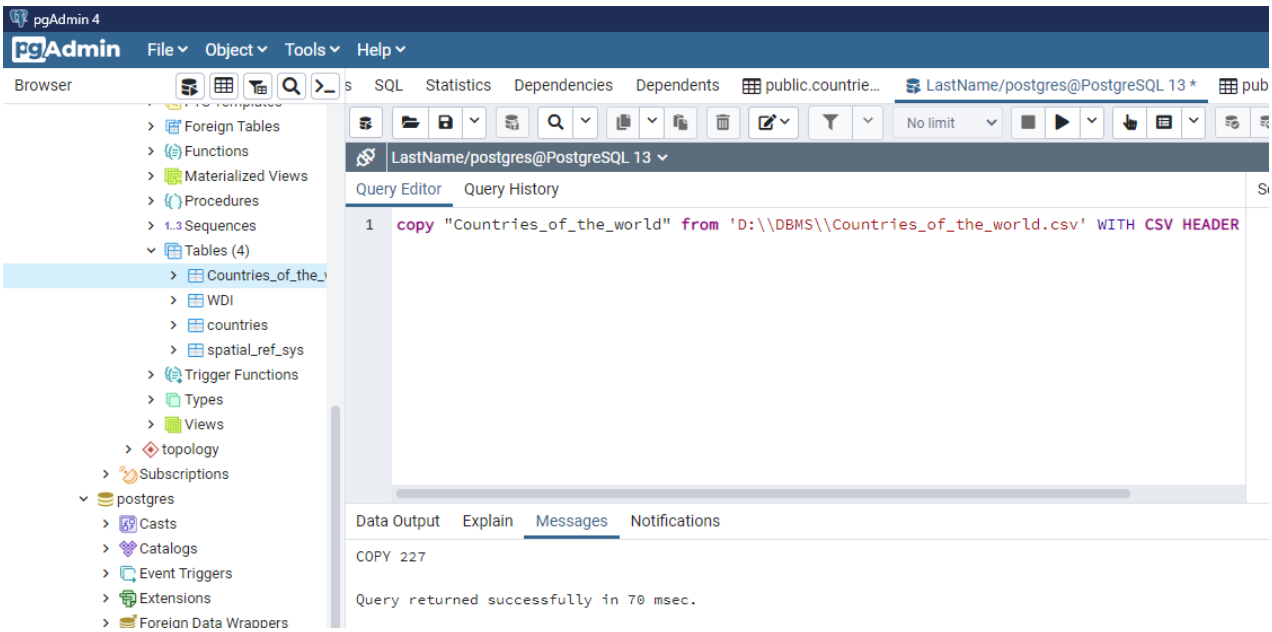


Рис. 6.17. Запрос для импорта файла *Countries_of_the_world.csv*

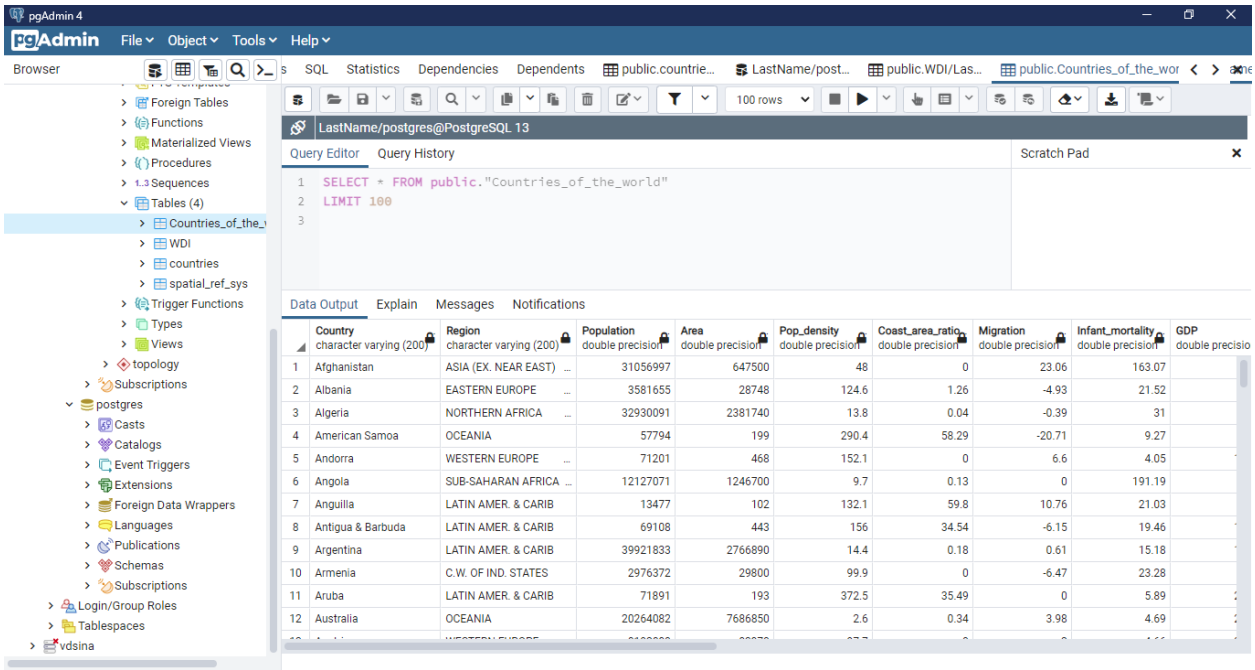


Рис. 6.18. Просмотр результатов импорта

Для импорта городов сначала выполняются стандартные операции по импорту CSV-файла, аналогичные предыдущим. Нужно также обратить внимание на то, что в этом файле разделителем колонок является запятая.

Затем нужно создать в таблице с городами столбец с геометрий точечной локализации (тип столбца *geometry*) и указанием EPSG-кода системы координат (рис. 6.19):

ALTER TABLE "Cities" ADD COLUMN geom geometry(Point, 4326);

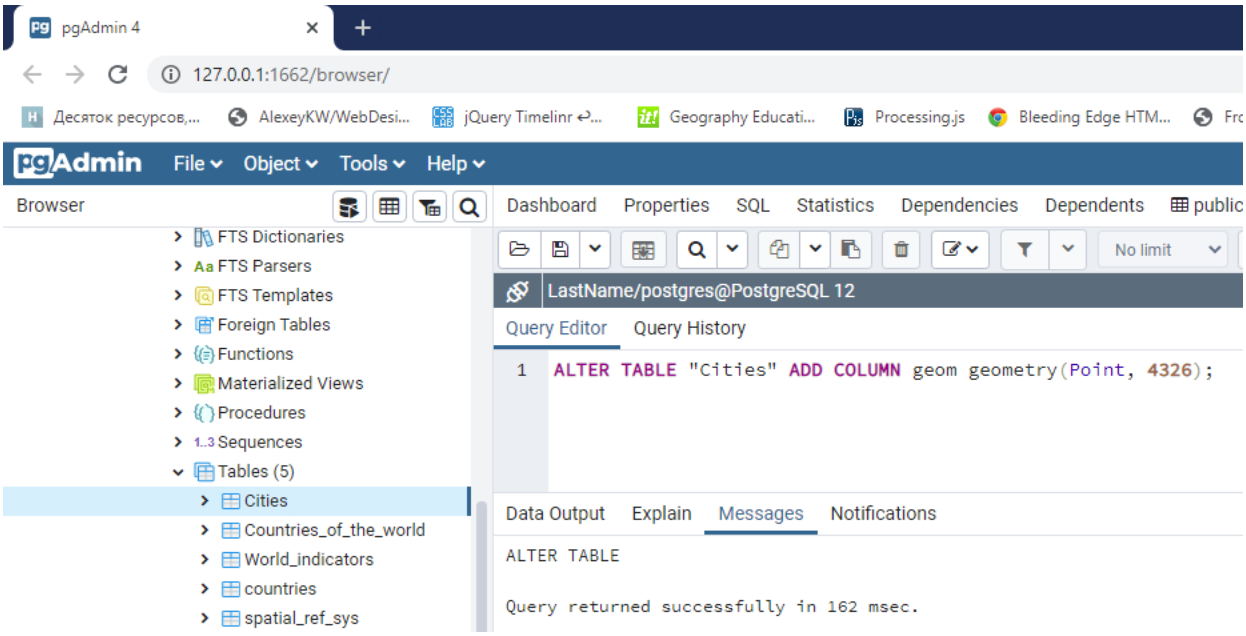


Рис. 6.19. Запрос на создания столбца для хранения координат объектов

Сами объекты создаются командой *ST_MakePoint* с выбором колонок, содержащих координаты, и также указанием системы координат (рис. 6.20).

UPDATE "Cities" SET geom = ST_SetSRID(ST_MakePoint("Longitude", "Latitude"), 4326);

Для проверки правильности создания точечных объектов также нужно воспользоваться QGIS или Geometry Viewer (во втором случае из-за большого количества объектов рекомендуется для отображения брать первые или последние 100 объектов *View/Edit Data – First/Last 100 Rows*) (рис. 6.21).

Для загрузки данных о демографии из файла SQLite можно воспользоваться генератором SQL-скриптов QGIS. Для его использования нужно вызвать контекстное меню слоя и выбрать пункт *Экспорт – Сохранить объекты как*. В открывшемся диалоге указать формат *PostgreSQL SQL дамп*, указать имя и местоположение создаваемого файла (рис. 6.22).

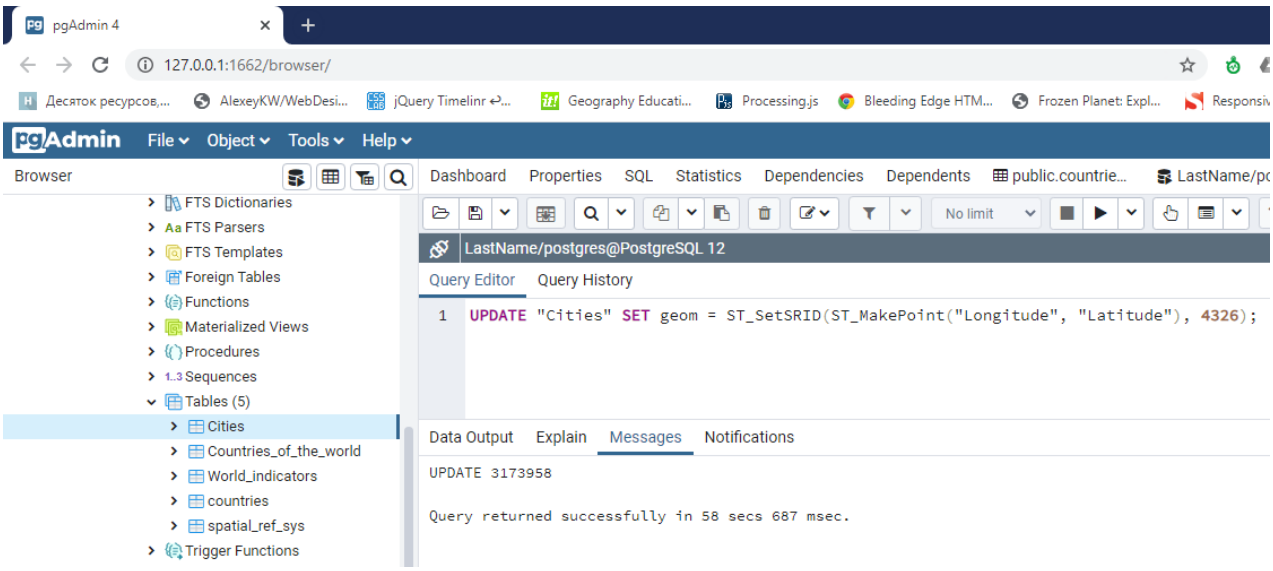


Рис. 6.20. Запрос на создание точечных объектов
на основе данных в таблице

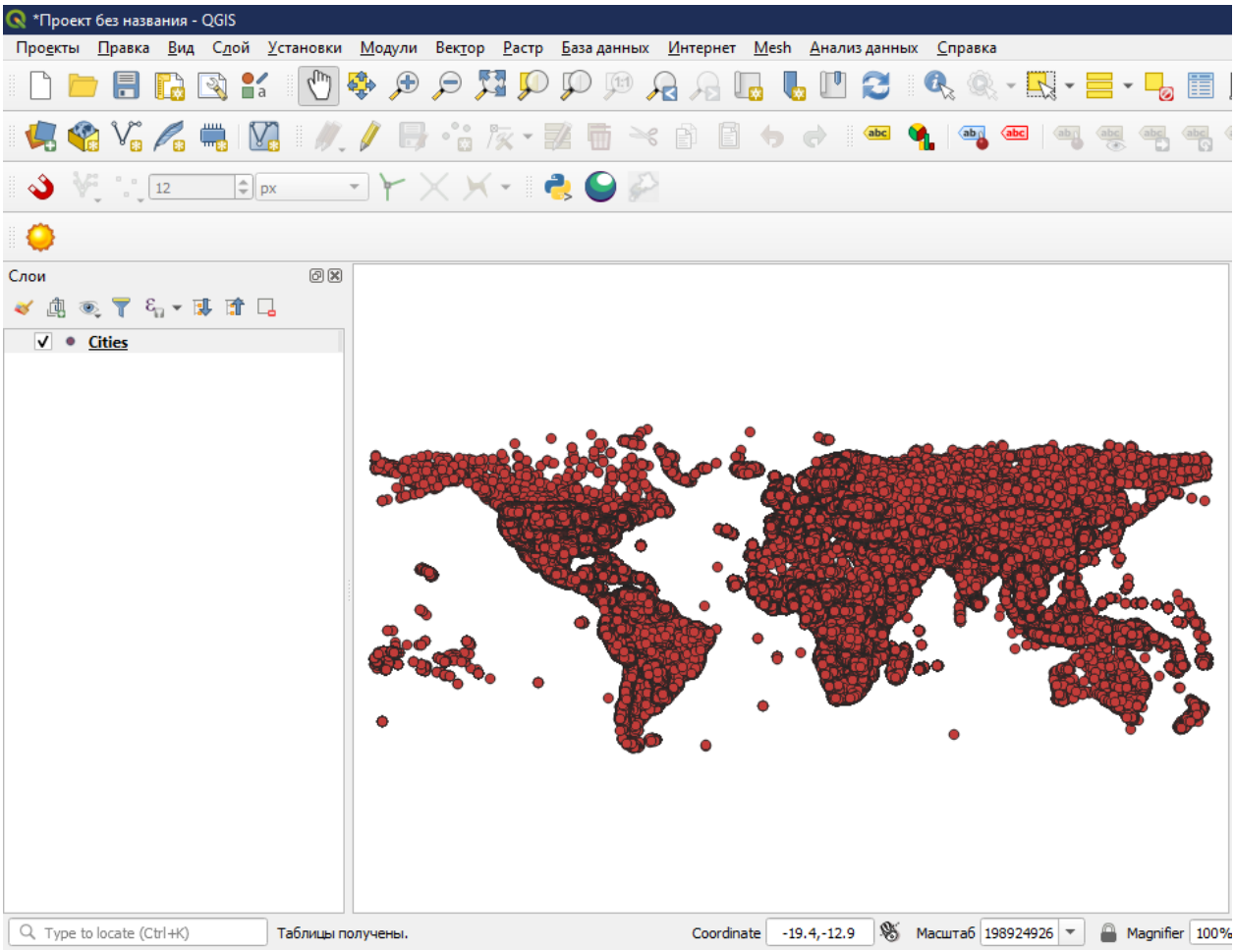


Рис. 6.21. Проверка корректности создания точечных объектов в QGIS

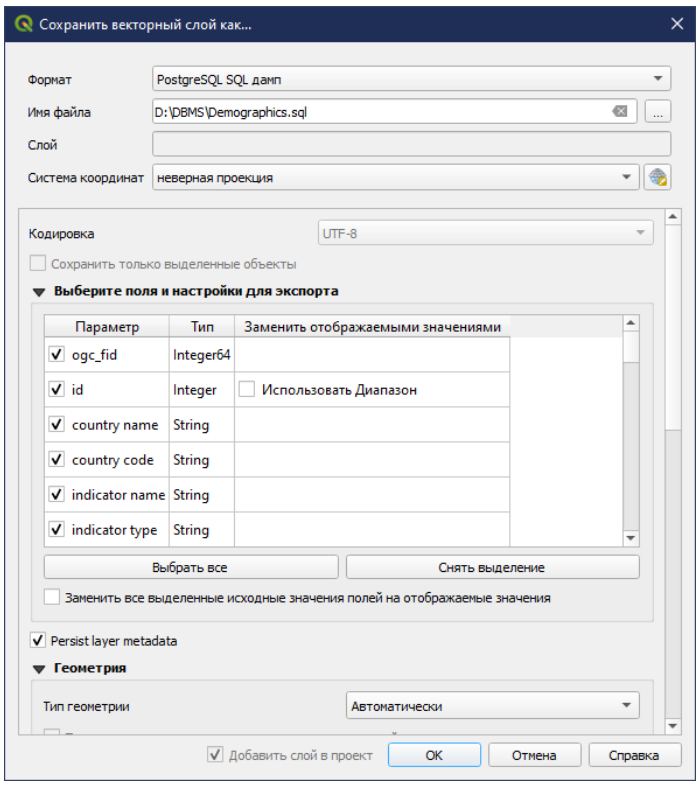


Рис. 6.22. Параметры выгрузки таблицы *Demographics* в SQL-файл

После сохранения файла нужно перейти в pgAdmin, открыть новое окно запросов (*Query Tool*) и загрузить SQL-скрипт с помощью кнопки *Open File* (рис. 6.23).

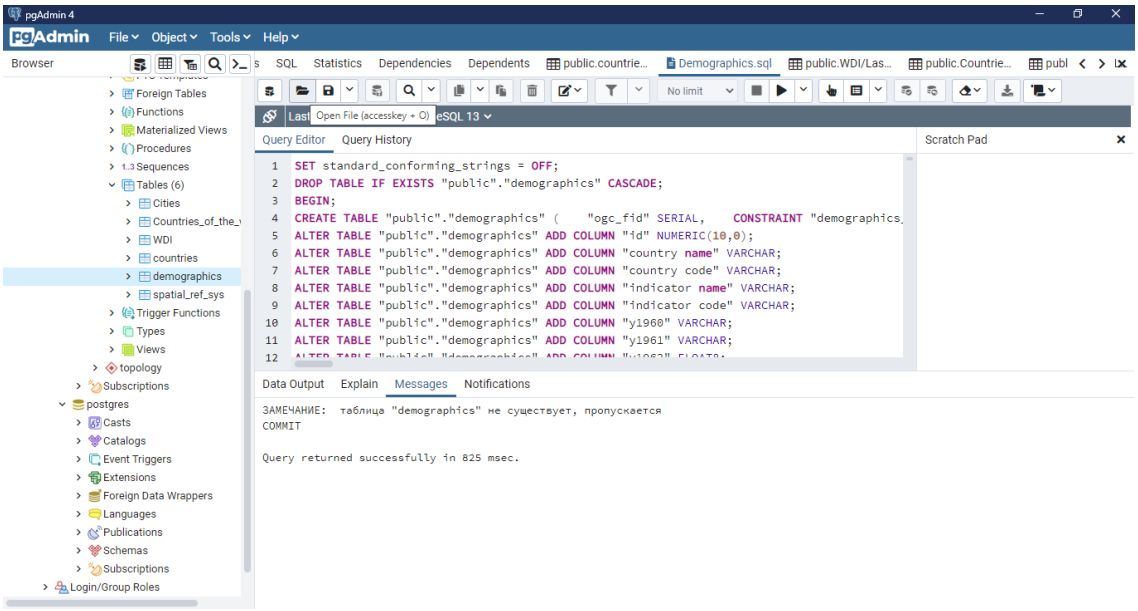


Рис. 6.23. Загрузка SQL-скрипта в pgAdmin

Затем выполнить скрипт и проверить правильность структуры таблицы и данных (рис. 6.24).

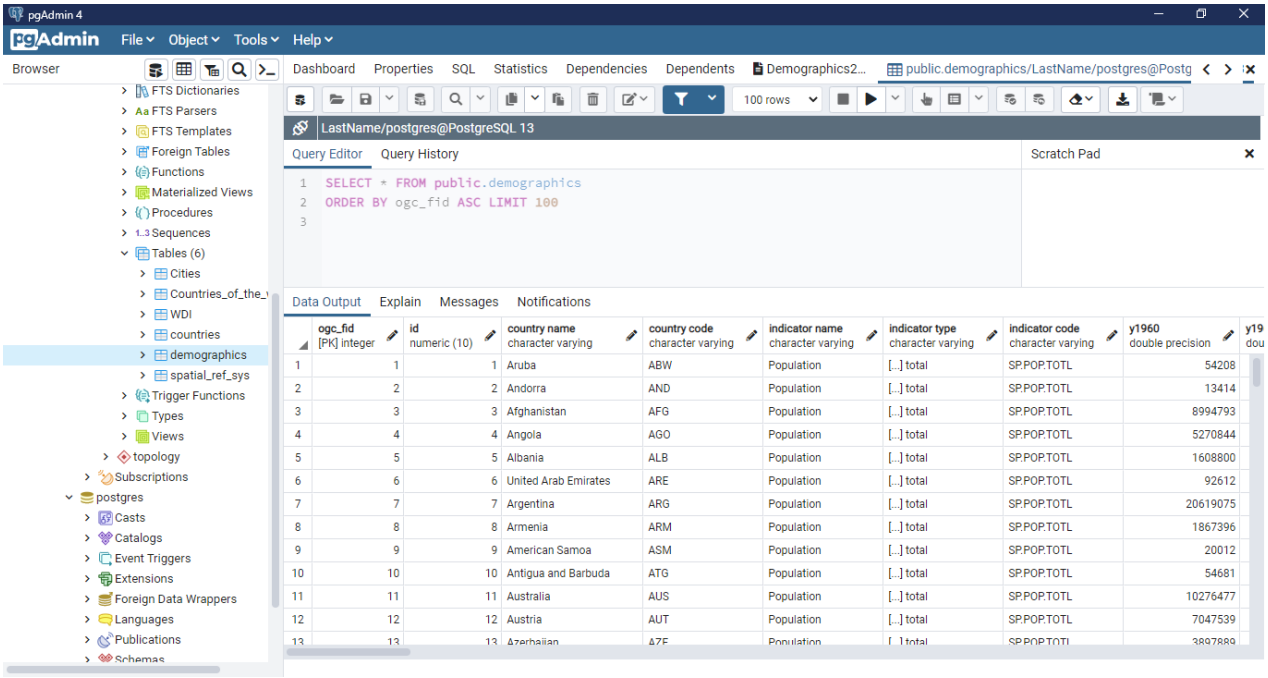


Рис. 6.24. Проверка результатов импорта таблицы *Demographics* из SQL-файла

6.4. Оценка производительности запроса и создание индексов

Составить запрос на выборку всех населенных пунктов с нулевым населением и оценить скорость выполнения этого запроса, используя *EXPLAIN ANALYZE*. Для этого нужно указать ключевые слова *EXPLAIN ANALYZE* перед текстом запроса. Дополнить результаты аналогичного запроса из таблицы практической работы № 1, указав значения Planning time и Execution time (табл. 6.1).

Таблица 6.1

Статистика времени выполнения запроса

Программное обеспечение и параметры выполнения	Время выполнения запроса на поиск городов с ненулевым населением, с
QGIS	...
PostgreSQL Planning time	
PostgreSQL Execution time	

Далее создадим индекс для таблицы с городами и оценим влияние индексов на скорость выполнения запроса. Для этого в контекстном меню категории *Indexes* выбираем *Create – Index* (рис. 6.25).

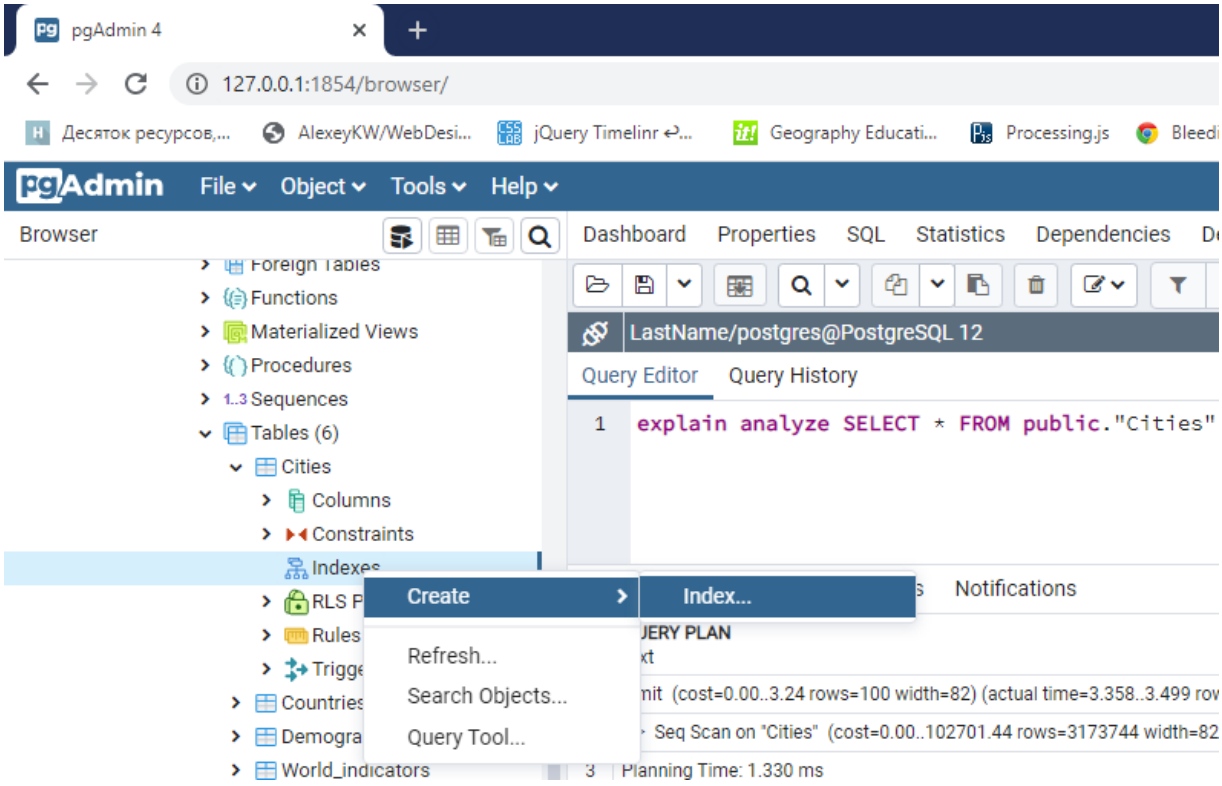


Рис. 6.25. Контекстное меню создания индекса

В открывшемся диалоге указывается название индекса. Затем на вкладке *Definition* выбирается тип индекса (нужно оставить стандартный *B-Tree*) и колонка, для которой создается индекс. В данном случае это колонка с количеством населения города – *Population* (рис. 6.26).

Оценить скорость выполнения предыдущего запроса на поиск населенных пунктов с нулевым населением после создания индекса, также используя *EXPLAIN ANALYZE*, и дополнить табл. 6.2 новыми измерениями.

Для визуализации результатов запросов, выполняемых PostgreSQL в QGIS, можно использовать вставку данных в новые таблицы либо воспользоваться представлениями (View). Рассмотрим второй вариант.

Для отображения в QGIS представления с геоданными (также, как и сами исходные таблицы) должны содержать ключевое поле (*PRIMARY*

KEY). Первичный ключ к таблице с городами добавляется следующим запросом:

ALTER TABLE cities ADD COLUMN id SERIAL PRIMARY KEY;

Если при создании таблица была названа с большой буквы (например, *Cities*), то в запросах имя этой таблицы нужно брать в двойные кавычки (“*Cities*”) (рис. 6.27).

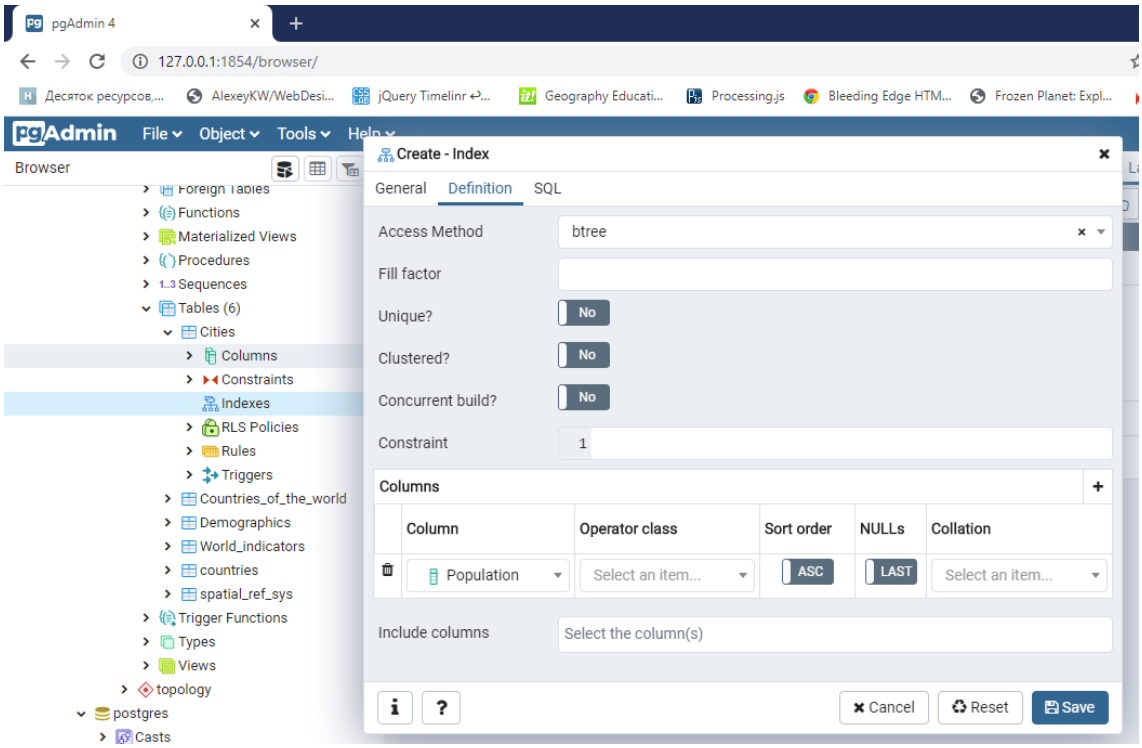


Рис. 6.26. Диалог настройки параметров индекса

Таблица 6.2

Статистика времени выполнения запроса при использовании индексов

Программное обеспечение и параметры выполнения	Время выполнения запроса на поиск городов с ненулевым населением, с
QGIS	...
PostgreSQL Planning time	...
PostgreSQL Execution time	...
PostgreSQL Planning time после создания индекса	
PostgreSQL Execution time после создания индекса	

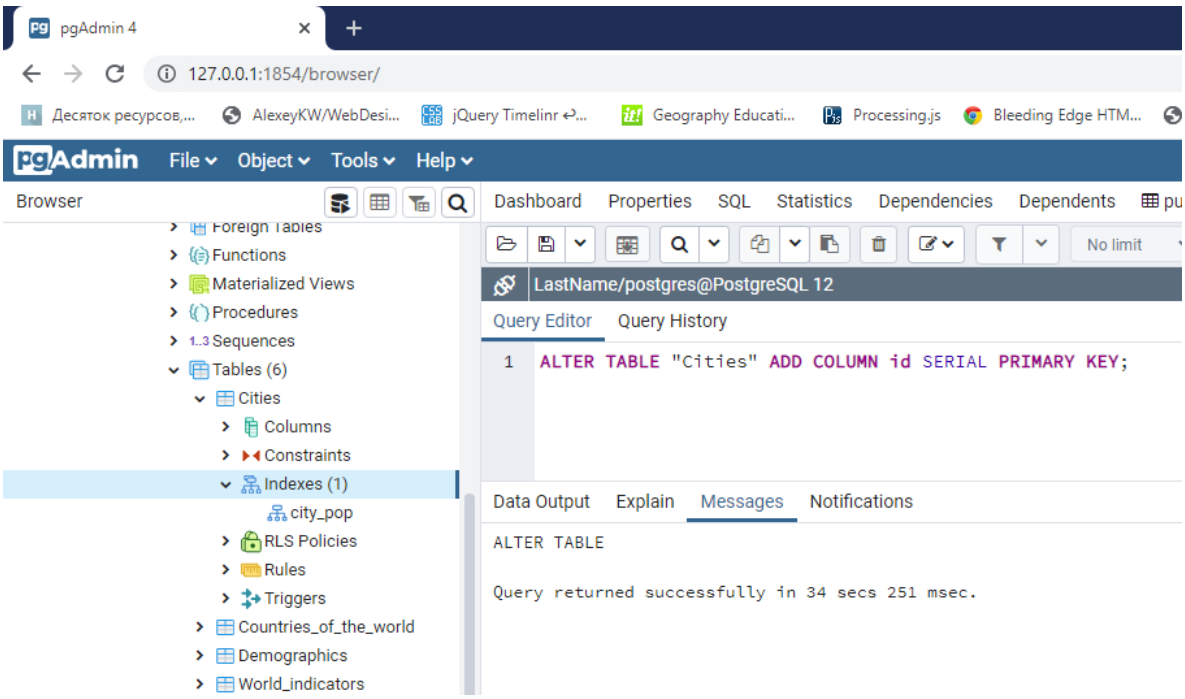


Рис. 6.27. Запрос на создание и автоматическое заполнение ключевого поля

Далее переходим к созданию представления: открываем контекстное меню в категории *Views*, далее *Create – View* (рис. 6.28).

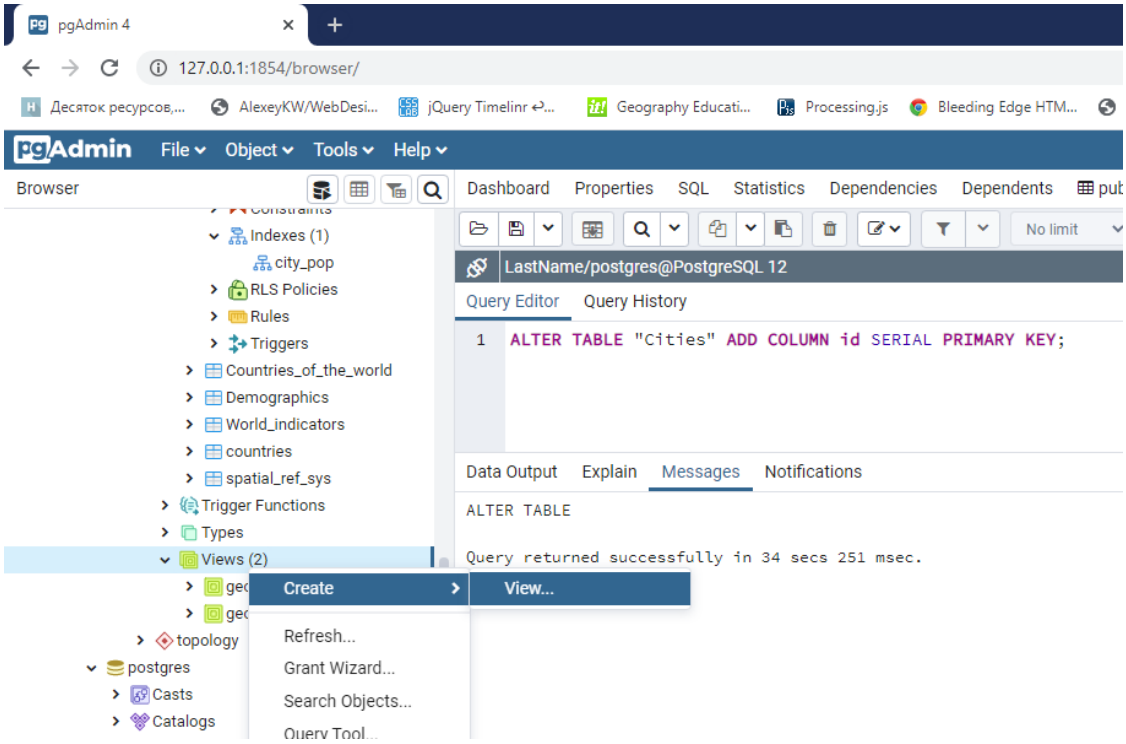


Рис. 6.28. Контекстное меню создания представления

Имя представления указывается произвольное. В поле *Code* прописывается запрос *SELECT* (рекомендуется заранее проверить его правильность в отдельном окне *Query Tool*) (рис. 6.29).

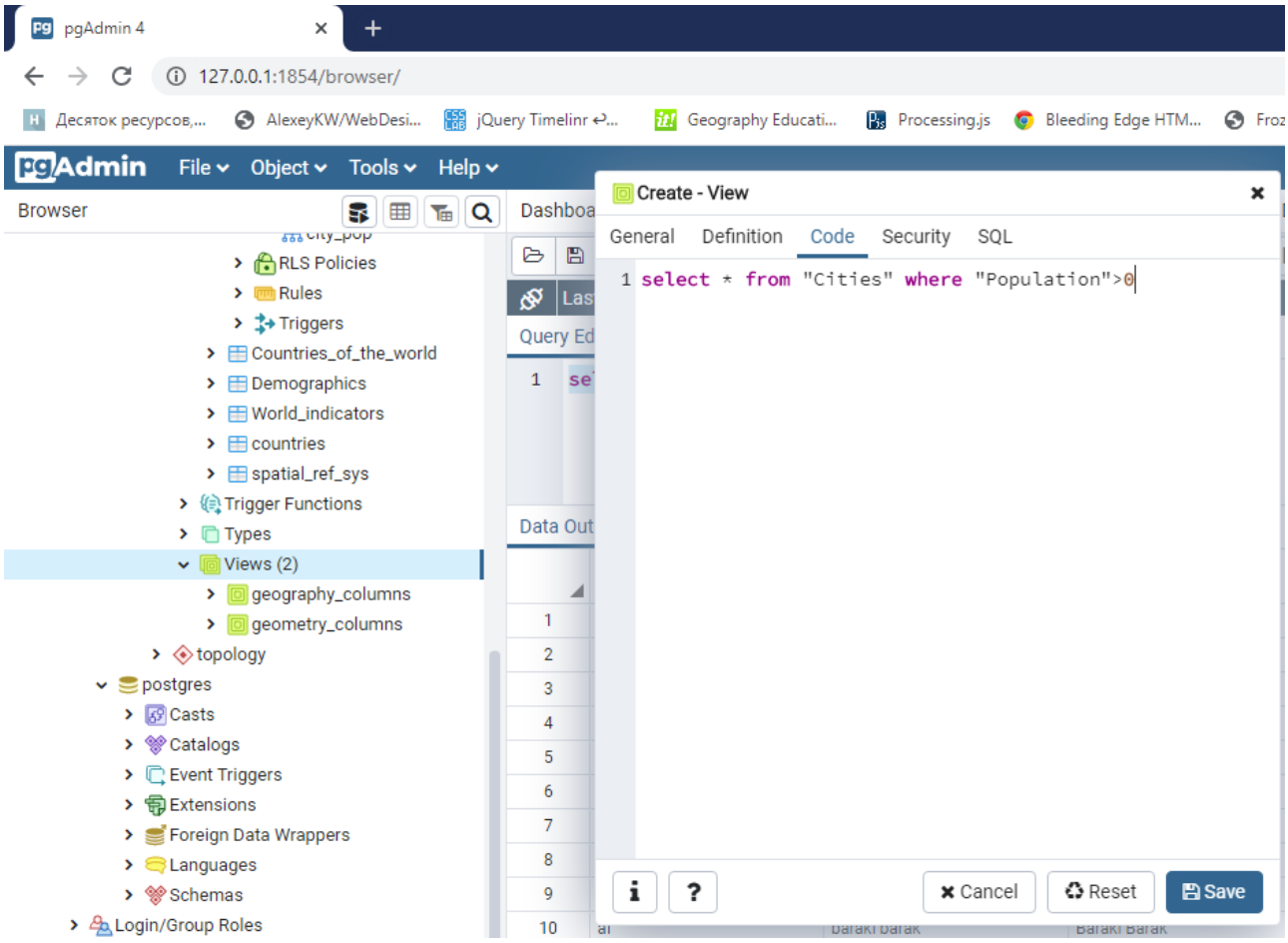


Рис. 6.29. Диалог указания SQL-запроса для отбора данных представления

После подключения к базе данных в общем списке должно появиться созданное представление, и для добавления его как слоя QGIS нужно в списке *ID объекта* выбрать ранее созданный ключ (рис. 6.30 и 6.31).

Далее оценим влияние индексов на выполнение пространственных запросов. Для этого с помощью функции *ST_Contains* (или *ST_Within*) нужно выбрать города, которые имеют ненулевое население (из исходной таблицы *Cities*, а не из представления) и находятся на территории России (геометрия из таблицы *Countries*).

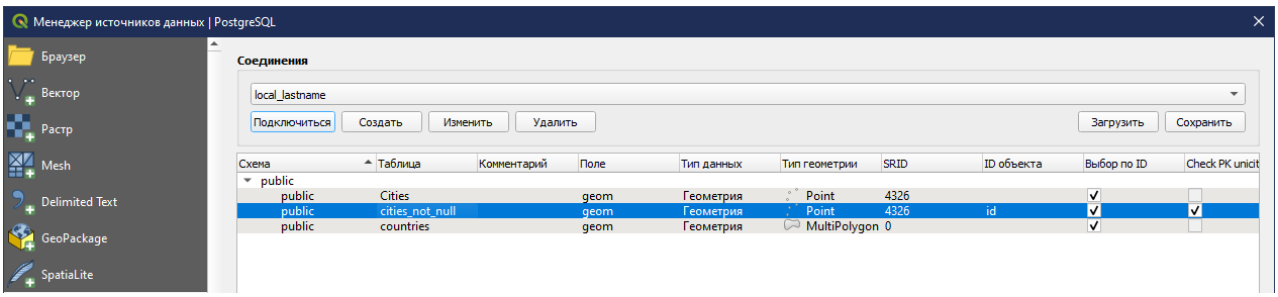


Рис. 6.30. Диалог настройки параметров подключения представления в QGIS

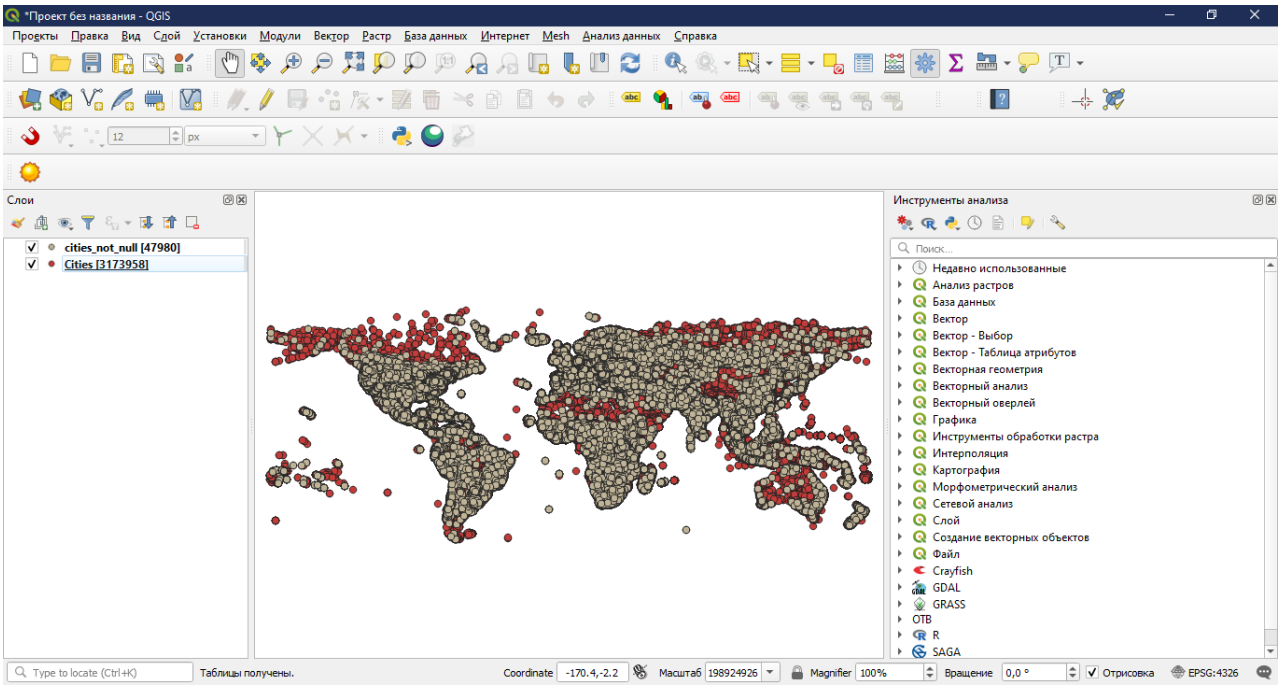


Рис. 6.31. Просмотр представления в QGIS

Для выполнения этого запроса требуется, чтобы у всех таблиц были указаны идентичные системы координат, поэтому у таблицы с контурами стран (*Countries*) потребуется с помощью запроса прописать конкретную систему координат WGS84 – *SELECT UpdateGeometrySRID('countries','geom',4326);*.

Запрос нужно выполнить три раза: для исходной таблицы; для таблицы с индексом *B-Tree*, созданным для колонки с геометрией; для таблицы с индексом *GIST*, созданным для колонки с геометрией. Результаты нужно занести в табл. 6.3.

Таблица 6.3

Статистика времени выполнения запроса при использовании индексов

Программное обеспечение и параметры выполнения	Время выполнения запроса на поиск городов внутри указанной страны, с
QGIS	
PostgreSQL Planning time	
PostgreSQL Execution time	
PostgreSQL Planning time после создания индекса B-Tree	
PostgreSQL Execution time после создания индекса B-Tree	
PostgreSQL Planning time после создания индекса GIST	
PostgreSQL Execution time после создания индекса GIST	

6.5. Индивидуальные задания

Запрос № 1. Выбрать города с ненулевым населением в указанном радиусе от определенного города. Для расчета дистанции используется функция *ST_DistanceSphere(поле_с_геометрией_анализируемых_объектов, точка_отсчета).* Для использования вложенных запросов в функции *ST_DistanceSphere* их нужно брать в круглые скобки. Город и радиус поиска взять на основе номера варианта (табл. 6.4). В отчет нужно вставить скриншот, текст составленного запроса и таблицу с результатами его работы.

Таблица 6.4

Варианты для запроса № 1

Номер варианта	Город	Радиус поиска, км
1	Санкт-Петербург	100
2	Владимир	200
3	Рига	85
4	Прага	90
5	Мюнхен	95
6	Милан	70
7	Ташкент	90
8	Красноярск	150
9	Анкара	150

Окончание табл. 6.4

Номер варианта	Город	Радиус поиска, км
10	Марракеш	120
11	Бахрейн	150
12	Сан-Паулу	105
13	Буэнос-Айрес	115
14	Лима	300
15	Богота	145
16	Порту	105
17	Барселона	80
18	Бейрут	150
19	Верона	85
20	Триест	120
21	Сараево	100
22	Варна	90
23	Самарканд	80
24	Гуанчжоу	50
25	Бангкок	145
26	Ченнай	120
27	Бангалор	90
28	Карачи	135
29	Катар	135
30	Сан-Марино	75

Запрос № 2. Найти минимальное значение и относящуюся к нему страну для указанных показателя и региона мира на основе таблиц *Countries_of_the_world* и *WDI* (нужно обратить внимание на написание названия региона в БД). Показатель и регион взять на основе номера варианта (табл. 6.5). В отчет нужно вставить скриншот, текст составленного запроса и таблицу с результатами его работы.

Таблица 6.5

Варианты для запроса № 2

Номер варианта	Показатель	Регион мира
1	Доступ к чистому топливу и технологиям для приготовления пищи	Asia (ex. Near East)
2	Доступ к электричеству для городской территории	Eastern Europe
3	Процент подростков, не посещающих школу	Near East
4	Процент площади сельскохозяйственных земель	Western Europe
5	Количество пассажиров, перевезенных воздушным транспортом	Asia (ex. Near East)
6	Годовой рост добавленной стоимости сельского, лесного и рыбного хозяйства, %	Latin Amer. & Carib

Окончание табл. 6.5

Номер варианта	Показатель	Регион мира
7	Доступ к электричеству для сельских населенных пунктов	Sub-Saharan Africa
8	Использование альтернативной и ядерной энергии, % от общего потребления	Northern Africa
9	Общий годовой забор пресной воды, млрд $м^3$	Oceania
10	Продукция аквакультуры (марикультуры), т	Latin Amer. & Carib
11	Пашни, га / чел.	C.W. Of Ind. States
12	Количество видов птиц, находящихся под угрозой исчезновения	Northern America
13	Продукция рыболовства, т	Near East
14	Производство зерновых, т	Northern Africa
15	Сумма сборов за использование интеллектуальной собственности (платежный баланс в долларах США)	Northern Africa
16	Количество работников здравоохранения на 1 000 человек	Sub-Saharan Africa
17	Продолжительность обязательного образования, лет	C.W. Of Ind. States
18	Охват программами социального страхования, в % от всего населения	Near East
19	Производство электроэнергии на основе угля, % от общего объема	C.W. Of Ind. States
20	Производство электроэнергии из гидроэнергетических источников, % от общего объема	Baltics
21	Занятость в сельском хозяйстве (смоделированная оценка ILO), % от общей занятости	Northern America
22	Импорт энергии, нетто, % от использованной энергии	Oceania
23	Фирмы, занимающиеся (инвестирующие) в Научно-исследовательские и опытно-конструкторские работы (НИОКР), % всех фирм	Western Europe
24	Количество видов рыб, находящихся под угрозой исчезновения	Baltics
25	Количество абонентов фиксированной телефонной связи (на 100 человек)	Northern America
26	Площадь лесов, % от площади всех земель	Latin Amer. & Carib
27	Уровень грамотности всего взрослого населения для людей в возрасте 15 лет и старше, %	Baltics
28	Процент физических лиц, пользующихся интернетом, относительно всего населения	Asia (ex. Near East)
29	Экспорт средних и высокотехнологичных товаров, % от всего экспорта промышленных товаров	Eastern Europe
30	Количество абонентов сотовой связи (на 100 человек)	Eastern Europe

Запрос № 3. Вычислить количество городов с ненулевым населением, находящихся в пределах указанной страны. Страну взять на основе номера варианта (табл. 6.6). В отчет нужно вставить скриншот, текст составленного запроса и итоговое число.

Таблица 6.6

Варианты для запроса № 3

Номер варианта	Страна
1	Kenya
2	Albania
3	Nepal
4	Guatemala
5	Pakistan
6	Indonesia
7	Finland
8	Hungary
9	Denmark
10	Norway
11	Canada
12	Colombia
13	Bolivia
14	Austria
15	Liechtenstein
16	Bulgaria
17	Algeria
18	Qatar
19	Armenia
20	Romania
21	Philippines
22	Argentina
23	Portugal
24	Croatia
25	Bahrain
26	Estonia
27	Mexico
28	Belgium
29	Georgia
30	Brazil

Запрос № 4. С помощью оконной функции составить запрос для сравнения значений определенного показателя за указанный год для всех стран региона и среднего показателя для всего региона в целом за этот же год. Результаты упорядочить по названию страны. Показатель, регион (нужно обратить внимание на написание названия региона в БД) и год взять на основе номера варианта (табл. 6.7). В отчет нужно вставить скриншот, текст составленного запроса и таблицу с результатами его работы.

Таблица 6.7

Варианты для запроса № 4

Номер варианта	Показатель	Регион мира	Год
1	Процент физических лиц, пользующихся интернетом, относительно всего населения	Asia (ex. Near East)	2016
2	Фирмы, занимающиеся (инвестирующие) в НИОКР, % всех фирм	Western Europe	2017
3	Использование альтернативной и ядерной энергии, % от общего потребления	Northern Africa	2014
4	Доступ к электричеству для сельских населенных пунктов	Sub-Saharan Africa	2013
5	Количество видов птиц, находящихся под угрозой исчезновения	Northern America	2012
6	Количество абонентов сотовой связи (на 100 человек)	Eastern Europe	2013
7	Уровень грамотности всего взрослого населения для людей в возрасте 15 лет и старше, %	Baltics	2015
8	Занятость в сельском хозяйстве (смоделированная оценка ILO), % от общей занятости	Northern America	2015
9	Площадь сельскохозяйственных земель, %	Western Europe	2015
10	Экспорт средних и высокотехнологичных товаров, % от всего экспорта промышленных товаров	Eastern Europe	2017
11	Продолжительность обязательного образования, лет	C.W. Of Ind. States	2017
12	Годовой рост добавленной стоимости сельского, лесного и рыбного хозяйства, %	Latin Amer. & Carib	2012
13	Импорт энергии, нетто, % от использованной энергии	Oceania	2016
14	Количество абонентов фиксированной телефонной связи (на 100 человек)	Northern America	2013

Окончание табл. 6.7

Номер варианта	Показатель	Регион мира	Год
15	Производство электроэнергии на основе угля, % от общего объема	C.W. of Ind. States	2013
16	Количество пассажиров, перевезенных воздушным транспортом	Asia (ex. Near East)	2016
17	Охват программами социального страхования, % от всего населения	Near East	2012
18	Производство электроэнергии из гидроэнергетических источников, % от общего объема	Baltics	2014
19	Процент подростков, не посещающих школу	Near East	2014
20	Общий годовой забор пресной воды, млрд $м^3$	Oceania	2015
21	Производство зерновых, т	Northern Africa	2014
22	Площадь лесов, % от площади всех земель	Latin Amer. & Carib	2014
23	Количество видов рыб, находящихся под угрозой исчезновения	Baltics	2012
24	Сумма сборов за использование интеллектуальной собственности (платежный баланс в долларах США)	Northern Africa	2015
25	Доступ к чистому топливу и технологиям для приготовления пищи	Asia (ex. Near East)	2012
26	Продукция рыболовства, в т	Near East	2013
27	Доступ к электричеству для городской территории	Eastern Europe	2013
28	Пашни, га/чел.	C.W. of Ind. States	2017
29	Количество работников здравоохранения на 1 000 человек	Sub-Saharan Africa	2016
30	Продукция аквакультуры (марикультуры), т	Latin Amer. & Carib	2016

Представление № 1. Составить запрос и создать на его основе представление, которое будет содержать следующие данные:

− полигон, созданный с помощью объединения всех стран (их полигонов), относящихся к указанному региону;

− среднее значение указанного показателя для этого региона за определенный год.

Показатель, год и регион взять на основе номера варианта (табл. 6.8). В отчет нужно вставить скриншот и текст запроса, формирующего представление, таблицу с результатами его работы, скриншот из QGIS с визуализацией данных созданного представления.

Таблица 6.8

Варианты для представления № 1

Номер варианта	Показатель	Регион мира	Год
1	Занятость в сельском хозяйстве (смоделированная оценка ILO), % от общей занятости	Northern America	2015
2	Продолжительность обязательного образования, лет	C.W. of Ind. States	2017
3	Использование альтернативной и ядерной энергии, % от общего потребления	Northern Africa	2014
4	Количество абонентов фиксированной телефонной связи (на 100 человек)	Northern America	2013
5	Процент подростков, не посещающих школу	Near East	2014
6	Сумма сборов за использование интеллектуальной собственности (платежный баланс в долларах США)	Northern Africa	2015
7	Годовой рост добавленной стоимости сельского, лесного и рыбного хозяйства, %	Latin Amer. & Carib	2012
8	Уровень грамотности всего взрослого населения для людей в возрасте 15 лет и старше, %	Baltics	2015
9	Количество видов птиц, находящихся под угрозой исчезновения	Northern America	2012
10	Производство электроэнергии из гидроэнергетических источников, % от общего объема	Baltics	2014
11	Доступ к электричеству для городской территории	Eastern Europe	2013
12	Доступ к чистому топливу и технологиям для приготовления пищи	Asia (ex. Near East)	2012
13	Производство зерновых, т	Northern Africa	2014
14	Площадь лесов, % от площади всех земель	Latin Amer. & Carib	2014
15	Пашни, га/чел.	C.W. of Ind. States	2017
16	Производство электроэнергии на основе угля, % от общего объема	C.W. of Ind. States	2013
17	Охват программами социального страхования, % от всего населения	Near East	2012
18	Количество пассажиров, перевезенных воздушным транспортом	Asia (ex. Near East)	2016
19	Количество работников здравоохранения на 1 000 человек	Sub-Saharan Africa	2016
20	Продукция рыболовства, т	Near East	2013
21	Процент площади сельскохозяйственных земель	Western Europe	2015
22	Продукция аквакультуры (марикультуры), т	Latin Amer. & Carib	2016

Окончание табл. 6.8

Номер варианта	Показатель	Регион мира	Год
23	Процент физических лиц, пользующихся интернетом, относительно всего населения	Asia (ex. Near East)	2016
24	Количество видов рыб, находящихся под угрозой исчезновения	Baltics	2012
25	Экспорт средних и высокотехнологичных товаров, % от всего экспорта промышленных товаров	Eastern Europe	2017
26	Общий годовой забор пресной воды, млрд $м^3$	Oceania	2015
27	Доступ к электричеству для сельских населенных пунктов	Sub-Saharan Africa	2013
28	Фирмы занимающиеся (инвестирующие) в НИОКР, % всех фирм	Western Europe	2017
29	Импорт энергии, нетто, % от использованной энергии)	Oceania	2016
30	Количество абонентов сотовой связи (на 100 человек)	Eastern Europe	2013

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Постановка задачи.
4. Описание выполненной работы.
5. Описание выполнения индивидуальных заданий.

Вопросы и задания для защиты работы

1. Опишите основные агрегирующие функции в SQL-запросах.
2. Опишите принципы оценки производительности SQL-запросов.
3. Перечислите основные этапы загрузки векторных данных в реляционную базу данных.
4. Перечислите основные этапы загрузки табличных данных в реляционную базу данных.
5. Опишите основные параметры, которые указываются при создании представлений в реляционных базах данных.
6. Опишите основные параметры, которые указываются при создании индексов.

ЗАКЛЮЧЕНИЕ

Учебно-методическое пособие «Базы пространственных данных» позволяет обучающимся получить теоретические базовые знания и практические навыки по основным разделам систем управления данными. Ознакомиться с принципами использования систем управления базами пространственных данных, реализующих новые информационные технологии, и изучить современные программные средства работы с базами пространственных данных. Изучить методы составления атрибутивных и пространственных запросов к данным для применения в профессиональной деятельности. Получить сведения об использовании программного обеспечения для хранения, обработки и анализа пространственных данных на основе типовых этапов формирования базы пространственных данных: импорт внешних данных в систему управления пространственными данными – выполнение аналитических атрибутивных и пространственных запросов – визуализация результатов с помощью геоинформационной системы.

Изложенные в учебно-методическом пособии материалы являются теоретической и практической основой для выполнения лабораторных работ по дисциплине «Базы пространственных данных».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дейт К. Жд. Введение в системы баз данных / пер. с англ. К. А. Птицын. – 8-е изд. – М. : Вильямс, 2018. – 1328 с.

2. Базы данных в картографии и геоинформатике : учеб.-метод. пособие / А. А. Колесников [и др.]. – Новосибирск : СГУГиТ, 2018. – 63 с.

3. Кузовкин А. В., Цыганов А. А., Щукин Б. А. Управление данными : учеб. для вузов, допущено УМО. – М. : Академия, 2010. – 256 с.

4. Новиков Б. А., Горшкова Е. А., Графеева Н. Г. Основы технологий баз данных : учеб. пособие / под ред. Е. В. Рогова. – 2-е изд. – М. : ДМК Пресс, 2020. – 582 с.

5. Осипов Д. Л. Технологии проектирования баз данных. – М. : ДМК Пресс, 2019. – 498 с.

6. Волк В. К. Базы данных. Проектирование, программирование, управление и администрирование : учеб. для вузов. – 2-е изд., стер. – СПб : Лань, 2021. – 244 с.

7. Григорьев Ю. А., Плутенко А. Д., Плужникова О. Ю. Реляционные базы данных и системы NoSQL : учеб. пособие. – Благовещенск : АмГУ, 2018. – 424 с.

Учебное издание

Колесников Алексей Александрович
Бугаков Петр Юрьевич

БАЗЫ ПРОСТРАНСТВЕННЫХ ДАННЫХ

Редактор *О. В. Георгиевская*
Компьютерная верстка *О. И. Голиков*

Изд. лиц. ЛР № 020461 от 04.03.1997.
Подписано в печать 21.03.2022. Формат 60 × 84 1/16.
Усл. печ. л. 5,05. Тираж 75 экз. Заказ 40.
Гигиеническое заключение
№ 54.НК.05.953.П.000147.12.02. от 10.12.2002.
Редакционно-издательский отдел СГУГиТ
630108, Новосибирск, ул. Плахотного, 10.
Отпечатано в картопечатной лаборатории СГУГиТ
630108, Новосибирск, ул. Плахотного, 8.