СОДЕРЖАНИЕ

ВВЕДЕНИЕ…….9
ГЛАВА 1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ РАБОТЫ…….11
1.1 Сущность и основные понятия ГИС. Области применения……11
1.1.1 Преимущества геоинформационных систем……...12
1.1.2 Операции, осуществляемые ГИС…….12
1.1.3 Возможности ГИС…….15
1.1.4 Состав ГИС……15
1.1.5 Области применения ГИС……16
1.2 Проблемы информационной безопасности картографических данных……18
1.3 Существующие решения……19
1.4 Архитектура сервера ЗКБД……...22
ГЛАВА 2. ОРГАНИЗАЦИЯ СИСТЕМЫ И ОПИСАНИЕ РЕАЛИЗАЦИИ ПРОЕКТА……..24
2.1 Выбор ГИС……..24
2.2 Тестовая БД. Пространственные запросы……28
2.3 Выбор СУБД……31
ГЛАВА 3. РЕАЛИЗАЦИЯ ДИПЛОМНОГО ПРОЕКТА……...36
3.1 Установка и настройка объектно-реляционной СУБД PostgreSQL…….36
3.1.1 Установка PostGIS…….41
3.2 Реализация модуля расшифровки картографической базы данных……42
3.3 Применение библиотеки TPL к разработанному программному проект…….44
3.4 Импорт картографической базы данных……..47
3.4.1 Импорт с использованием менеджера БД…….47
3.4.2 Импорт данных без предварительного сохранения на жестком диске……...51
ЗАКЛЮЧЕНИЕ…….54

СПИСОК ЛИТЕРАТУРЫ……55
ПРИЛОЖЕНИЕ 1……..57
ПРИЛОЖЕНИЕ 2……..58
ПРИЛОЖЕНИЕ 3……..61
ПРИЛОЖЕНИЕ 4……..64

ВВЕДЕНИЕ

В работе рассматривается организация системы защиты картографических баз данных. Разрабатывается организация импорта защищенных картографических данных, установка и конфигурирование объектно-реляционной системы управления базами данных и геоинформационной системы, разрабатывается программный модуль для удаленного доступа к серверу.

Актуальность Картографическая продукция отличается высокой себестоимостью работ по ее получению: топографами и геодезистами затрачиваются немалые усилия на формирование этой продукции. Кроме того, карты могут содержать конфиденциальные сведения, например, новые места локализации нефтегазовых и урановых месторождений, цветных металлов. Поэтому актуальной задачей является защита картографических данных.

В основе программной части разрабатываемой системы используется программное обеспечение с открытым исходным кодом, что значительно снижает себестоимость системы в целом.

Разработка модулей обработки запросов к защищенным картографическим базам данных может существенно повысить эффективность работы с такими базами данных, увеличить быстродействие при достаточном уровне защиты.

Предметом работы являлось создание модуля расшифровки картографических данных, с целью анализа и увеличения быстродействия обработки запросов.

Задачи В рамках данной работы необходимо решить следующие задачи:

1. Раскрыть актуальность модуля импорта данных системы защиты картографических баз данных.
2. Разработать алгоритм расшифровки данных.
3. Организовать сетевое хранилище на базе объектно-реляционной СУБД PostgreSQL для повышения надежности хранения данных и возможности их восстановления вследствие выхода из строя оборудования.

4. Произвести установку и конфигурирование объектно-реляционной СУБД PostgreSQL для удаленного доступа к хранилищу.

5. Импортировать картографическую базу данных из ГИС QGIS.

6. Разработать программный модуль доступа к серверу и дешифрования файлов.

7. Применить библиотеку TPL к разработанному программному проекту.

ГЛАВА 1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ РАБОТЫ

Производится теоретический обзор предметной области, рассмотрение различных вариантов реализации и возможность их применения в данном проекте.

1.1 Сущность и основные понятия ГИС. Области применения

Геоинформационные системы (далее ГИС) — системы, предназначенные для сбора, хранения, анализа и графической визуализации пространственных данных и связанной с ними информации о представленных в ГИС объектах. Другими словами ГИС – современная компьютерная технология для картографирования и анализа объектов реального мира, происходящих и прогнозируемых событий и явлений. Научные, технические, технологические и прикладные аспекты проектирования, создания и использования ГИС изучаются геоинформатикой.

Если обойтись без определений, а ограничиться описанием, то ГИС объединяет традиционные операции при работе с базами данных - запрос и статистический анализ - с преимуществами полноценной визуализации и географического (пространственного) анализа, которые предоставляет карта. Эта особенность дает уникальные возможности для применения ГИС в решении широкого спектра задач, связанных с анализом явлений и событий, прогнозированием их вероятных последствий, планированием стратегических решений. [21]

Данные в геоинформационных системах хранятся в виде набора тематических слоев, которые объединены на основе их географического положения. Этот гибкий подход и возможность геоинформационных систем работать как с векторными, так и с растровыми моделями данных, эффективен при решении любых задач, касающихся пространственной информации.

Геоинформационные системы тесно связаны с другими информационными системами и используют их данные для анализа объектов.

ГИС - это инструментарий, с помощью которого можно решить такие задачи, для которых до этого не существовало готовых законченных решений. [13]

1.1.1 Преимущества геоинформационных систем

- Удобное для пользователя отображение пространственных данных.

Картографирование пространственных данных, в том числе в трехмерном измерении, наиболее удобно для восприятия, что упрощает построение запросов и их последующий анализ.

- Интеграция данных внутри организации.

Геоинформационные системы объединяют данные, накопленные в различных подразделениях компании или даже в разных областях деятельности организаций целого региона. Коллективное использование накопленных данных и их интеграция в единый информационный массив дает существенные конкурентные преимущества и повышает эффективность эксплуатации геоинформационных систем.

- Принятие обоснованных решений.

Автоматизация процесса анализа и построения отчетов о любых явлениях, связанных с пространственными данными, помогает ускорить и повысить эффективность процедуры принятия решений.

- Удобное средство для создания карт.

Геоинформационные системы оптимизируют процесс расшифровки данных космических и аэросъемок и используют уже созданные планы местности, схемы, чертежи. ГИС существенно экономят временные ресурсы, автоматизируя процесс работы с картами, и создают трехмерные модели местности.

1.1.2 Операции, осуществляемые ГИС

ГИС общего назначения обычно выполняет пять типов задач с данными: ввод, манипулирование, управление, запрос и анализ и визуализацию

Ввод данных. Данные для использования их в геоинформационных системах должны быть преобразованы в подходящий цифровой формат. Процесс преобразования данных с бумажных карт в компьютерные файлы называется оцифровкой. В современных ГИС этот процесс может быть автоматизирован с применением сканерной технологии, что особенно важно при выполнении крупных проектов, либо, при сравнительно небольшом объеме работ, данные можно вводить с помощью дигитайзера. Некоторые ГИС имеют встроенные векторизаторы, автоматизирующие процесс оцифровки растровых изображений. Многие данные уже переведены в форматы, напрямую воспринимаемые ГИС-пакетами.

Манипулирование данными. Часто для выполнения конкретного проекта имеющиеся данные нужно дополнительно видоизменить в соответствии с требованиями системы. Например, географическая информация может быть представлена в разных масштабах (осевые линии улиц представлены в масштабе 1: 100 000, границы округов переписи населения - в масштабе 1: 50 000, а жилые объекты - в масштабе 1: 10 000). Для совместной обработки и визуализации все данные удобнее представить в едином масштабе и в одинаковой картографической проекции. ГИС-технология предоставляет разные способы манипулирования пространственными данными и выделения данных, нужных для решения конкретной задачи.

Управление. В небольших проектах географическая информация может храниться в виде обычных файлов. Но при увеличении объема информации и росте числа пользователей для хранения, структурирования и управления данными эффективнее применять системы управления базами данных, специальные компьютерные средства для работы с интегрированными наборами данных. В ГИС наиболее удобно использовать реляционную структуру, при которой данные хранятся в табличной форме. При этом для связывания таблиц применяются общие поля. Этот простой подход достаточно гибок и широко используется во многих ГИС.

Запрос и анализ. При наличии геоинформационной системы и географической информации можно получать ответы как на простые вопросы (Кто владелец данного земельного участка? На каком расстоянии друг от друга расположены эти объекты? Где расположена данная промышленная зона?), так и на более сложные, требующие дополнительного анализа, запросы (Где есть места для строительства нового дома? Каков основный тип почв под еловыми лесами? Как повлияет на движение транспорта строительство новой дороги?). Запросы можно задавать как простым щелчком мыши на определенном объекте, так и посредством развитых аналитических средств. С помощью ГИС можно выявлять и задавать шаблоны для поиска, проигрывать сценарии по типу "что будет, если...". Современные ГИС имеют множество мощных инструментов для анализа, среди них наиболее значимы два: анализ близости и анализ наложения. Для проведения анализа близости объектов относительно друг друга в ГИС применяется процесс, называемый буферизацией. Он помогает ответить на вопросы типа: Сколько домов находится в пределах 100 м от заданного водоема? Сколько покупателей живет не далее 1км от данного магазина? Какова доля добытой нефти из скважин, находящихся в пределах 10 км от здания управления? Процесс наложения включает интеграцию данных, расположенных в разных тематических слоях. В простейшем случае это операция отображения, но при ряде аналитических операций данные из разных слоев объединяются физически. Наложение, или пространственное объединение, позволяет, например, интегрировать данные о почвах, уклоне, растительности и землевладении со ставками земельного налога.

Визуализация. Для многих типов пространственных операций конечным результатом является представление данных в виде карты или графика. Карта - это очень эффективный и информативный способ хранения, представления и передачи географической информации. Раньше карты создавались на столетия. ГИС предоставляет новые удивительные инструменты, расширяющие и развивающие искусство и научные основы картографии. С ее помощью визуализация самих карт может быть легко дополнена отчетными документами,

трехмерными изображениями, графиками, таблицами, диаграммами, фотографиями и другими средствами, например, мультимедийными.

1.1.3 Возможности ГИС

ГИС-система позволяет:

- определить какие объекты располагаются на заданной территории;
- определить местоположение объекта (пространственный анализ);
- дать анализ плотности распределения по территории какого-то явления (например, плотность расселения);
- определить временные изменения на определенной площади);
- смоделировать, что произойдет при внесении изменений в расположение объектов (например, если добавить новую дорогу). [12]

1.1.4 Состав ГИС

Геоинформационные системы включают в себя пять ключевых составляющих.

1. Аппаратные средства - это компьютер, на котором функционирует ГИС, и все периферийные устройства, которые используются для получения информации, ввода ее в ЭВМ и предоставление пользователю ГИС результатов. В настоящее время ГИС работают на различных типах компьютерных платформ, от централизованных серверов до отдельных или связанных сетью настольных компьютеров

2. Программное обеспечение. ГИС содержит инструменты, необходимые для хранения, анализа и визуализации географической (пространственной) информации. Ключевыми компонентами программных продуктов являются: средства ввода и оперирования географической информацией; система управления базой данных; инструменты поддержки пространственных запросов, анализа и визуализации (отображения); графический пользовательский интерфейс для легкого доступа к инструментам и функциям.

3. Данные - это наиболее важный компонент ГИС. Данные о пространственном положении (географические данные) и связанные с ними табличные или атрибутивные данные могут собираться и подготавливаться самим пользователем, либо приобретаться у поставщиков на коммерческой или другой основе. В процессе обработки данных ГИС интегрирует пространственные данные с другими типами и источниками данных, а также может использовать СУБД, применяемые многими организациями для упорядочивания и поддержки имеющихся в их распоряжении данных.

4. Исполнители. Широкое применение технологии ГИС невозможно без людей, которые работают с программными продуктами и разрабатывают планы их использования при решении реальных задач. Пользователями ГИС могут быть как технические специалисты, разрабатывающие и поддерживающие систему, так и обычные сотрудники (конечные пользователи), которым ГИС помогает решать текущие каждодневные дела и проблемы.

5. Методы. Успешность и эффективность (в том числе экономическая) применения ГИС во многом зависит от правильно составленного плана и правил работы, которые составляются в соответствии со спецификой задач и работы каждой организации.

1.1.5 Области применения ГИС

- Управление земельными ресурсами, земельные кадастры. Для решения проблем, имеющих пространственную привязку, и начали создавать ГИС. Типичные задачи — составление кадастров, классификационных карт, определение площадей участков и границ между ними и т. д.
- Инвентаризация, учет, планирование размещения объектов распределенной производственной инфраструктуры и управление ими. Например, нефтегазодобывающие компании или компании, управляющие энергетической сетью, системой бензоколонок, магазинов и т. п.
- Проектирование, инженерные изыскания, планировка в строительстве, архитектуре. Такие ГИС позволяют решать полный комплекс

задач по развитию территории, оптимизации инфраструктуры строящегося района, требующегося количества техники, сил и средств.

- Тематическое картографирование.
- Управление наземным, воздушным и водным транспортом. ГИС позволяет решать задачи управления движущимися объектами при условии выполнения заданной системы отношений между ними и неподвижными объектами. В любой момент можно узнать, где находится транспортное средство, рассчитать загрузку, оптимальную траекторию движения, время прибытия и т. п.
- Управление природными ресурсами, природоохранная деятельность и экология. ГИС помогает определить текущее состояние и запасы наблюдаемых ресурсов, моделирует процессы в природной среде, осуществляет экологический мониторинг местности.
- Геология, минерально-сырьевые ресурсы, горнодобывающая промышленность. ГИС осуществляет расчеты запасов полезных ископаемых по результатам проб (разведочное бурение, пробные шурфы) при известной модели процесса образования месторождения.
- Чрезвычайные ситуации. С помощью ГИС производится прогнозирование чрезвычайных ситуаций (пожаров, наводнений, землетрясений, селей, ураганов), расчет степени потенциальной опасности и принятие решений об оказании помощи, расчет требуемого количества сил и средств для ликвидации чрезвычайных ситуаций, расчет оптимальных маршрутов движения к месту бедствия, оценка нанесенного ущерба.
- Военное дело. Решение широкого круга специфических задач, связанных с расчетом зон видимости, оптимальных маршрутов движения по пересеченной местности с учетом противодействия и т. п.
- Сельское хозяйство. Прогнозирование урожайности и увеличения производства сельскохозяйственной продукции, оптимизация ее транспортировки и сбыта. [10,22]

1.2 Проблемы информационной безопасности картографических данных

Как известно, картографические данные всегда представляли информационную ценность различного характера: коммерческого, исследовательского, экологического, кадастрового, стратегического, геологического и др. С ростом доступности геоинформационных технологий, обострилась и проблема безопасности пространственных данных.

Геоинформатика – сравнительно молодая наука, которая начала развиваться в 1960-е годы. Со временем было создано множество геоинформационных систем различного уровня в сфере картографии. Геоинформационные технологии захватывали и захватывают все большие сферы влияния, растет спрос на рынке информационных технологий на комплексы программ обработки картографической информации. Основными событиями, повлиявшими на интенсивное развитие ГИС-технологий, стали:

- развитие и доступность систем спутниковой навигации (глобальных систем позиционирования);

- уменьшение стоимости спутниковых снимков высокого качества и доступность их получения, и как следствие – возможность их использования коммерческими организациями.

Полученные снимки могут быть использованы для реализации в собственных информационных системах, на базе ГИС, решающих множество частных задач, например, картографирования, кадастра, управления территориями, мониторинг экологической обстановки и чрезвычайных ситуаций, геомаркетинг и ряд других.

Модели пространственных данных могут иметь векторное или растровое представление. Векторные модели чаще применяются на практике, чем растровые, ибо требуют на порядки меньше памяти и времени на обработку. Кроме того, векторные данные имеют более высокую точность позиционирования и представления. При построении векторных моделей

объекты создаются путем соединения точек прямыми линиями, дугами окружностей, полилиниями. Площадные объекты – ареалы задаются наборами линий.

Организация хранения данных картографии в векторном формате и их обработка сопряжена с определенными трудностями, и при некоторых операциях невозможно обойтись без участия человека, что значительно увеличивает их себестоимость.

Таким образом, формирование векторной модели данных картографии является сложной задачей. Сами же данные картографии, как было отмечено выше, отличаются высокой себестоимостью работ по их получению и содержащейся информации и, соответственно, нуждаются в информационной безопасности. Необходимо ограничивать несанкционированный доступ к ним с целью хищения, нарушения целостности, а так же иметь возможность их восстанавливать в случае утраты по техническим причинам. [19]

1.3 Существующие решения

В связи с широким распространением ГИС, насыщением рынка цифровой картографической продукцией в последнее время достаточно остро встала проблема защиты цифровых карт и картографических баз данных от несанкционированного копирования. Известны случаи, когда цифровая карта, сделанная по заказу для одной организации, моментально распространялась третьим лицам, в "неофициальном" порядке. Из-за сложностей по отслеживанию пиратского копирования и идентификации "подлинности" цифровых карт практически невозможно воспользоваться имеющимися в Российском уголовном и административном законодательстве возможностями по преследованию такого рода операций. Технически данная проблема так же решается непросто и, надо констатировать, специализированного решения на настоящий момент не найдено.

Как и для всего класса информационных систем, сущность обеспечения комплексной защиты ГИС основывается на выполнении трех важнейших свойств защищенной системы:

1) Конфиденциальность. Пользователи системы должны получать доступ только к тем данным, для которых они имеют явное или неявное разрешение на доступ. Как правило, решению проблем обеспечения конфиденциальности специалистами в области безопасности уделяется наиболее пристальное внимание. Комплекс мер обеспечения конфиденциальности включает разработку и описание политик доступа, разработку и реализацию мер по обеспечению разграничения доступа, по обеспечению надежной идентификации и аутентификации пользователей, а также мер криптографической защиты.

2) Целостность — обеспечение защиты от преднамеренного или непреднамеренного изменения информации или процессов ее обработки.

3) Доступность — обеспечение возможности доступа авторизированных в системе пользователей к информации в соответствии с принятой политикой доступа. [9]

Существуют две основные возможности по защите цифровых карт: защита встроенными средствами ГИС и защита при помощи программных и аппаратных решений третьих фирм.

Наиболее удобным для пользователя было бы решение данной проблемы в рамках программных продуктов ГИС. Но, к сожалению, разработчики не торопятся встраивать системы защиты цифровых карт в свои программные продукты, т.к. это не соответствует их коммерческим интересам (цифровые карты - бесплатно, программные продукты - за деньги). Имеющиеся в некоторых программных продуктах возможности создают только иллюзию защиты и не являются проблемой для знающих людей. Так, например, в ArcView 3.0, все пароли, предназначенные для блокировки, вы можно легко обнаружить в файле *.apr, просто поискав теги с именами "password". Использование специализированных продуктов третьих фирм является, пожалуй, единственной серьезной возможностью, позволяющей снять остроту проблемы.

Наиболее распространенный тип защиты на сегодняшний день - защита от несанкционированного копирования, как посторонними, так и доверенными лицами. Это, пожалуй, самый сложный для исполнения тип защиты, т.к. пользователи, работающие с защищаемыми файлами данных, имеют доступ, как на чтение, так и на запись. В то же время необходимо обеспечить, чтобы информация, содержащаяся в файлах, не выходила, в своем исходном виде, за пределы некоторого ограниченного "пространства" (локального компьютера, организации) или была растиражирована в строго определенном количестве экземпляров. При этом заранее считается, что пользователи не заинтересованы в обеспечении защиты организационными методами. Наиболее надежным способом защиты данного типа, разработанным к настоящему времени является использование специальных электронных ключей. Одним из самых распространенных типов продукции такого вида являются электронные ключи Hardlock, выпускаемые израильской фирмой Aladdin.

Система защиты с использованием электронных ключей Hardlock функционирует следующим образом: программа, считывающая и интерпретирующая данные в каком-либо файле, обращается к диску компьютера (или какому-либо логическому устройству) для считывания данных посредством вызова специальных функций операционной системы (подсистема ввода-вывода). Она получает данные из участка файла, интерпретирует их и затем записывает (при необходимости) обратно на диск. Решение проблемы состоит в том, чтобы перехватить операции "чтения-записи" либо на уровне операционной системы, либо на аппаратном уровне и провести дешифрование-шифрование считываемых или записываемых данных.

Программное средство дешифрования-шифрования данных перехватывает операции файлового ввода-вывода, осуществляемого программным средством ГИС; проверяет - необходимо ли осуществлять дешифрирование или шифрование считываемых данных; осуществляет идентификацию пользователя путем запрашивания пароля или проверки наличия специального ключа защиты; проверяет некоторые другие параметры,

такие как счетчик запусков программы или время, прошедшее с начала работы (для ограничения по времени); на основании этих проверок принимает решение о возможности осуществления операций чтения-записи; осуществляет процессы кодирования/декодирования или разрешает/запрещает доступ к данным, в зависимости от режима функционирования.

Программные средства защиты Hardlock, как уже было замечено выше, определяют необходимость шифрования или дешифрования того или иного файла путем анализа его имени или расширения. Однако зачастую ГИС имеют файлы нескольких расширений. Таким образом, основная сложность при защите цифровых карт состоит в отключении каким-либо способом операций копирования или конвертирования, которые имеются в каждой системе.

Однозначным решением этой проблемы является защита всех файлов которые могут быть открыты или созданы ГИС-программой. Программное обеспечение Hardlock позволяет сделать это, но последствия очевидны: ГИС превращается в замкнутую систему, которая "все впускает, но ничего не выпускает". Такой подход приемлем только в том случае, если не предполагается создавать в ГИС-программе "открытые" копии исходных или обработанных данных, включая даже текстовые файлы, а весь информационный обмен осуществляется в рамках однотипных программных средств или с использованием одного и того же защищенного формата, защищенных с использованием одного и того же кода и ключевого слова. [14]

1.4 Архитектура сервера ЗКБД

Архитектура системы управления ЗКБД показана на Рис.1. Организация модулей в данной архитектуре является классикой [11], поэтому не нуждается в исчерпывающем описании. Рассмотрим отдельные модули.

Клиентская сторона.

Конвертер – это модуль, который преобразует результирующие отношения, получаемые от сервера СУБД, в формат ГИС-данных и проводит обратное преобразование для передачи ГИС-данных на сервер (и то, и другое выполняются опосредованно через клиентское хранилище).

Клиентский модуль выполняет два типа запросов к серверу ЗКБД: pull и push. Pull-запрос выполняет пространственную выборку данных, push-запрос выполняет изменение ЗКБД.

Серверная сторона.

Арбитр запросов – это модуль, который блокирует выполнение любых поступающих от клиента запросов в ходе выполнения pull-запроса.

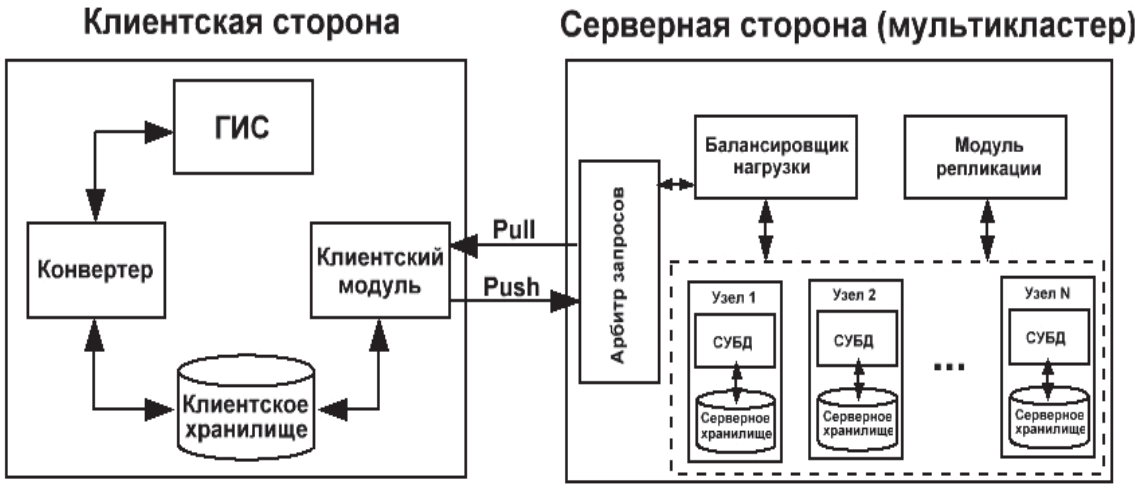


Рис.1. Архитектура системы управления ЗКБД

При успешном обновлении ЗКБД выполняется процедура репликации баз данных на узлах, в противном случае выполняется откат базы данных до предыдущей версии.

Серверная сторона организована по принципу *мультикластера*. В случае такой организации каждый запрос, поступающий от клиентов, обрабатывается одним узлом сервера ЗКБД. За равномерное распределение запросов между узлами отвечает *балансировщик нагрузки*.

ГЛАВА 2. ОРГАНИЗАЦИЯ СИСТЕМЫ И ОПИСАНИЕ РЕАЛИЗАЦИИ ПРОЕКТА

Производится выбор необходимых параметров, средств и методов реализации проекта. Рассматривается целесообразность и обоснование выбранных вариантов.

2.1 Выбор ГИС

ГИС в целом и программное обеспечение (далее ПО), как важнейший компонент ГИС, прошли путь от мейнфреймов до персональных компьютеров. На сегодняшний день наиболее распространенными программными продуктами для разработки и управления ГИС являются уже так называемые пользовательские ГИС, то есть ГИС управляемые одним пользователем и располагающиеся на одном персональном компьютере (хотя данные могу быть распределены внутри локальной сети или интернет). Базовая функциональность ПО ГИС основных производителей во многом одинакова, однако расширенная может сильно различаться. Было проведено сравнение наиболее распространенных пользовательских ГИС.

Mapinfo (Mapinfo Corp) - полнофункциональная инструментальная геоинформационная система (ГИС). С ее помощью можно создавать и редактировать карты, хранить и обрабатывать информацию, связанную с картографическими объектами. С точки зрения общепринятой терминологии ГИС MapInfo является системой управления базой пространственных данных. В дополнение к традиционным для СУБД функциям MapInfo позволяет обрабатывать также картографические данные, хранящиеся в базе, с учетом пространственных отношений объектов. ГИС MapInfo может выступать в роли «картографического клиента» при работе с такими известными СУБД, как Informix, Sybase, Oracle8 и DB2, поскольку поддерживает эффективный механизм взаимодействия с ними как через протокол ODBC, так и «напрямую».

MapInfo Professional обеспечивает:

- картографический интерфейс;

- создание и редактирование карт высокого качества;
- пространственные данные, поставляемые с программным обеспечением;
- простоту интеграции карт — несколько строк кода внедряют окно MapInfo в приложения Windows (Excel, Access, Word);
- анимационный слой — быстрая перерисовка при частых изменениях на слое (полезно для систем слежения за движущимися объектами);
- работу с 3D-данными — построение изолиний, триангуляция Делоне и трехмерная визуализация.

Мощным средством анализа и наглядного представления пространственных данных в ГИС MapInfo является тематическая картография. На тематической карте легко понять связи между различными объектами и проследить тенденции в развитии различных явлений. Картограммы, столбчатые и круговые диаграммы, метод значков, плотность точек, метод качественного фона, непрерывной поверхности-грида — вот только некоторые виды тематических карт в MapInfo. Сочетание тематических слоев и методов буферизации, районирования, слияния и разбиения объектов, пространственной и атрибутивной классификации позволяет создавать синтетические многокомпонентные карты с иерархической структурой легенды. Mapinfo одна из наиболее развитых, мощных и простых в использовании система настольной картографии, позволяющая решать широкий спектр задач в различных сферах деятельности. ГИС MapInfo широко распространена в нашей стране, однако значительно меньше за рубежом. [2]

ArcGIS (ESRI) - одна из наиболее распространенных ГИС на сегодняшний день. ArcGIS представляет собой полную систему, которая позволяет собирать, организовывать, управлять, анализировать, обмениваться и распределять географическую информацию. Являясь мировым лидером среди платформ для построения и использования геоинформационных систем, ArcGIS используется людьми по всему миру для применения географических знаний в практической сфере государственного управления, бизнеса, науки, образования

и СМИ. Платформа ГИС позволяет публиковать географическую информацию для доступа и использования любыми пользователями. Система доступна в любой точке, где возможно использование веб-браузеров, мобильных устройств в виде смартфонов, а также настольных компьютеров. ArcGIS по своей сути является набором программных продуктов и инструментов, который используется для профессионального выполнения ГИС-задач. Система ArcGIS позволяет создавать надежную географическую информацию ГИС-сообществом, легко и просто использовать ее любым заинтересованным лицам (а также определять, с кем ею можно обмениваться). Данная система включает в себя программное обеспечение, интерактивную облачную инфраструктуру, профессиональные инструменты, настраиваемые ресурсы, например, шаблоны приложений, готовые к использованию веб и мобильные приложения, готовые к использованию базовые карты, а также надежное содержание, которое распространяется сообществом пользователей. Поддержка серверов и облачных платформ позволяет выполнять совместную обработку и обмен; при этом критическая для планирования и принятия решений информация немедленно становится доступной для всех. ArcGIS используется многими, благодаря непревзойденной скорости работы, огромной базе модулей расширения и дополнительного ПО. Имеет собственный формат данных, используемый и в других продуктах этой компании. [1]

QGIS (QGIS Development Team) — свободная кроссплатформенная геоинформационная система. Работа над QGIS была начата в мае 2002 года, а в июне того же года — создан проект на площадке SourceForge. Создатели много работали, чтобы сделать программное обеспечение ГИС (которое традиционно является дорогим проприетарным ПО) доступным любому, кто имеет доступ к персональному компьютеру. В настоящее время QGIS работает на большинстве платформ. QGIS разработана с использованием инструментария Qt и языка программирования C++. Это означает, что QGIS легка в использовании, имеет приятный и простой графический интерфейс. QGIS стремится быть легкой в использовании ГИС, предоставляя общую функциональность. Первоначальная

цель заключалась в облегчении просмотра геоданных и QGIS достигла той стадии в своем развитии, когда многие используют ее в своих ежедневных задачах просмотра. QGIS поддерживает множество растровых и векторных форматов данных, а поддержка новых форматов реализуется с помощью модулей.

QGIS выпускается на условиях лицензии GNU General Public License (GPL). Разработка QGIS под этой лицензией означает, что любой может просмотреть и изменить исходный код, и гарантирует, что пользователь, всегда будете иметь доступ к программному обеспечению ГИС, которое является бесплатным и может свободно адаптироваться. QGIS позволяет использовать большое количество распространенных ГИС функций, обеспечиваемых встроенными инструментами и модулями. Можно создавать, редактировать, управлять, просматривать и накладывать друг на друга векторные и растровые данные в различных форматах и проекциях без преобразования во внутренний или общий формат.

Поддерживаются следующие основные форматы:

• пространственные таблицы и представления PostGIS, SpatiaLite, MSSQL Spatial и Oracle Spatial;

• векторные форматы, поддерживаемые установленной библиотекой OGR, включая shape-файлы ESRI, MapInfo, SDTS , GML и многие другие;

• форматы растров и графики, поддерживаемые библиотекой GDAL (Geospatial Data Abstraction Library);

• пространственные данные, публикуемые в сети Интернет с помощью OGC-совместимых сервисов.

QGIS позволяет выполнять пространственный анализ с использованием баз данных и OGR-совместимых форматов, осуществлять анализ векторных данных, геообработку, предоставляет широкий набор инструментов управления данными. Возможно использовать инструменты GRASS, которые предоставляют доступ к более чем 400 модулям GRASS. Или же использовать

модуль Processing, который позволяет вызывать множество встроенных и сторонних алгоритмов, например GDAL, SAGA, GRASS и др.

QGIS может быть адаптирована к особым потребностям благодаря расширяемой модульной архитектуре. QGIS предоставляет библиотеки, которые могут использоваться для создания модулей. Можно создавать отдельные приложения, используя языки программирования C++ или Python. [6]

По основным причинам открытости программного кода, а соответственно доступности любому пользователю, достаточно мощному функционалу и совместимости с объектно-реляционной СУБД PostgreSQL, для работы была выбрана ГИС QGIS.

2.2 Тестовая БД. Пространственные запросы

Пространственные данные (географические данные, геоданные) — данные о пространственных объектах и их наборах. Иными словами - всё, что имеет координаты. Погода, рельеф, леса, поля, встречи животных, дороги, дома и страны, города и сёла, все это и многое другое - объекты и явления, которые имеют координаты. Объекты могут быть разбросаны случайно или закономерно, регулярно или нерегулярно. Пространственные данные составляют основу информационного обеспечения геоинформационных систем.

Существует два типа пространственных данных geometry и geography. Тип данных geometry поддерживает планарные или эвклидовы данные (система координат для плоской Земли). Geometry соответствует спецификации «Simple Features for SQL» консорциума OGC версии 1.1.0 и стандарту SQL MM (стандарт ISO). Тип данных geography используется для хранения эллиптических данных, таких как координаты GPS широты и долготы. [16]

В нашем проекте был использован тип geometry.

Для работы была выбрана тестовая база данных, представленная четырьмя субъектами Российской Федерации: Кемеровской, Новосибирской областями, Алтайским краем и Республикой Алтай. [15]

Данные в открытом наборе геоданных представлены послойно, в векторном формате. Данные находятся в географической системе координат WGS84, проекция отсутствует. Названия файлов данных и слоёв в проектах англоязычные: населенные пункты - settlements, речная сеть - hydro и т.д.. Кодировка атрибутивных данных Win1251. Для других форматов - UTF-8.

База данных состоит из 12 слоев (таблиц):

1. Административные границы (admin)
2. Границы экопегионов (ecoregions)
3. Гидросеть полигональная (hydro-a)
4. Гидросеть линейная (hydro-l)
5. Охраняемые природные территории (Заповедники, национальные парки) (oopt)
6. Точечные объекты (рестораны автозаправки и т.п.) (poi-osm) Тип слоя: векторный, точечный.
7. Железные дороги (railroad-l) Тип слоя: векторный, линейный.
8. Дорожная сеть (road-l-osm) Тип слоя: векторный, линейный.
9. Населенные пункты (settlements)
10. Почвы (soils)
11. Рельеф (spatial_ref_sys)
12. Растительность (veg)

Смысл функциональности пространственных баз данных заключается в выполнении запросов к базе данных, для которых иначе потребовалась бы настольная ГИС.

Пространственный запрос – это структурированный запрос к пространственным данным, критериями которого являются условия, связанные с координатами векторной геометрии. К примеру, можно запросить все объекты определенного типа, которые находятся внутри заданного контура, пересекают заданную границу или находятся на определенном расстоянии от заданной точки.

С пространственными данными может быть связана также любая неграфическая информация, дополнительно характеризующая тот или иной объект в системе. Более того, любой объект информационной модели в ГИС может быть представлен совокупностью пространственных объектов и наборов, связанных с ними семантических атрибутов, описывающих этот объект точно так же, как если бы он был представлен в любой неграфической системе. Допустим, если ГИС использует для хранения своих данных СУБД, то семантическая часть описания объектов – это записи в таблицах реляционной базы данных. Довольно часто структурированные запросы к данным под управлением ГИС представляют собой симбиоз традиционных SQL запросов к базе данных и параметров пространственных запросов. Отчасти SQL основан на формальном языке запросов, реляционной алгебре, его отличают простота использования, интуитивность и универсальный характер. Системы управления пространственными базами данных представляют собой частный случай расширяемых СУБД и работают как с пространственными, так и с непространственными данными, поэтому вполне естественно попытаться найти расширение SQL, позволяющее обращаться к пространственным данным.

Вообще говоря, запросы можно разделить на две категории: те, что требуют однопроходного просмотра (single-scan queries), и те, что требуют многопроходного просмотра (multiscan queries). При обработке запроса первой категории обращение к запрашиваемой записи (кортежу) таблицы (отношения) должно производиться не более одного раза. Следовательно, при реализации наихудшего - с точки зрения времени - сценария каждая запись будет извлекаться из таблицы и подвергаться проверке на предмет соответствия критерию поиска.

Примером запроса, требующего многопроходного просмотра, является запрос соединения (join query). Чтобы выдать ответ на запрос соединения, СУБД должна произвести выборку и комбинирование двух таблиц базы данных. Если в запросе требуется обработать более двух таблиц, они могут обрабатываться попарно. Две таблицы «соединяются» по общему атрибуту.

Поскольку записи в одной таблице могут быть связаны более чем с одной записью в другой таблице, есть вероятность того, что для завершения соединения одного обращения к записи будет недостаточно. В контексте пространственных БД, когда атрибуты соединения являются пространственными по своей природе, такой запрос носит название запроса пространственного соединения (spatial-join query).

Конструкции объектно-ориентированного программирования, такие, как пользовательские типы, а также наследование данных и функций, нашли непосредственное применение в создании моделей сложных данных. Повсеместное использование реляционной модели и языка SQL в приложениях, включающих простые типы данных, в сочетании с функциональностью объектно-ориентированной модели привело к рождению новой «гибридной» парадигмы систем управления базами данных, именуемой объектно-реляционная СУБД. [23]

2.3 Выбор СУБД

Базы данных используются уже долгое время. Особенно популярны они стали благодаря системам управления, реализующим реляционную модель данных. Была поставлена задача о выделении основных различий между разными системами управления базами данных с открытым исходным кодом, рассмотреть их функциональные отличия и особенности, как они работают и какую СУБД лучше выбрать исходя из потребностей разработки.

СУБД должна обеспечивать реляционную модель работы с данными. Сама модель подразумевает определенный тип связи между сущностями из разных таблиц. Чтобы хранить и работать с данными, такой тип СУБД должен иметь определенную структуру (таблицы). В таблицах каждый столбец может содержать данные разного типа. Каждая запись состоит из множества атрибутов (столбцов) и имеет уникальный ключ, хранящийся в той же таблице - все эти данные взаимосвязаны между собой, как описано в реляционной модели.

Отношения в базах данных можно рассматривать как математическое множество, содержащее в себе число атрибутов, которые суммарно представляют собой базу данных и информацию, хранящуюся в ней.

Во время создания структуры таблицы каждое поле записи должно иметь заранее описанный тип (например: строка, целочисленное значение и т.д.). Все СУБД имеют в своем составе различные типы данных, которые не всегда являются взаимозаменяемыми. При работе с СУБД всегда приходится сталкиваться с подобными ограничениями.

Было рассмотрено три основных свободно распространяемых СУБД: SQLite, MySQL и PostgreSQL.

SQLite (Richard Hipp). Легко встраиваемая в приложения база данных. Так как это система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам (в эти файлах хранятся данные), вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек.

К преимуществам SQLite можно отнести, пожалуй, лишь файловую структуру - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины. Основным недостатком данной СУБД является отсутствие системы пользователей, что крайне важно для нашего проекта.[7]

MySQL (Oracle) - это самая распространенная полноценная серверная СУБД. MySQL очень функциональная, свободно распространяемая СУБД, которая успешно работает с различными сайтами и веб приложениями. Несмотря на то, что в ней не реализован весь SQL функционал, MySQL предлагает довольно много инструментов для разработки приложений. Так как это серверная СУБД, приложения для доступа к данным, в отличии от SQLite работают со службами MySQL.

К преимуществам MySQL относят: простоту в работе, богатый функционал, большое количество функций обеспечивающих безопасность, возможность работать с большими объемами данных, скорость работы.

Однако, так как MySQL не ставит для себя целью - полностью соответствовать стандартам SQL, то эта СУБД не полностью поддерживает SQL. Так же к недостаткам относят и многопоточность, хотя некоторые движки БД довольно легко выполняют параллельное чтение, параллельные операции чтения-записи могут создать проблемы.[3]

PostgreSQL (The PostgreSQL Global Development Group) является самым профессиональным из всех трех рассмотренных нами СУБД. Она свободно распространяемая и максимально соответствует стандартам SQL. PostgreSQL или Postgres стараются полностью применять ANSI/ISO SQL стандарты своевременно с выходом новых версий.

От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций (ACID). Благодаря мощным технологиям PostgreSQL очень производительна. Параллельность достигнута не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом (MVCC), что также обеспечивает соответствие ACID. PostgreSQL очень легко расширять своими процедурами, которые называются хранимые процедуры. Эти функции упрощают использование постоянно повторяемых операций.

Хотя PostgreSQL и не может похвастаться большой популярностью в отличии от MySQL, существует довольно большое число приложений облегчающих работу с PostgreSQL, несмотря на всю мощность функционала.

Важным критерием для нас явился объем поддерживаемых типов данных, которым не могли похвастаться две предыдущие СУБД вместе взятые:

bigint - знаковое 8-ми битное целочисленное значение

bigserial - автоматически инкрементируемое 8-ми битное целочисленное значение

bit[(n)] - строка постоянной длины

bit varying [(n)] - строка переменной длины

boolean - булево значение (true/false)

box - прямоугольник на плоскости

bytea - бинарные данные (массив байтов)

character varying [(n)] - строковое значение переменной длины

character [(n)] - строковое значение постоянной длины

cidr - IPv4/IPv6 сетевой адрес

circle - круг на плоскости

date - календарная дата (год, месяц, день)

double precision - число с плавающей точкой двойной точности (8 байт)

inet - IPv4/IPv6 адрес хоста

integer - знаковое 4-ех байтовое целочисленное значение

interval [fields][(p)] - отрезок времени

line - бесконечная прямая на плоскости

lseg - отрезок на плоскости

macaddr - MAC адрес

mony - валютное значение

numeric [(p, s)] - точное численное значение с выбранной точностью

path - геометрическая кривая на плоскости

point - геометрическая точка на плоскости

polygony - многоугольник на плоскости

real - число с плавающей точкой одинарной точности (4 байта)

smallint - знаковое целочисленное значение (4 байта)

serial - автоматические инкрементируемое целочисленное значение (4 байта)

text - строковое значение переменной длины

time [(p)] [without time zone] - время суток (без часового пояса)

time [(p)] with time zone - время суток (включая часовой пояс)

timestamp [(p)] [without time zone] - дата и время (без часового пояса)

timestamp [(p)] with time zone - дата и время (с часовым поясом)

tsquery - текстовый поисковый запрос

tsvector - документ текстового поиска

txid_snapshot - пользовательский снимок транзакции с ID

uuid - универсальный уникальный идентификатор

xml - XML данные

Основными достоинствами PostgreSQL так же являются:

• соответствие стандарту SQL;

• большое сообщество - существует довольно большое сообщество в котором вы запросто найдёте ответы на свои вопросы;

• большое количество дополнений - несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими;

• расширения - существует возможность расширения функционала за счет сохранения своих процедур;

• объектность - PostrgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого.

Так же из преимуществ над другими СУБД хотелось бы отметить:

• надежность и целостность хранимых данных;

• наличие встроенной поддержки для использования сложных пользовательских процедур;

• легкость интеграции (переход на платные СУБД, например Oracle, сделать с PostgreSQL будет довольно просто по сравнению с другими бесплатными СУБД).

Сложная структура данных - по сравнению с другими открытыми СУБД PostgreSQL предоставляет больше возможностей для создания сложных структур данных без необходимости жертвовать какими-либо аспектами.[5,8]

Изучив все плюсы и минусы, было принято решение в данном проекте использовать объектно-реляционную СУБД PostgreSQL.

ГЛАВА 3. РЕАЛИЗАЦИЯ ДИПЛОМНОГО ПРОЕКТА

Рассматриваются некоторые технические аспекты реализации проекта.

3.1 Установка и настройка объектно-реляционной СУБД PostgreSQL

Как говорилось ранее, PostgreSQL является свободной объектно-реляционной системой управления базами данных. Последнюю версию PostgreSQL можно скачать на официальном сайте (http://www.postgresql.org). Начиная с версии 9.0 для Windows предоставляются собранные версии как 32-bit так и 64-bit. В данном проекте рассматривалась установка 64-битной версии PostgreSQL 9.5 на 64-битную версию Windows 8.1. Установка производилось от пользователя, имеющего административные права.

Первым делом устанавливается Microsoft Visual C++ Redistributable – рабочие модули библиотек Visual C++, необходимые для запуска созданных при помощи этого языка программирования приложений (см. Рис.2).

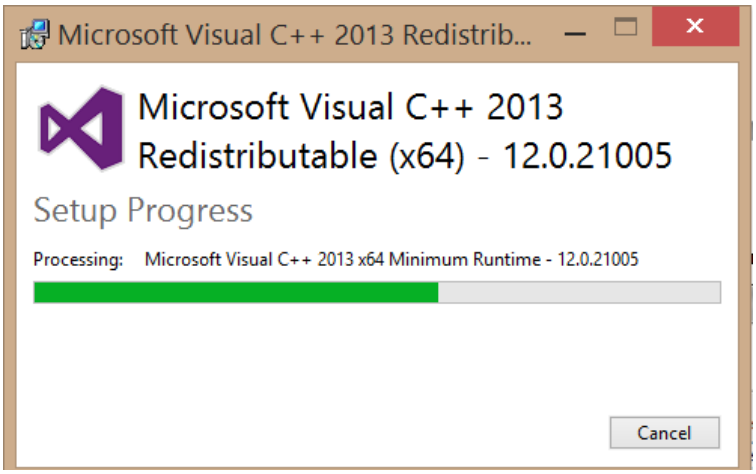


Рис.2. Установка Microsoft Visual C++ Redistributable

Затем выбирается каталог установки программы (см. Рис.3).

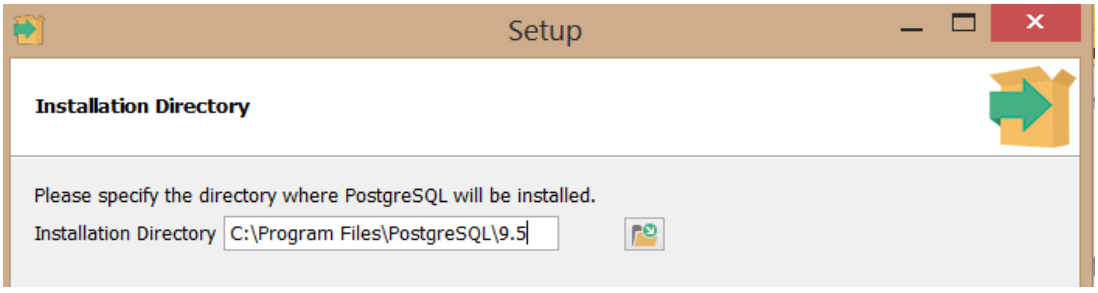


Рис.3. Выбор каталога установки

Следующим шагом выбирается каталог хранения данных (см. Рис.4). Это довольно удобно и разработчики логично предположили, что многие могут захотеть хранить данные на других дисках, скажем более быстрых, для увеличения производительности баз данных. Раньше, конечно, это тоже можно было настроить через файл конфигурации, но теперь это можно сделать уже на этапе установки. Также можно выбрать существующий каталог с данными, которые будут использоваться повторно, если они совместимы с сервером.

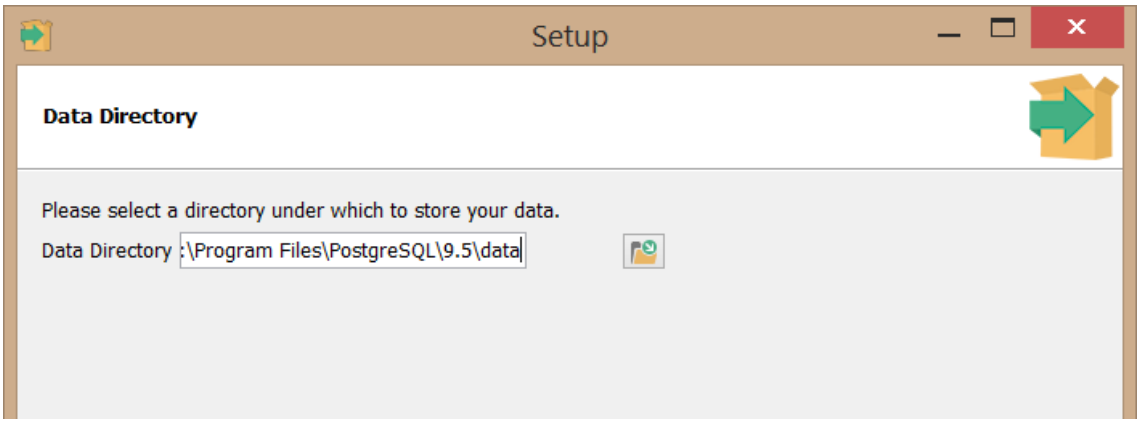


Рис.4. Выбор каталога хранения данных

Вводится и подтверждается пароль, который будет использоваться для учётной записи суперпользователя (postgres) в базе данных кластера (см. Рис.5). PostgreSQL в Windows работает не от администратора, а от имени учётной записи пользователя, имя которого "postgres". Сделано это было, прежде всего, из соображений безопасности, чтобы никакие вредители системы не смогли получить права администратора, даже если они каким-то образом найдут уязвимость в безопасности самого PostgreSQL. В самой СУБД PostgreSQL есть такой специальный пользователь - суперпользователь, который имеет максимальные права внутри СУБД PostgreSQL, только он может создавать или удалять любые базы данных и любых пользователей. Этот суперпользователь тоже имеет имя пользователя "postgres". Но, несмотря на то, что имена пользователей учётной записи в Windows и суперпользователя PostgreSQL абсолютно одинаковы, это абсолютно разные пользователи, абсолютно никаким образом не связанные друг с другом. Предлагается задать один и тот же пароль для них обоих, чтобы потом не путаться с разными паролями. В

Windows, этот пароль также используется для учётной записи службы "postgres" и должен соответствовать текущему паролю в том случае, если учётная запись суперпользователя уже существует. Если учётная запись службы уже существует в Windows, нужно ввести текущий пароль этой учётной записи. Если данная учётная запись не существует, она будет создана. Можно изменить пароль суперпользователя независимо от пароля учётной записи службы "postgres" в любое время после установки.

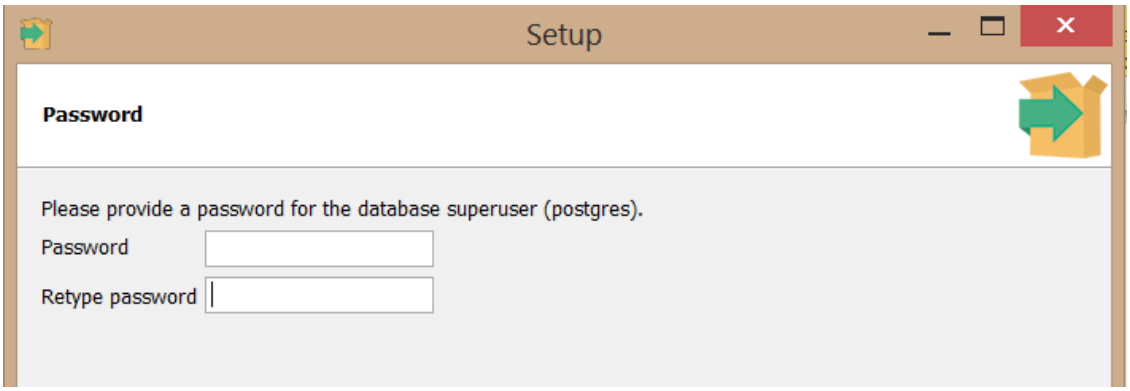


Рис.5. Задание пароля суперпользователя

Выбирается порт подключения (см. Рис.6). По умолчанию, входящие соединения будут приниматься только с локального компьютера. Это можно изменить, отредактировав файл "pg_hba.conf" в каталоге с данными и перезагрузив сервер. Значение по умолчанию: 5432.

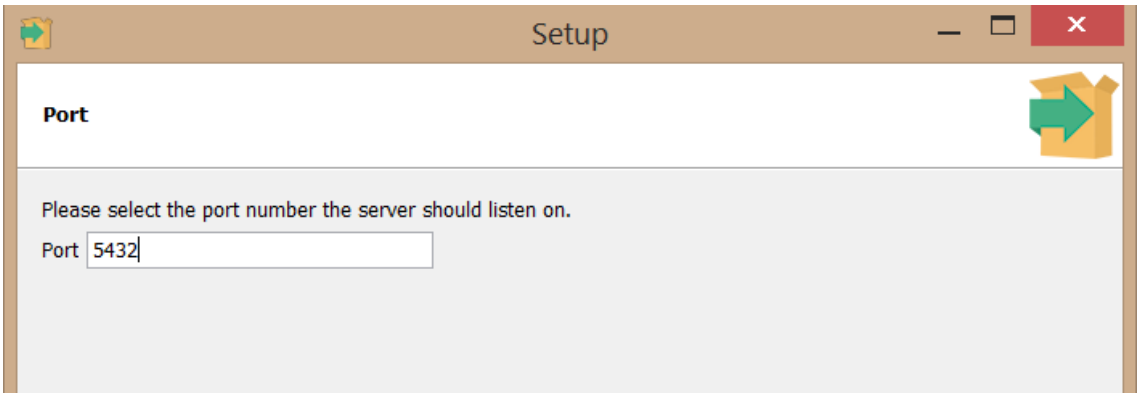


Рис.6. Выбор порта подключения

Выбирается локаль, с использованием которой будет инициализирован кластер баз данных и которая в дальнейшем будет использоваться по умолчанию при создании других баз данных (см. Рис.7). Это довольно важный шаг, ибо локаль определяет такой важный параметр как кодировка данных в

базах. Большинство людей всё-таки предпочитает работать с кодировкой UTF-8. Эта кодировка будет установлена в том случае, если в данном окне вы выберите локаль по умолчанию: [Default locale].

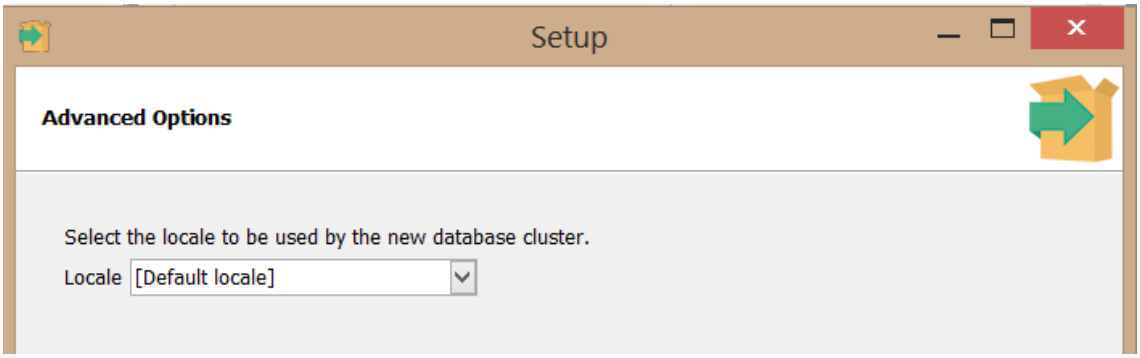


Рис.7. Выбор локали

После завершения установки, предлагается возможность запуска Stack Builder (см. Рис.8,9), с помощью которого можно загрузить и установить дополнительные компоненты, популярные приложения и драйверы к дополнению установки PostgreSQL.

Для хранения в базе данных географической информации требуется установить расширение PostGIS. PostGIS включает поддержку пространственных индексов R-Tree/GiST, а также функции обработки геоданных. PostGIS так же, как и PostgreSQL является свободным ПО с открытым исходным кодом. Так же требуется установить драйвер программного интерфейса доступа к базам данных (ODBC) и поставщик данных Npgsql, позволяющий клиентским приложениям отправлять и получать данные с сервера PostgreSQL.

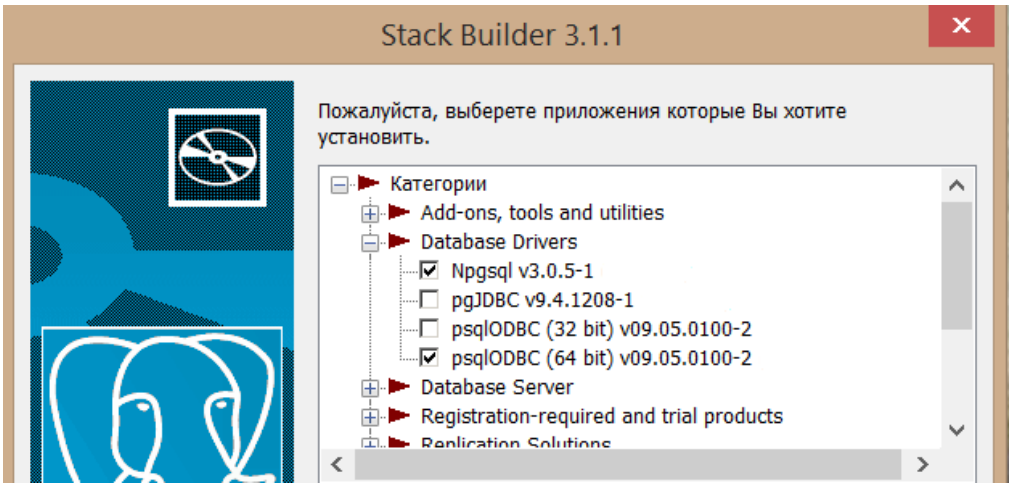


Рис.8. Окно Stack Builder

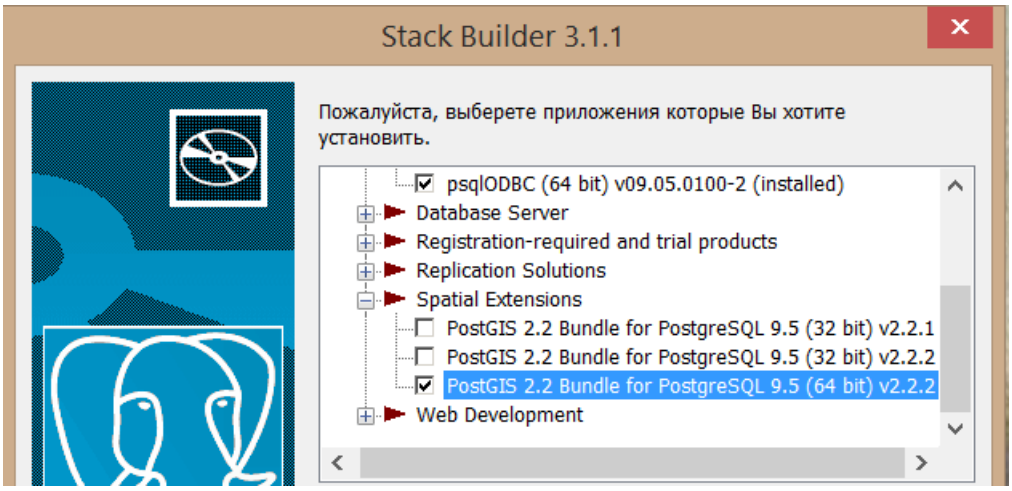


Рис.9. Окно Stack Builder

Следующим шагом выбирается каталог загрузки (см. Рис.10).

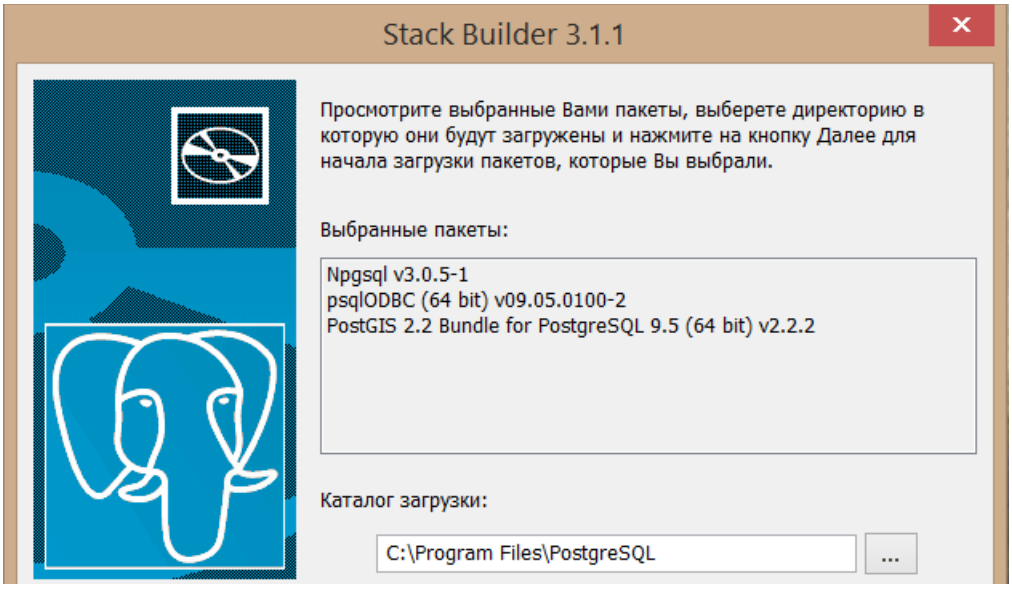


Рис.10. Выбор каталога загрузки расширений

Загружаются файлы (см. Рис.11). Общий объем загружаемых файлов составляет около 47 МБ.

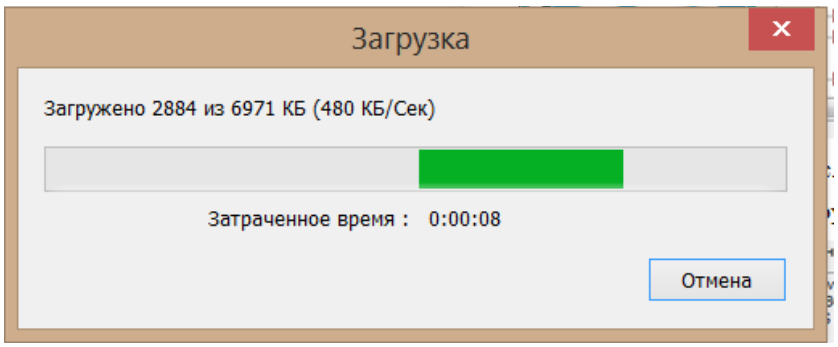


Рис.11. Загрузка файлов

Далее устанавливаются Npgsql и psqlODBC.

3.1.1 Установка PostGIS

В открывшемся окне установки PostGIS ставится отметка на пункте «Create spatial database» (см. Рис.12). «Create spatial database» позволяет создать пространственную базу данных автоматически. В нашем проекте требуется создание такой базы, это делается, прибегая к услугам установщика PostGIS, и поэтому на данном этапе отметка на этом пункте ставится.

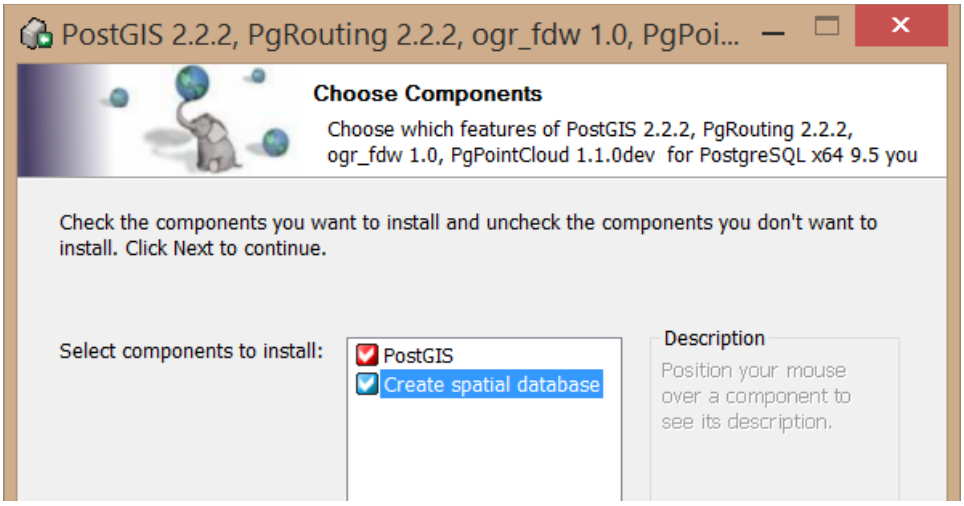


Рис.12. Окно выбора компонентов PostGIS

Выбирается путь для установки (см. Рис.13). На диске расширение занимает 139.9 МБ памяти.

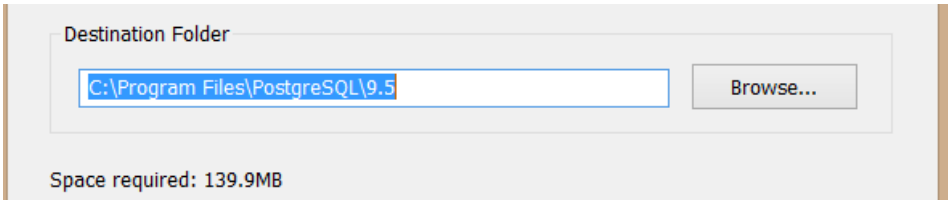


Рис.13. Выбор каталога установки PostGIS

Заполняются представленные поля (см. Рис.14). В поле «Password» вводится пароль указанный ранее (см. стр.35).

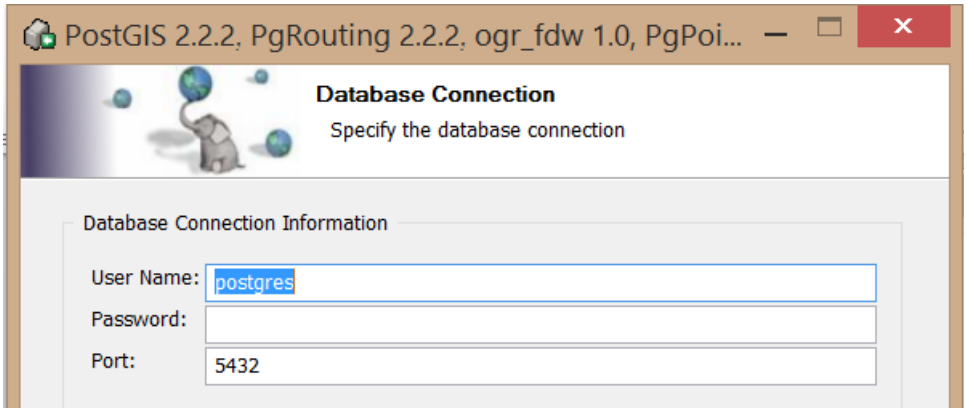


Рис.14. Ввод параметров подключения

Вводится название для создаваемой базы данных (см. Рис.15). Стоит отметить, что эта база данных в последующем может использоваться как шаблон для всех остальных баз.

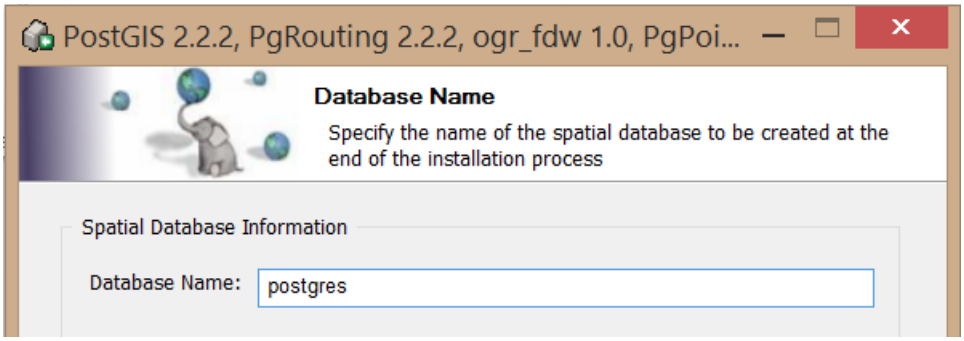


Рис.15. Ввод названия создаваемой базы данных

После настройки параметров, происходит непосредственная установка PostGIS (см. Рис.16).

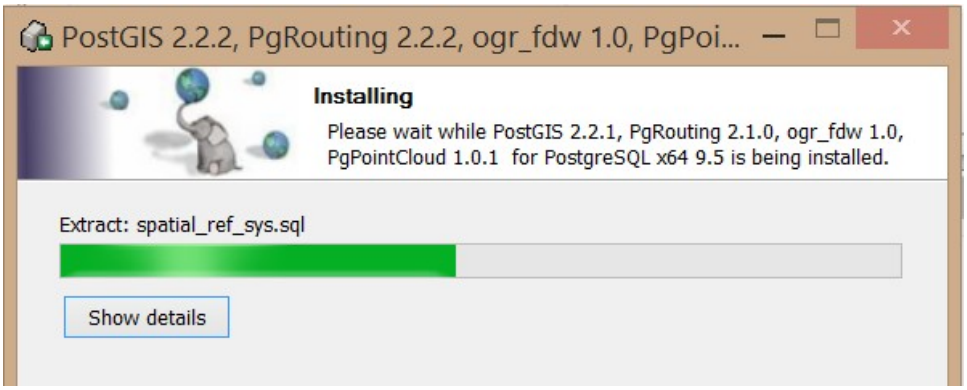


Рис.16. Процесс установки PostGIS

После сообщения об успешной установке, мастер установки Stack Builder закрывается. [20]

3.2 Реализация модуля расшифровки картографической базы данных

PostgreSQL предоставляет возможность шифрования на разных уровнях, что позволяет защитить данные от кражи с сервера, недобросовестного администратора и небезопасной сети. Шифрование так же может использоваться для защиты конфиденциальных данных, таких как медицинская информация или финансовые транзакции. В работе были использованы лишь некоторые из имеющихся механизмов шифрования объектно-реляционной СУБД PostgreSQL.

Основу проекта составляет шифрование определённых колонок. Модуль pgcrypto позволяет шифровать определённые поля. Это полезно в случае, когда лишь некоторые данные не должны попасть в чужие руки. В нашем случае зашифрованными являются лишь столбцы типа geometry, которые представляют наибольшую ценность, так как содержат все географические данные. Клиенту предоставляется ключ для расшифровки, которым данные дешифруются и становятся «понятными» для ГИС. [4]

Таким образом, на сервере хранится таблица с преобразованным в текст и зашифрованным полем geom, которое хранит в себе все геометрические данные (см. Рис.17).

id_0 integer	name character varying(50)	id integer	geotext text
1	Кемеровская област	0	\xc30d04070302deb8a8398ef9c3b27dd2ee01d21a8917
2	Новосибирская обла	1	\xc30d04070302fef5c3dfd78ac06376d2ee01f5312607
3	Алтайский край	2	\xc30d04070302821d24cb66bd951166d2ee019d7f187b
4	Республика Алтай	3	\xc30d040703026b57b8ee6f9606c46cd2ee01f34efb79

Рис.17. Фрагмент исходной таблицы

Первым делом предстоит расшифровать интересующий нас столбец. Для этого используется функция модуля pgcrypto - pgp_sym_decrypt. Для успешной расшифровки, необходимо указать тот же ключ, что использовался при зашифровке данных, поэтому для обеспечения надежной защиты данных, необходимо предотвращение попадания данного ключа посторонним лицам. В случае верного ввода ключа, произойдет расшифровка интересующего нас поля. Однако оно по-прежнему будет типа text (см. Рис.18).

id_0 integer	name character varying(50)	id integer	geotext text
1	Кемеровская област	0	SRID=4326;MULTIPOLYGON(((88.6241455078125 56.8
2	Новосибирская обла	1	SRID=4326;MULTIPOLYGON(((84.4106750488281 56.0
3	Алтайский край	2	SRID=4326;MULTIPOLYGON(((85.0794677734375 54.4
4	Республика Алтай	3	SRID=4326;MULTIPOLYGON(((88.3648834228516 52.1

Рис.18. Фрагмент расшифрованной таблицы

Следующим шагом требуется преобразовать получившееся уже незашифрованное текстовое поле в геометрический тип данных. Это можно сделать используя геометрический конструктор PostgreSQL, а конкретно функцию ST_GeomfromEWKT.

Стоит отметить, что не следует все преобразования производить в одном столбце, так как если производить преобразование text→geometry в столбце типа данных text, то преобразование не произойдет, поскольку для геометрического типа данных поле должно иметь тип geometry. Поэтому, используя стандартный SQL оператор ALTER TABLE ADD COLUMN, создается новое пустое поле нужного типа, в которое и записываются данные полученные при преобразовании (см. Рис.19).

id_0 integer	name character varying(50)	id integer	geotext text	geom geometry
1	Кемеровская област	0	SRID=4:	0106000020E6100000010000000103000000010000007D06(
2	Новосибирская обла	1	SRID=4:	0106000020E6100000010000000103000000010000007F107(
3	Алтайский край	2	SRID=4:	0106000020E6100000010000000103000000010000003D07(
4	Республика Алтай	3	SRID=4:	0106000020E6100000010000000103000000010000007906(

Рис.19. Фрагмент преобразованной таблицы

Текстовое поле, после успешного проделывания всех операций можно удалить. Для этого используется стандартный SQL оператор ALTER TABLE DROP (см. Рис.20).

id_0 integer	name character varying(50)	id integer	geom geometry
1	Кемеровская област	0	0106000020E6100000010000000103000000010000007D06(
2	Новосибирская обла	1	0106000020E6100000010000000103000000010000007F107(
3	Алтайский край	2	0106000020E6100000010000000103000000010000003D07(
4	Республика Алтай	3	0106000020E6100000010000000103000000010000007906(

Рис.20. Фрагмент конечной таблицы

Блок-схема программы приведена в приложении 1. Листинг программы представлен в приложении 2.

3.3 Применение библиотеки TPL к разработанному программному проекту

Библиотека распараллеливания задач (TPL) усовершенствует многопоточное программирование двумя основными способами. Во-первых, она упрощает создание и применение многих потоков. И во-вторых, она позволяет автоматически использовать несколько процессоров. Иными словами, TPL открывает возможности для автоматического масштабирования приложений с целью эффективного использования ряда доступных процессоров.

Библиотека TPL определена в пространстве имен System.Threading.Tasks. Но для работы с ней, обычно, требуется также включать в программу класс System.Threading, поскольку он поддерживает синхронизацию и другие средства многопоточной обработки, в том числе и те, что входят в класс Interlocked.

Применяя TPL, параллелизм в программу можно ввести двумя основными способами. Первый из них называется параллелизмом данных. При таком подходе одна операция над совокупностью данных разбивается на два параллельно выполняемых потока или больше, в каждом из которых обрабатывается часть данных.

Второй способ ввода параллелизма называется параллелизмом задач. При таком подходе две операции или больше выполняются параллельно. [17]

Для удобства, предлагается возможность расшифровать сразу четыре таблицы (см. Рис.21). Ввиду того, что поля с геоданными могут быть очень громоздкими, в целях повышения производительности, а соответственно и уменьшения времени ожидания, в нашей работе было принято решение применить библиотеку TPL.

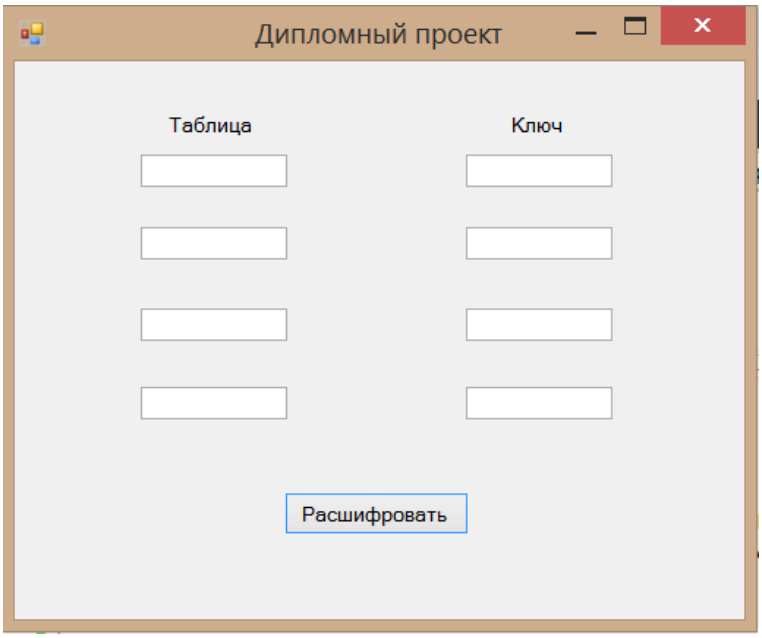


Рис.21. Главное окно клиентской программы

Так как при расшифровке таблицы применяются однотипные процедуры, используемый способ можно назвать своеобразным частным случаем

параллелизма данных, при котором операции искусственно разделяются на четыре потока. Таким образом, при наличии у клиента четырехъядерного процессора, каждое ядро будет расшифровывать по одной таблице (при наличии иного числа ядер TPL автоматически распределит нагрузку между доступными ядрами в динамическом режиме), что должно привести к уменьшению времени работы программы.

Серия тестов программы на двухъядерном процессоре показало, что применение библиотеки TPL повышает производительность примерно в 2 раза (см. Рис.22,23).

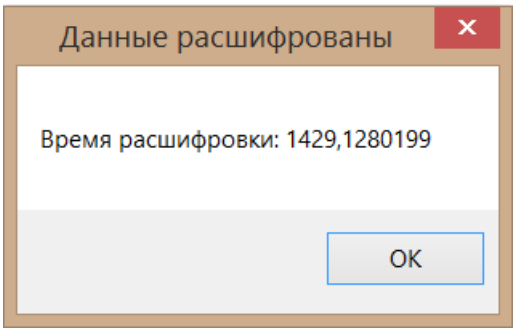


Рис.22. Время расшифровки последовательной программы

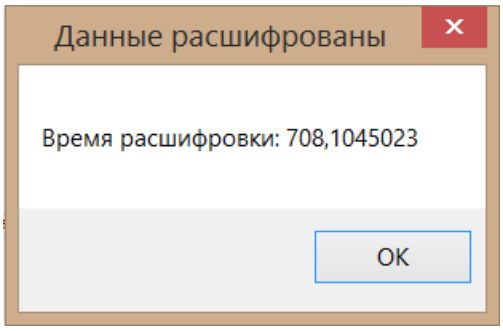


Рис.23. Время расшифровки программы с применением TPL

Стоит учесть, что в результате применения библиотеки TPL, нагрузка на центральный процессор значительно возрастает (см. Рис.24,25).

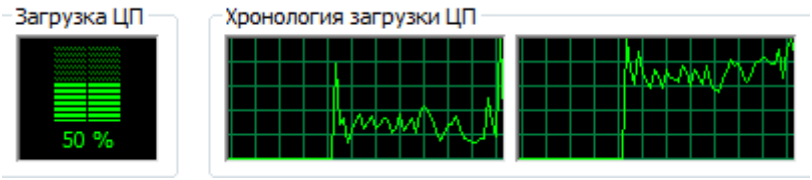


Рис.24. Загрузка ЦП программой с последовательной расшифровкой

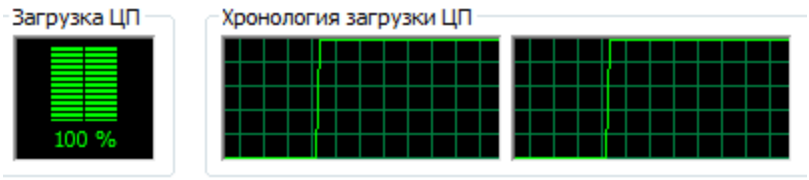


Рис.25. Загрузка ЦП программой с применением TPL

Полный листинг программы с применением библиотеки TPL приведен в приложении 3.

3.4 Импорт картографической базы данных

Существует два способа импорта картографической базы данных из ГИС QGIS в объектно-реляционный СУБД PostgreSQL. В первом случае таблицы картографической базы данных сначала сохраняются на жесткий диск пользователя, а лишь потом открываются в СУБД. Во втором же случае, требуемые данные открываются непосредственно в рабочей области QGIS.

3.4.1 Импорт с использованием менеджера БД

Для начала создается (см. Рис.26) и настраивается подключение QGIS с объектно-реляционной СУБД PostgreSQL.

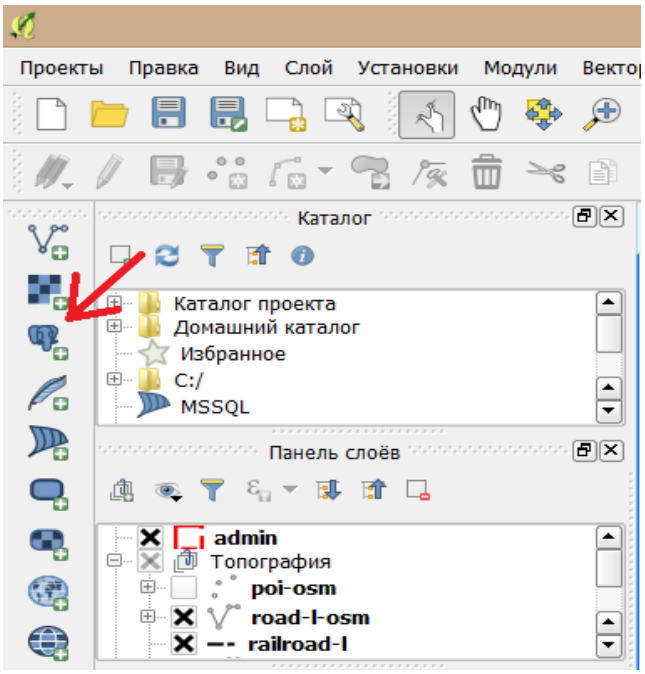


Рис.26. Создание подключения к PostgreSQL

В появившемся окне выбирается «Создать» для создания нового подключения (см. Рис.27).

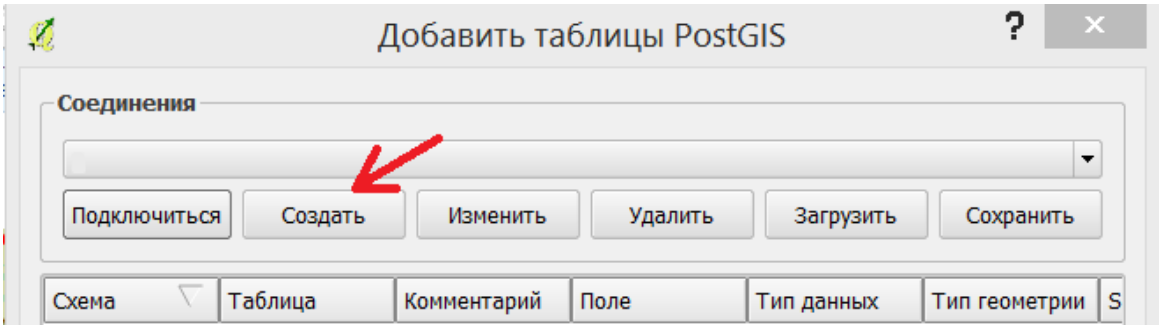


Рис.27. Создание нового подключения

Далее происходит настройка подключения (см. Рис.28). Необходимо указать следующие параметры:

- Имя: имя для данного соединения. Может совпадать с именем базы данных;
- Служба: Этот параметр может использоваться в качестве альтернативы для узла и порта (и, теоретически, базы данных). Настройка выполняется в файле pg_service.conf;
- Сервер: имя узла, на котором хранится база данных. Имя узла должно быть допустимым — таким, какие используют для сетевого доступа или для пинга узла. Если база данных находится на том же компьютере, что и QGIS, просто вводится localhost;
- Порт: номер порта, который «слушает» сервер базы данных PostgreSQL. По умолчанию используется порт 5432;
- База данных: имя базы данных;
- Режим SSL: настройка SSL-режима работы с сервером. Следует отметить, что значительного прироста скорости рендеринга слоя PostGIS можно достигнуть путём отключения SSL в менеджере соединений. Можно выбрать:
  - запретить: использовать только не зашифрованное SSL-соединение,
  - разрешить: будет произведена попытка установки не SSL-соединения, если она не удастся, будет использовано SSL-соединение,
  - предпочитать (по умолчанию): будет произведена попытка установки SSL-соединения, если она не удастся, будет использовано не SSL-соединение,
  - требовать: использовать только SSL-соединение;
- Пользователь: имя пользователя, которое используется для доступа к базе данных;
- Пароль: пароль, используемый вместе с именем пользователя для подключения к базе данных. [18]

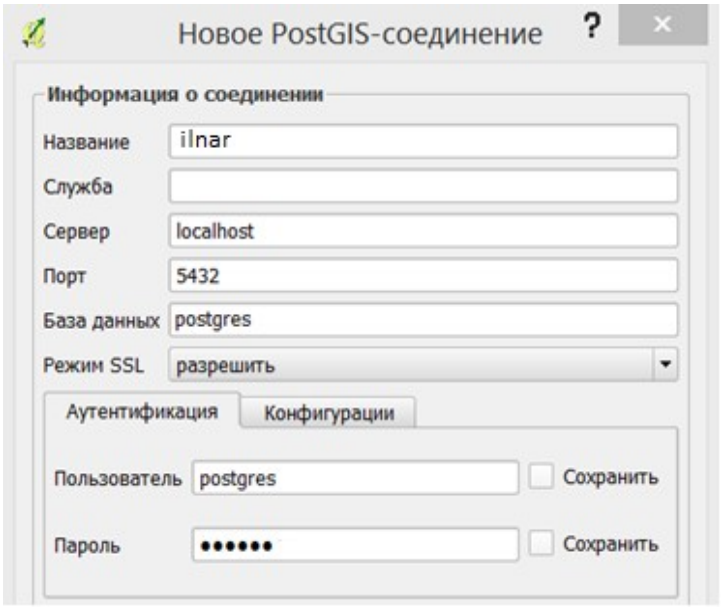


Рис.28. Настройка параметров соединения

После сохранения настроек следует подключиться к СУБД, при правильной настройке, QGIS подключится к заданной БД и станут доступны все содержащиеся в ней таблицы (см. Рис.29).

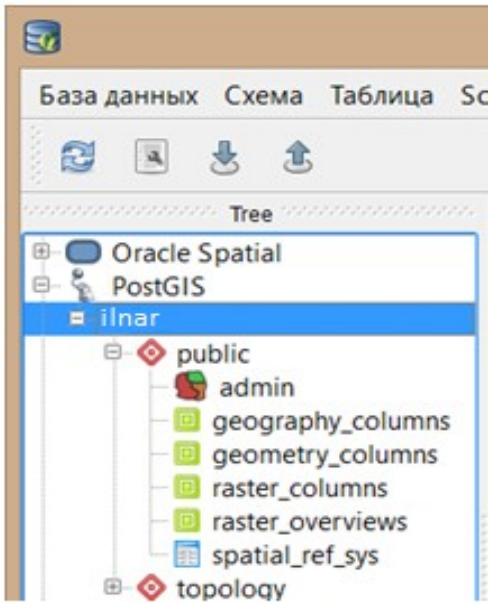


Рис.29. Окно подключения

После подключения, закрывается окно подключений и открывается инструмент «Менеджер БД» (см. Рис.30).

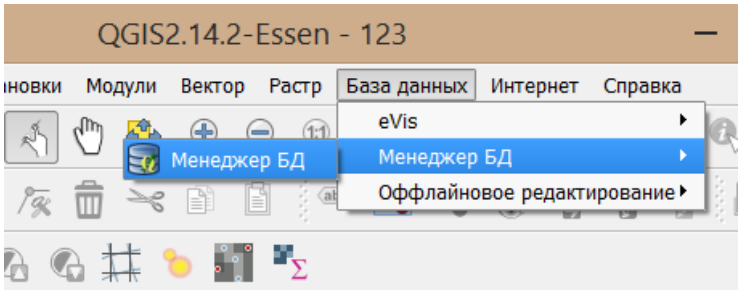


Рис.30. Запуск менеджера БД

С помощью инструмента «Менеджер БД» возможно экспортировать/импортировать данные между БД и QGIS. Под «Импортом» подразумевается импорт данных из QGIS в PostgreSQL, именно эта операция необходима в данной работе (см. Рис.31).

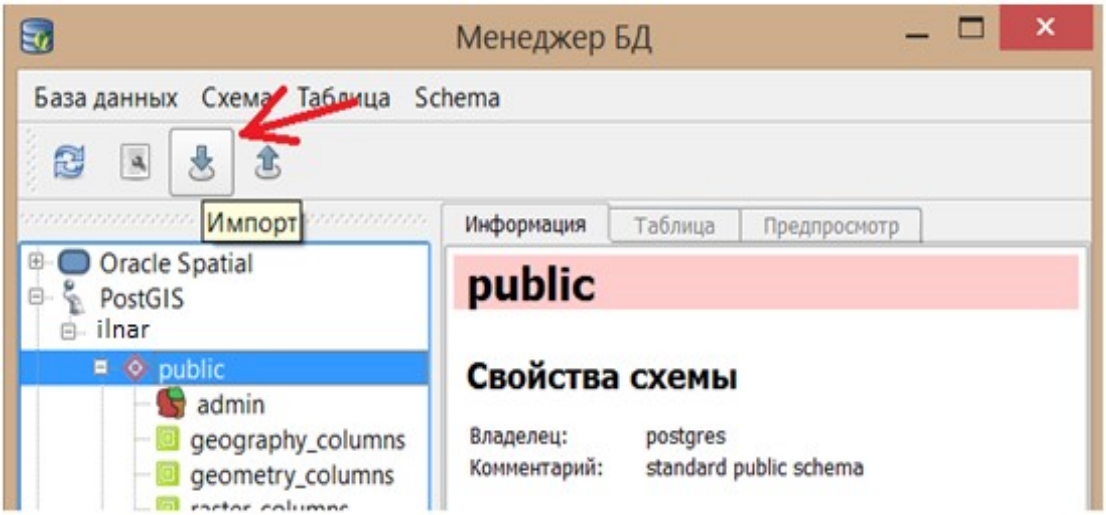


Рис.31. Импорт таблицы

Далее предлагается назначить каталог сохранения выбранной таблицы (см. Рис.32).

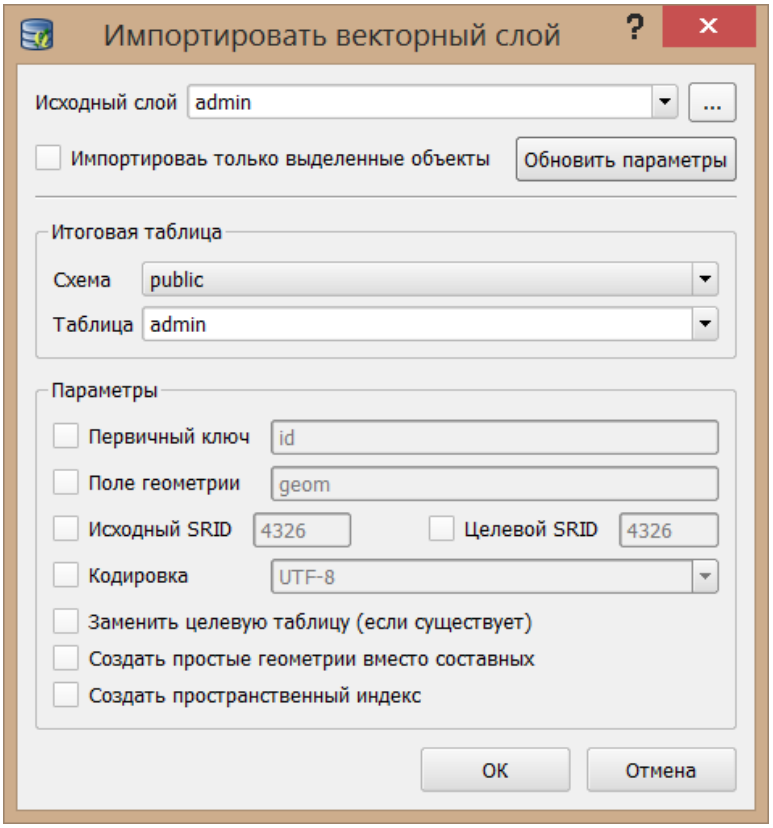


Рис.32. Выбор каталога импорта

После успешного сохранения данных на жестком диске появляется соответствующее сообщение (см. Рис.33).

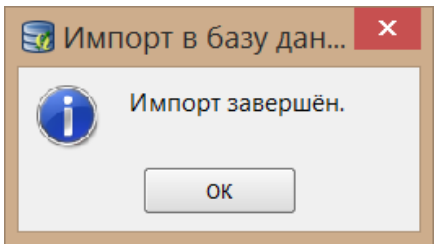


Рис.33. Успешный импорт

После этого возможно открыть сохраненную таблицу в PostgreSQL.

3.4.2 Импорт данных без предварительного сохранения на жестком диске

В ГИС QGIS так же предусмотрена возможность добавления требуемых слоев, с помощью расширения PostGIS, без предварительного сохранения на жестком диске (см. Рис.34).

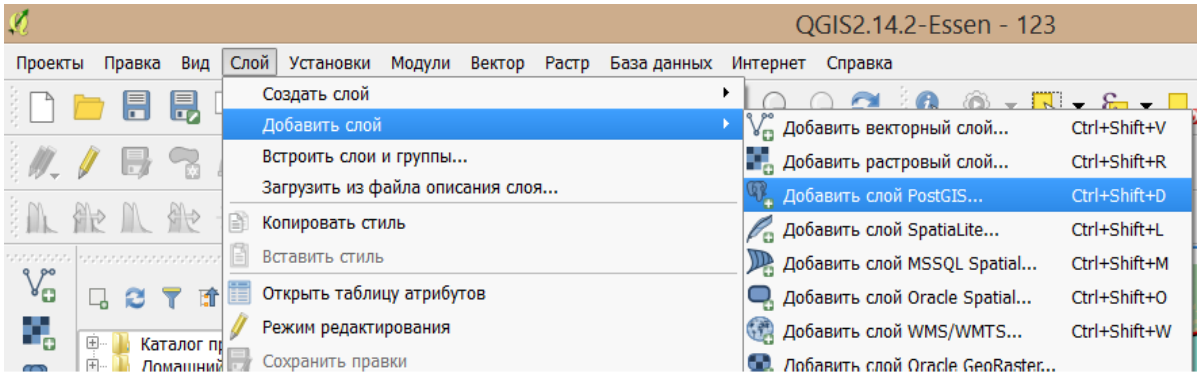


Рис.34. Добавление слоя с использованием PostGIS

Для этого требуется так же, как и в первом случае, подключиться к PostgreSQL, после чего в появившемся окне выбрать требуемую таблицу и нажать «Добавить» (см. Рис.35).

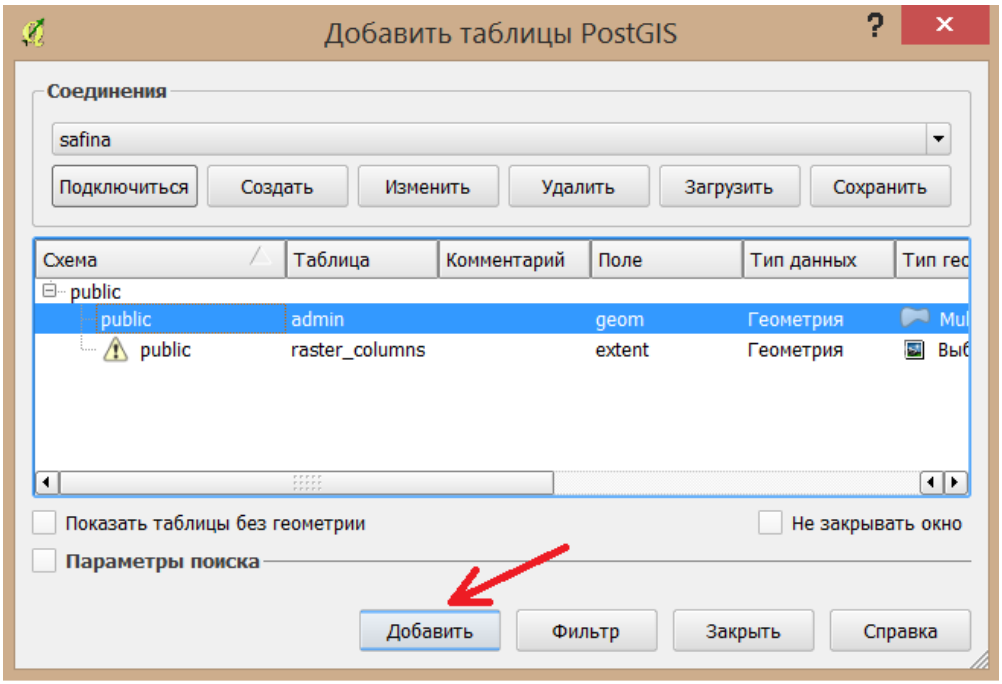


Рис.35. Выбор и добавление требуемого слоя

После этого, выбранный слой отобразится в рабочей области ГИС приложения (см. Рис.36).

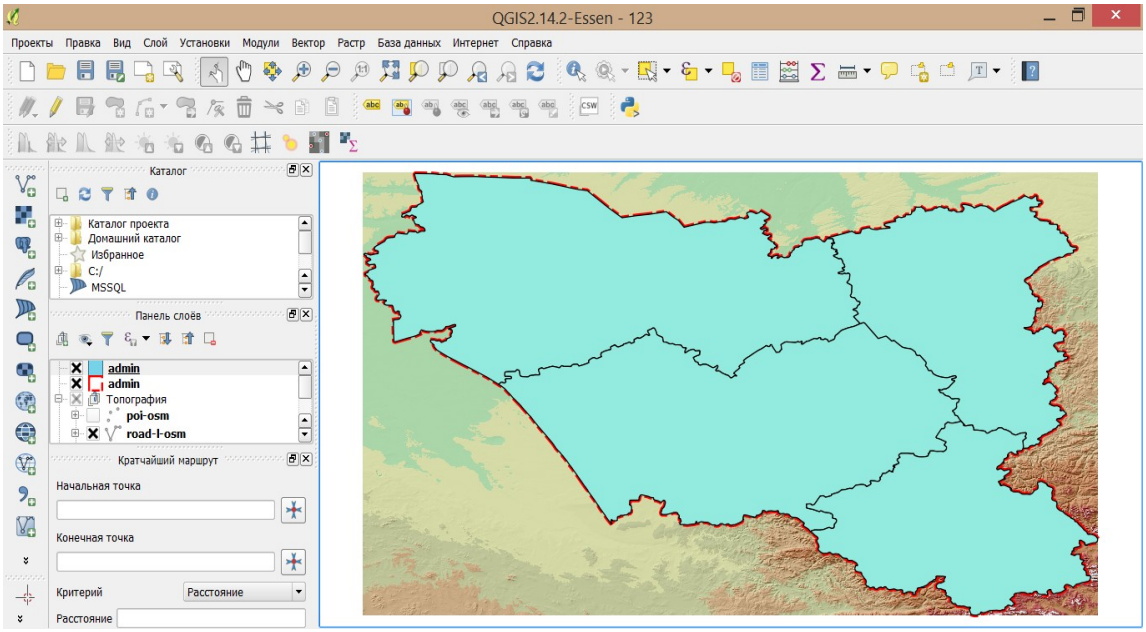


Рис.36. Рабочая область QGIS

ЗАКЛЮЧЕНИЕ

В ходе работы поставленные цели были выполнены и решены следующие задачи:

1. Раскрыта актуальность модуля импорта данных системы защиты картографических баз данных.
2. Разработан алгоритм расшифровки данных.
3. Организовано сетевое хранилище на базе объектно-реляционной СУБД PostgreSQL.
4. Произведена установка и конфигурирование объектно-реляционной СУБД PostgreSQL для удаленного доступа к хранилищу.
5. Импортирована картографическая база данных из ГИС QGIS.
6. Разработан программный модуль доступа к серверу и дешифрования файлов.
7. Применена библиотека TPL к разработанному программному проекту.

Для тестирования использовалась картографическая база данных Российской Федерации. Тестирование проводилось на одном узле с двумя ядрами на одном физическом процессоре.

По результатам работы я могу утверждать, что использование параллельного модуля запросов дает нам прирост в производительности. Используемая конфигурация предположительно не будет являться порогом масштабируемости для рассматриваемой системы. Следовательно, с увеличением количества процессоров и числа ядер на них предполагается дальнейшее повышение производительности.

СПИСОК ЛИТЕРАТУРЫ

1. ArcGIS [Электронный ресурс] // ESRI. 2016. URL: https://www.arcgis.com/features.

2. Mapinfo [Электронный ресурс] // Pitney Bowes. © 2016. URL: http://www.mapinfo.com.

3. MySQL [Электронный ресурс] // Oracle Corporation. © 2016. URL: http://www.mysql.com.

4. Pgcrypto [Электронный ресурс] // The PostgreSQL Global Development Group. © 1996-2016 URL: http://www.postgresql.org/docs/9.5/static/pgcrypto.

5. PostgreSQL [Электронный ресурс] // The PostgreSQL Global Development Group. © 1996-2016 URL: http://www.postgresql.org.

6. QGIS [Электронный ресурс] // Creative Commons Attribution-ShareAlike. © 2016. URL: http://www.qgis.org/ru/site.

7. SQLite [Электронный ресурс] // SQLITE.ORG. © 2002-2014 URL: http://www.sqlite.org/docs.html.

8. SQLite vs MySQL vs PostgreSQL: сравнение систем управления базами данных [Электронный ресурс] // DEVACADEMY. © 2016. URL: http://devacademy.ru/posts/sqlite-vs-mysql-vs-postgresql.

9. Бабенко Л.К., Басан А.С., Журкин И.Г., Макаревич О.Б. Защита данных геоинформационных систем: учеб. Пособие для студентов вузов / Под ред. И.Г. Журкина. – М.: Гелиос АРВ, 2010. – 336 с.

10. Бугаевский Л.М. Цветков В.Я. Геоинформационные системы: учебное пособие для ВУЗов. — М.: Златоуст, 2000. — 222с.

11. Вершинин И.С., Гибадуллин Р.Ф., Пыстогов С.В. Использование кластерных технологий при решении задач защиты картографических данных // Высокопроизводительные параллельные вычисления на кластерных системах (HPC–2009): материалы IX Междунар. конф.; Владимир. гос. ун-т. – Владимир, 2009. – С. 68–72.

12. Все о ГИС и их применении [Электронный ресурс] // GIStechnik. © 2012. URL: http://gistechnik.ru/publik.

13. Географическая информационная система [Электронный ресурс] // Википедия. Дата обновления 22.04.2016. URL: http://ru.wikipedia.org/wiki/Географическая_информационная_система.

14. Голубев И., Серов А. Защита цифровых карт с использованием электронных ключей HardLock [Электронный ресурс] //UcozServices. © 2016. URL: http://www.alex-serov.narod.ru/geo/articles/common/hardlock.htm.

15. Дубин М. Geosample: Открытый набор геоданных для различного ПО ГИС [Электронный ресурс] // © GIS-Lab. 2002 – 2015. Дата обновления 15.05.2014. URL: http://gis-lab.info/qa/geosample.html.

16. Основные сведения о типах пространственных данных [Электронный ресурс] // Microsoft Developer Network. © 2016. URL: http://msdn.microsoft.com/ru-ru/library/bb964711.aspx.

17. Параллельное программирование [Электронный ресурс] // Alexandr Erohin. © 2011-2016 URL: http://professorweb.ru/my/csharp/thread_and_files/level2/2_1.php.

18. Руководство пользователя QGIS [Электронный ресурс] // Creative Commons Attribution-ShareAlike. Дата обновления 04.12.2014. URL: http://docs.qgis.org/2.2/ru/docs/user_manual/index.

19. Скляров Д. В. С43 Искусство защиты и взлома информации. — СПб.: БХВ-Петербург, 2004. - 288 с.

20. Установка PostgreSQL [Электронный ресурс] // Виктор Вислобоков. © 2008-2010. URL: http://www.postgresql.org/docs/9.5/static/index.html.

21. Цветков В.Я. Геоинформационные системы и технологии. — М.: Финансы и статистика, 1998. —288с.

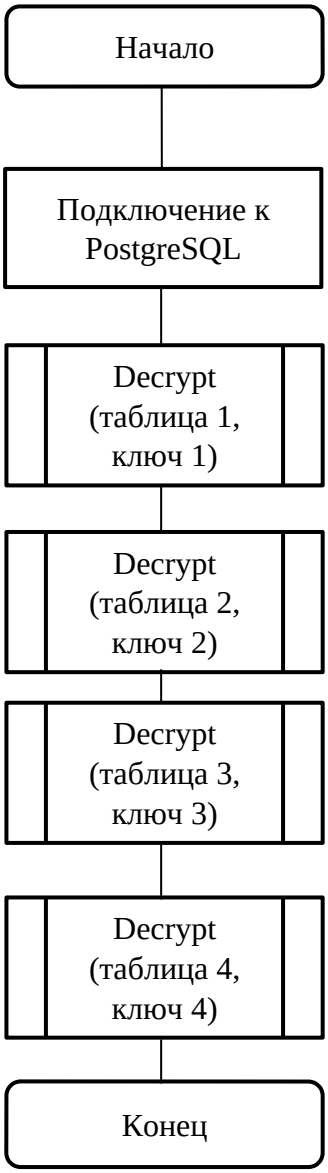
22. Ципилева Т.А. Геоинформационные системы: учебное пособие для студентов специальности «Государственное и муниципальное управление». — Томск.: Томский межвузовский центр дистанционного образования, 2010 — 121 с.

23. Шаши Шекхар, Санжей Чаула Основы Пространственных баз данных (перевод с английского). — М.: КУДИЦ – Образ. 2004 — 330 с.

ПРИЛОЖЕНИЕ 1

Блок-схема программы

Блок-схема основной программы



Блок-схема подпрограммы Decrypt



ПРИЛОЖЕНИЕ 2

Листинг клиентской программы

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Npgsql;
using Mono.Security;
using System.Threading.Tasks;
using System.Diagnostics;

namespace Дипломный_проект
{
   public partial class Form1 : Form
   {
      NpgsqlConnection conn;
      public Form1()
      {
         InitializeComponent();
         conn = new NpgsqlConnection("Server=localhost;Port=5432;User
ID=postgres;Password=postgres;Database=postgres"); // подключение к объектно-
реляционной СУБД PostgreSQL
         conn.Open();

      }
```

```
public void decrypt(Object atr)
{
    NpgsqlCommand cmd = new NpgsqlCommand("ALTER TABLE public.\"" + ((Object[])atr)[0] + "\"add  \"geom\" geometry", conn); // создание новой колонки типа geometry
    NpgsqlDataReader NpgDReader = cmd.ExecuteReader();
    cmd = new NpgsqlCommand("update public.\"" + ((Object[])atr)[0] + "\" set \"geotext\" = pgp_sym_decrypt(cast(public.\"" + ((Object[])atr)[0] + "\". \"geotext\" as bytea),cast('" + ((Object[])atr)[1] + "' as text))", conn); // расшифровка данных
    cmd.CommandTimeout = 1000;
    NpgDReader = cmd.ExecuteReader();
    cmd = new NpgsqlCommand("update public.\"" + ((Object[])atr)[0] + "\" set \"geom\" = ST_GeomfromEWKT(public.\"" + ((Object[])atr)[0] + "\". \"geotext\")", conn); // преобразование и занесение данных в созданную колонку
    cmd.CommandTimeout = 1000;
    NpgDReader = cmd.ExecuteReader();
    cmd = new NpgsqlCommand("ALTER TABLE public.\"" + ((Object[])atr)[0] + "\" drop \"geotext\"", conn); // удаление колонки типа text
    NpgDReader = cmd.ExecuteReader();
}

private void button_Click(object sender, EventArgs e)
{
    Stopwatch sw = new Stopwatch();
    sw.Start();
    decrypt(new Object[] { tabl1.Text, key1.Text });
    decrypt(new Object[] { tabl2.Text, key2.Text });
    decrypt(new Object[] { tabl3.Text, key3.Text });
    decrypt(new Object[] { tabl4.Text, key4.Text });
```

```
                sw.Stop();
                MessageBox.Show("Время расшифровки: " +
sw.Elapsed.TotalSeconds, "Данные расшифрованы");


            }
        }
    }
```

ПРИЛОЖЕНИЕ 3

Листинг программы с применением библиотеки TPL

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Npgsql;
using Mono.Security;
using System.Threading.Tasks;
using System.Diagnostics;


namespace Дипломный_проект
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
      public void decrypt(Object atr)
      {
                                          NpgsqlConnection conn = new
NpgsqlConnection("Server=localhost;Port=5432;User
ID=postgres;Password=postgres;Database=postgres"); // подключение к объектно-
реляционной СУБД PostgreSQL
```

```
            conn.Open();
                NpgsqlCommand cmd = new NpgsqlCommand("ALTER TABLE public.\"" + ((Object[])atr)[0] + "\"add \"geom\" geometry", conn); // создание новой колонки типа geometry
            NpgsqlDataReader NpgDReader = cmd.ExecuteReader();
             cmd = new NpgsqlCommand("update public.\"" + ((Object[])atr)[0] + "\" set \"geotext\" = pgp_sym_decrypt(cast(public.\"" + ((Object[])atr)[0] + "\". \"geotext\" as bytea),cast('" + ((Object[])atr)[1] + "' as text))", conn); // расшифровка данных
            cmd.CommandTimeout = 1000;
            NpgDReader = cmd.ExecuteReader();
             cmd = new NpgsqlCommand("update public.\"" + ((Object[])atr)[0] + "\" set \"geom\" = ST_GeomfromEWKT(public.\"" + ((Object[])atr)[0] + "\". \"geotext\")", conn); // преобразование и занесение данных в созданную колонку
            NpgDReader = cmd.ExecuteReader();
                    cmd = new NpgsqlCommand("ALTER TABLE public.\"" + ((Object[])atr)[0] + "\" drop \"geotext\"", conn); // удаление колонки типа text
            NpgDReader = cmd.ExecuteReader();
        }

        private void button_Click(object sender, EventArgs e)
        {
            Task[] tasks = new Task[4];
            tasks[0] = new Task(decrypt, new Object[] { tabl1.Text, key1.Text });
            tasks[1] = new Task(decrypt, new Object[] { tabl2.Text, key2.Text });
            tasks[2] = new Task(decrypt, new Object[] { tabl3.Text, key3.Text });
            tasks[3] = new Task(decrypt, new Object[] { tabl4.Text, key4.Text });
            Stopwatch sw = new Stopwatch();
            sw.Start();
            tasks[0].Start();
```

```
                tasks[1].Start();
                tasks[2].Start();
                tasks[3].Start();
                Task.WaitAll(tasks);
                sw.Stop();
                                MessageBox.Show("Время расшифровки: " +
sw.Elapsed.TotalSeconds, "Данные расшифрованы");

            }
        }
    }
```

ПРИЛОЖЕНИЕ 4

Презентация

Параллельная обработка пространственных запросов к защищенной картографической базе данных

Выполнил студент группы 4434
Мухаметшин И.И

Руководитель:
Пыстогов С.В

Казань, 2018г

Слайд 1. Тема дипломного проекта

Актуальность

В настоящее время во всем мире различными структурами и учреждениями широко используются средства пространственного анализа данных.

Картографическая продукция имеет свои особенности. Она отличается высокой себестоимостью работ по ее получению. Кроме того, карты могут содержать конфиденциальные сведения. Именно поэтому задача их защиты очень актуальна.

Разработка модулей обработки запросов к защищенным картографическим базам данных может существенно повысить эффективность работы с такими базами данных, увеличить быстродействие при достаточном уровне защиты.

Слайд 2. Актуальность

Цель и задачи

Цель:

Создание модуля расшифровки картографических данных, с последующим импортом их в ГИС приложение.

Задачи:

1. Раскрыть актуальность модуля импорта данных системы защиты картографических баз данных.

2. Разработать алгоритм расшифровки данных.

3. Организовать сетевое хранилище на базе объектно-реляционной СУБД PostgreSQL для повышения надежности хранения данных и возможности их восстановления вследствие выхода из строя оборудования.

4. Произвести установку и конфигурирование объектно-реляционной СУБД PostgreSQL для удаленного доступа к хранилищу.

5. Импортировать картографическую базу данных в ГИС QGIS.

6. Разработать программный модуль доступа к серверу и дешифрования файлов.

7. Применить библиотеку TPL к разработанному программному проекту

Слайд 3. Цель работы

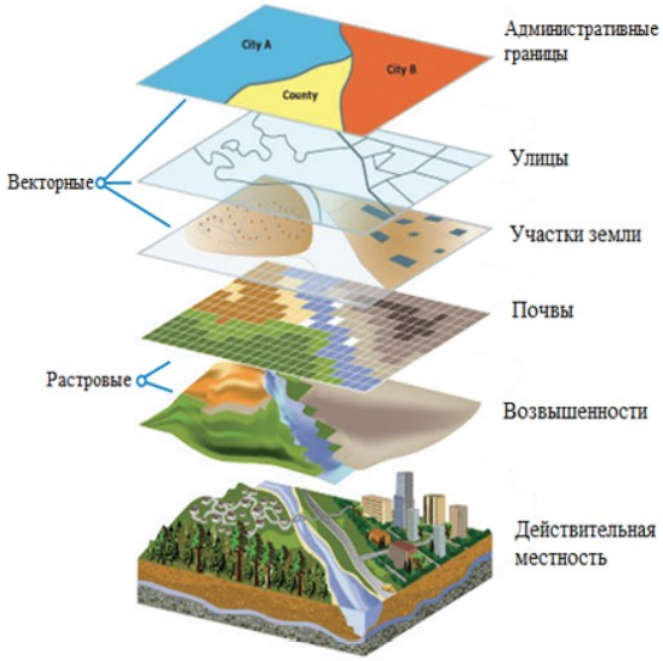
Геоинформационные системы (ГИС)

ГИС – это системы, предназначенные для сбора, хранения, анализа и графической визуализации пространственных данных и связанной с ними информации о представленных в них объектах.



Слайд 4. Геоинформационные системы

Геоинформационные системы (ГИС)



Данные в геоинформационных системах хранятся в виде набора тематических слоев, которые объединены на основе их географического положения. Этот гибкий подход и возможность геоинформационных систем работать как с векторными, так и с растровыми моделями данных, эффективен при решении любых задач, касающихся пространственной информации.

Слайд 5. Геоинформационные системы

Проблемы информационной безопасности картографических данных и существующие решения

Формирование векторной модели данных картографии является сложной задачей. Сами же данные картографии отличаются высокой себестоимостью работ по их получению и содержащейся информации и, соответственно, нуждаются в информационной безопасности. Необходимо ограничивать несанкционированный доступ к ним с целью хищения, нарушения целостности, а так же иметь возможность их восстанавливать в случае утраты по техническим причинам.

Наиболее надежным способом защиты данного типа, разработанным к настоящему времени является использование специальных электронных ключей. Одним из самых распространенных типов продукции такого вида являются электронные ключи Hardlock, выпускаемые израильской фирмой Aladdin.

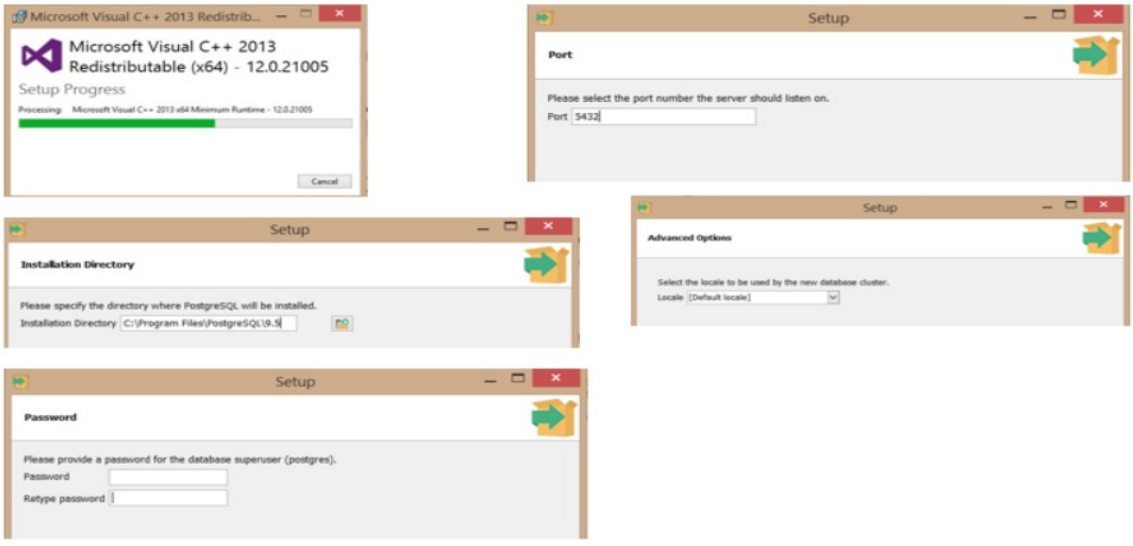
Слайд 6. Проблемы информационной безопасности картографических данных и существующие решения

Архитектура сервера ЗКБД

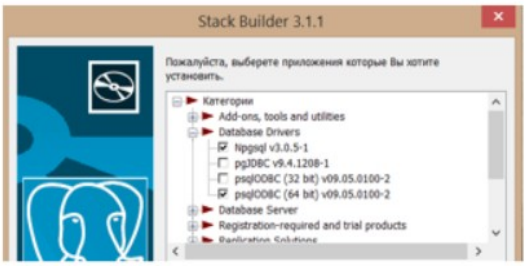


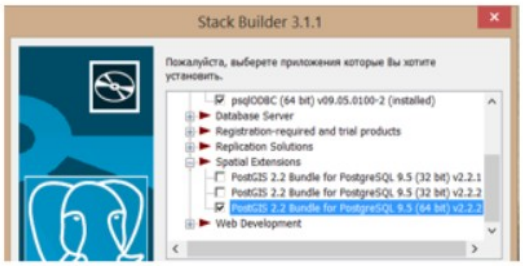
Слайд 7. Архитектура сервера ЗКБД

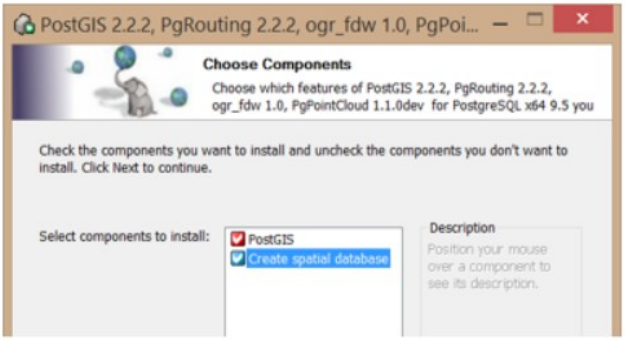
Установка и настройка объектно-реляционной СУБД PostgreSQL



Слайд 8. Установка и настройка объектно-реляционной СУБД PostgreSQL



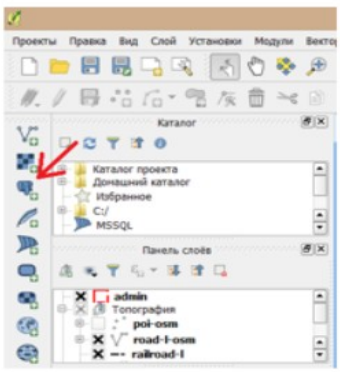


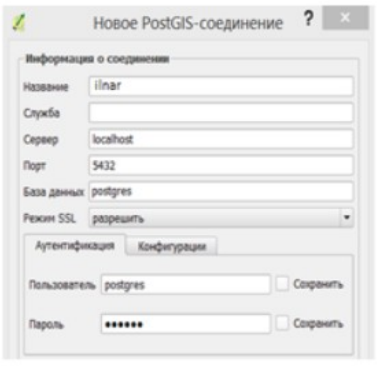


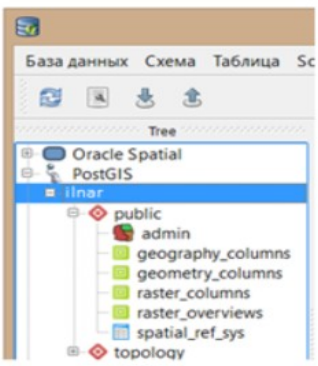
Слайд 9. Установка и настройка объектно-реляционной СУБД PostgreSQL

ИМПОРТ С ИСПОЛЬЗОВАНИЕМ МЕНЕДЖЕРА БД

1. Создается и настраивается подключение QGIS с объектно-реляционной СУБД PostgreSQL;
2. Происходит настройка подключения;
3. После сохранения настроек QGIS подключается к заданной БД и становятся доступны все содержащиеся в ней таблицы.



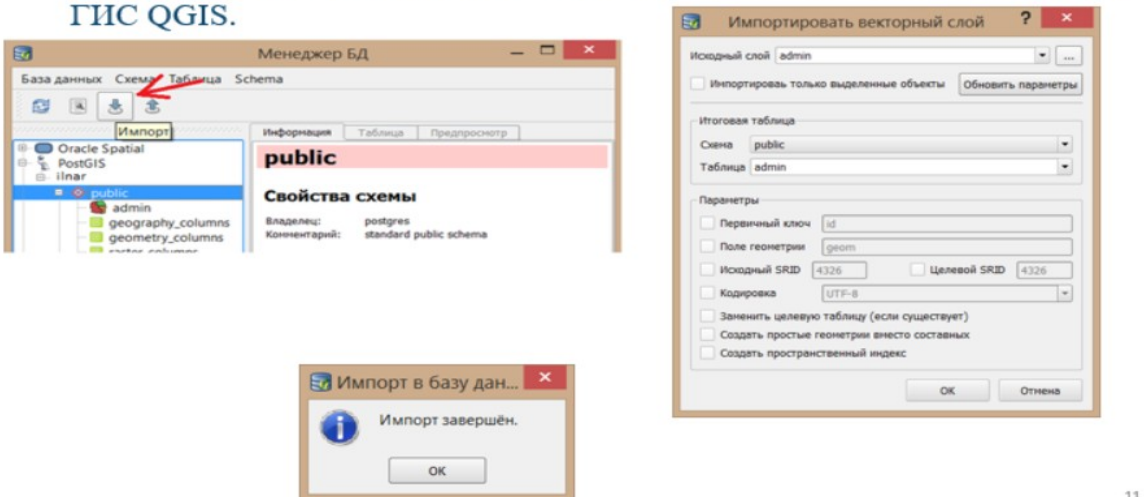




Слайд 10. Импорт с использованием менеджера БД

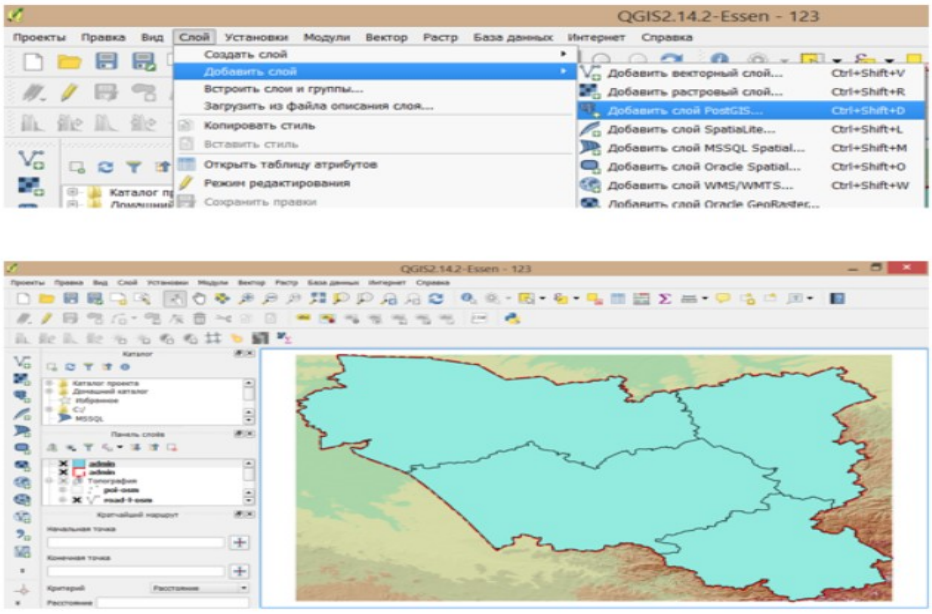
С помощью инструмента «Менеджер БД» возможно экспортировать/импортировать данные между БД и QGIS. Под «Импортом» подразумевается импорт данных из QGIS в PostgreSQL, именно эта функция вызывается.

После успешного сохранения данных на жестком диске появляется соответствующее сообщение, и можно открыть сохраненную таблицу в ГИС QGIS.



Слайд 11. Импорт с использованием менеджера БД

ИМПОРТ ДАННЫХ БЕЗ ПРЕДВАРИТЕЛЬНОГО СОХРАНЕНИЯ НА ЖЕСТКОМ ДИСКЕ



Слайд 12. Импорт данных без предварительного сохранения на жестком диске

РЕАЛИЗАЦИЯ МОДУЛЯ РАСШИФРОВКИ КАРТОГРАФИЧЕСКОЙ БАЗЫ ДАННЫХ

Основу проекта составляет шифрование определённых колонок. В нашем случае зашифрованными являются лишь столбцы типа geometry, которые представляют наибольшую ценность, так как содержат все географические данные.

Таким образом, на сервере хранится таблица с преобразованным в текст и зашифрованным полем geom, которое хранит в себе все геометрические данные.

id_0 integer	name character varying(50)	id integer	geotext text
1	Кемеровская област	0	\xc30d04070302deb8a8398ef9c3b27dd2ee01d21a8917
2	Новосибирская обла	1	\xc30d04070302fef5c3dfd78ac06376d2ee01f5312607
3	Алтайский край	2	\xc30d04070302821d24cb66bd951166d2ee019d7f187b
4	Республика Алтай	3	\xc30d040703026b57b8ee6f9606c46cd2ee01f34efb79

Слайд 13. Реализация модуля расшифровки картографической БД

РЕАЛИЗАЦИЯ МОДУЛЯ РАСШИФРОВКИ КАРТОГРАФИЧЕСКОЙ БАЗЫ ДАННЫХ

Первым делом предстоит расшифровать интересующий нас столбец. Для этого используется функция модуля pgcrypto - pgp_sym_decrypt. Для успешной расшифровки указывается тот же ключ, что использовался при зашифровке данных.

В случае верного ввода ключа, произойдет расшифровка интересующего нас поля. Однако, оно по-прежнему будет типа text.

id_0 integer	name character varying(50)	id integer	geotext text
1	Кемеровская област	0	SRID=4326;MULTIPOLYGON(((88.6241455078125 56.8
2	Новосибирская обла	1	SRID=4326;MULTIPOLYGON(((84.4106750488281 56.0
3	Алтайский край	2	SRID=4326;MULTIPOLYGON(((85.0794677734375 54.4
4	Республика Алтай	3	SRID=4326;MULTIPOLYGON(((88.3648834228516 52.1

Слайд 14. Реализация модуля расшифровки картографической БД

Следующим шагом требуется преобразовать получившееся уже незашифрованное текстовое поле в геометрический тип данных. Это можно сделать используя геометрический конструктор PostgreSQL, а конкретно функцию ST_GeomfromEWKT.

id_0 integer	name character varying(50)	id integer	geotext text	geom geometry
1	Кемеровская област	0	SRID=4	0106000020E6100000010000000103000000010000007D06
2	Новосибирская обла	1	SRID=4	0106000020E6100000010000000103000000010000000F107
3	Алтайский край	2	SRID=4	0106000020E6100000010000000103000000010000003D07
4	Республика Алтай	3	SRID=4	0106000020E6100000010000000103000000010000007906

id_0 integer	name character varying(50)	id integer	geom geometry
1	Кемеровская област	0	0106000020E6100000010000000103000000010000007D06
2	Новосибирская обла	1	0106000020E6100000010000000103000000010000000F107
3	Алтайский край	2	0106000020E6100000010000000103000000010000003D07
4	Республика Алтай	3	0106000020E6100000010000000103000000010000007906

Слайд 15. Реализация модуля расшифровки картографической БД

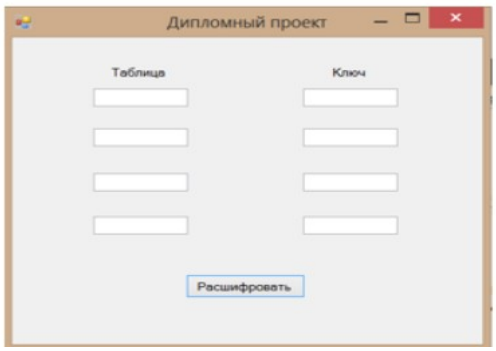
Применение библиотеки TPL к разработанному программному проекту

TPL открывает возможности для автоматического масштабирования приложений с целью эффективного использования ряда доступных процессоров.

Библиотека параллельных задач представляет собой набор открытых типов в пространствах имен System.Threading и System.Threading.Tasks. Цель TPL – повышение производительности труда разработчиков за счет упрощения процедуры добавления параллелизма в приложения.

Слайд 16. Применение библиотеки TPL к разработанному программному проекту

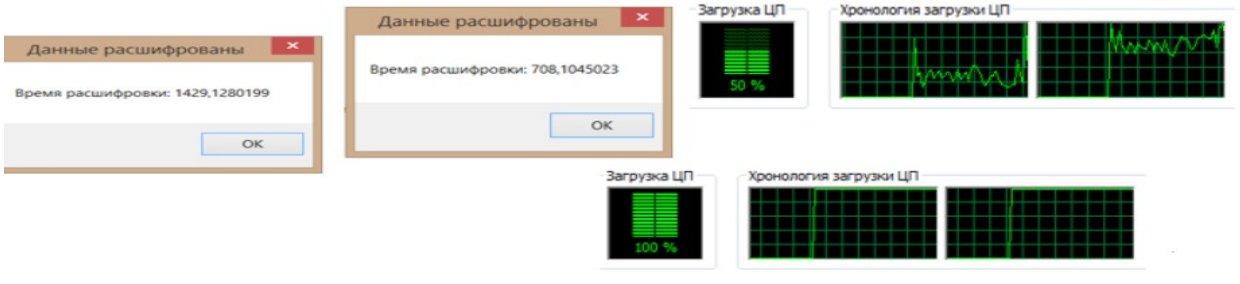
В данной программе предлагается возможность расшифровать сразу четыре таблицы. Ввиду того, что поля с геоданными могут быть очень громоздкими, в целях повышения производительности, а соответственно и уменьшения времени ожидания, в нашей работе было принято решение применить библиотеку TPL. Так как при расшифровке таблицы применяются однотипные процедуры, используемый способ можно назвать своеобразным частным случаем параллелизма данных, при котором операции искусственно разделяются на четыре потока.



Слайд 17. Применение библиотеки TPL к разработанному программному проекту

Серия тестов программы на двухъядерном процессоре показало, что применение библиотеки TPL повышает производительность примерно в 2 раза.

В результате применения библиотеки TPL, нагрузка на центральный процессор значительно возрастает.



Слайд 18. Применение библиотеки TPL к разработанному программному проекту

Заключение

Сделан анализ использования параллельного модуля запросов: он дает нам прирост в производительности.

Используемая конфигурация предположительно не будет являться порогом масштабируемости для рассматриваемой системы. Следовательно, с увеличением количества процессоров и числа ядер на них предполагается дальнейшее повышение производительности.

Для достижения цели в работе *решены задачи*:

- Произведена установка и наладка подключения объектно-реляционной СУБД PostgreSQL и ГИС QGIS.
- Разработан и реализован на языке C# клиентский модуль для расшифровки защищенных файлов.
- Применена библиотека TPL к разработанному программному проекту.
- Произведен импорт данных в ГИС QGIS.

Слайд 19. Заключение