

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ПРОСТРАНСТВЕННЫХ ЗАПРОСОВ К ЗАЩИЩЕННОЙ КАРТОГРАФИЧЕСКОЙ БАЗЕ ДАННЫХ

Выполнил студент группы 4434

Мухаметшин И.И

Руководитель:

Пыстогов С.В

Казань, 2018г

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.

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

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.



Геоинформационные системы (ГИС)



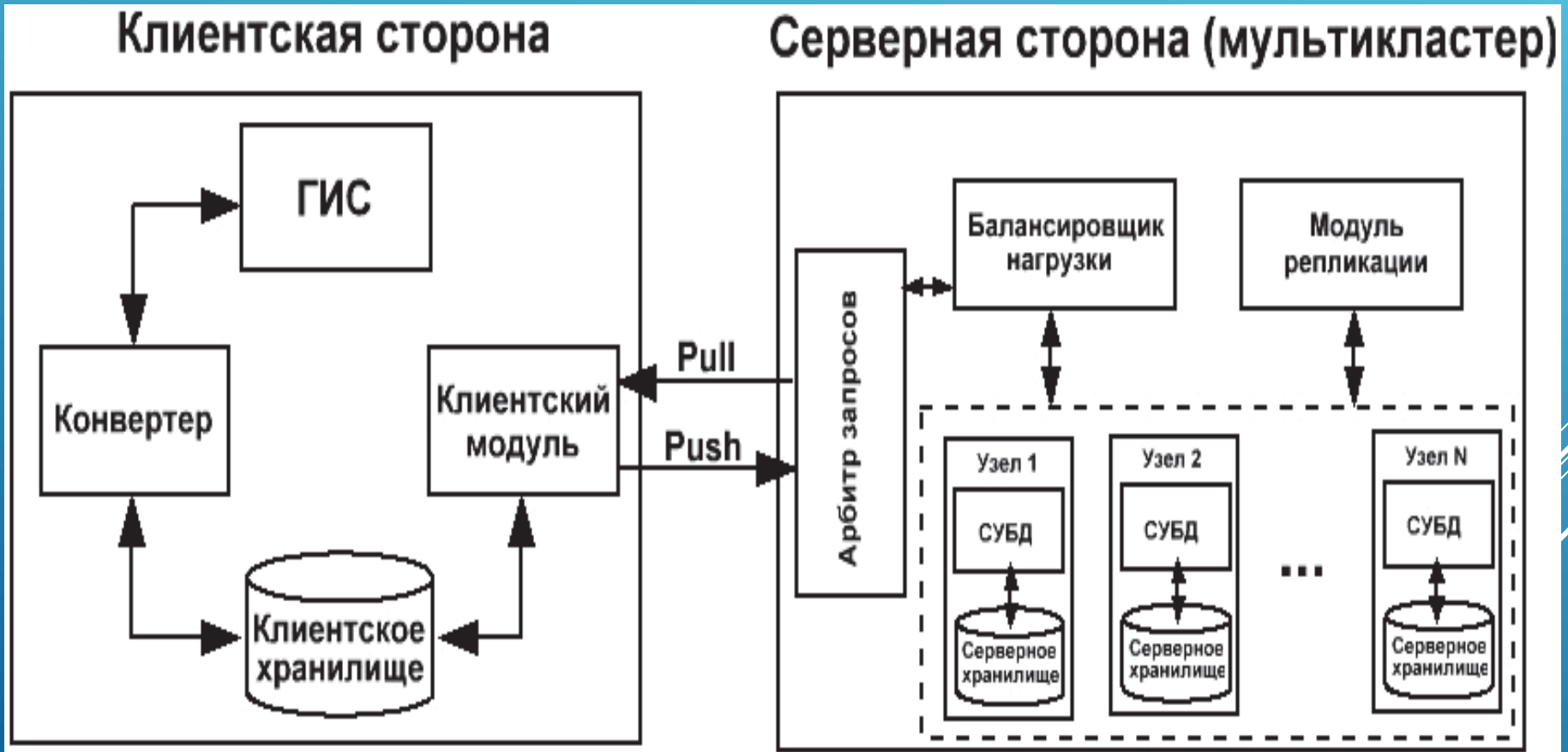
Данные в геоинформационных системах хранятся в виде набора тематических слоев, которые объединены на основе их географического положения. Этот гибкий подход и возможность геоинформационных систем работать как с векторными, так и с растровыми моделями данных, эффективен при решении любых задач, касающихся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

Проблемы информационной безопасности картографических данных и существующие решения

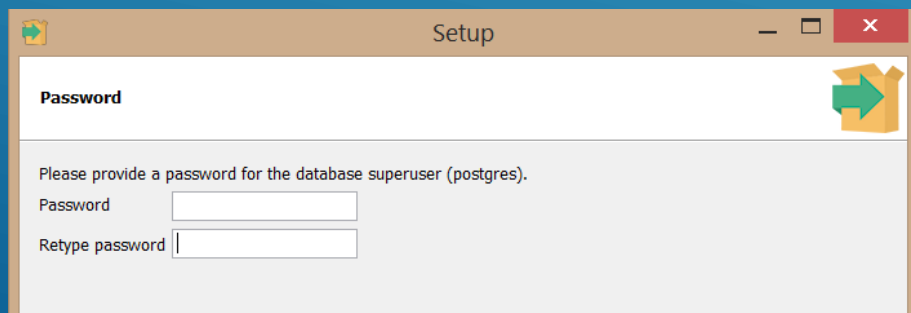
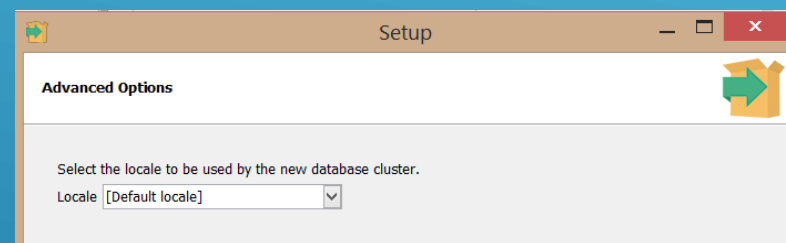
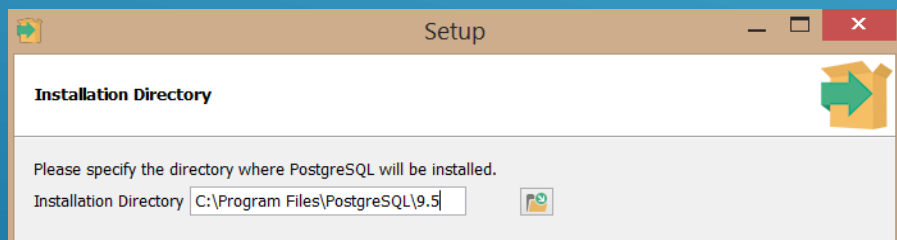
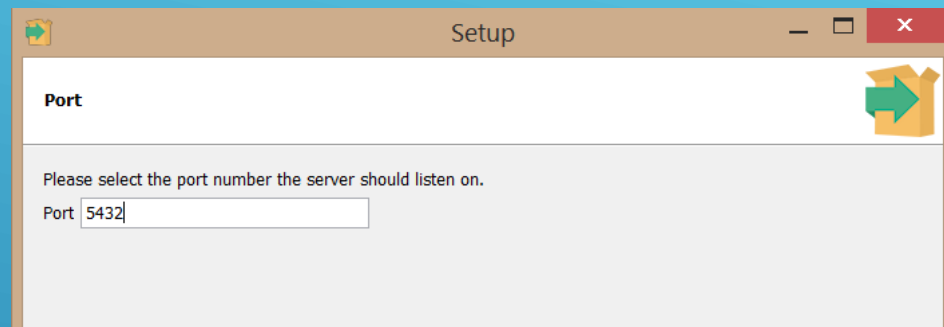
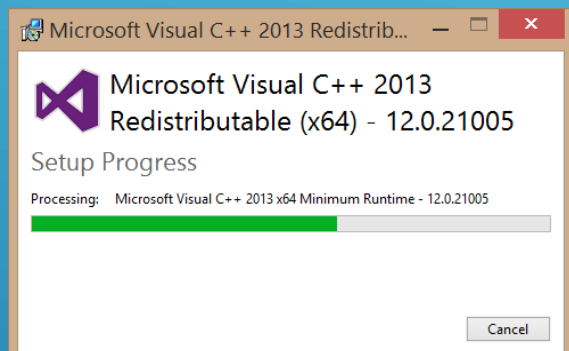
Формирование векторной модели данных картографии является сложной задачей. Сами же данные картографии отличаются высокой себестоимостью работ по их получению и содержащейся информации и, соответственно, нуждаются в информационной безопасности. Необходимо ограничивать несанкционированный доступ к ним 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.

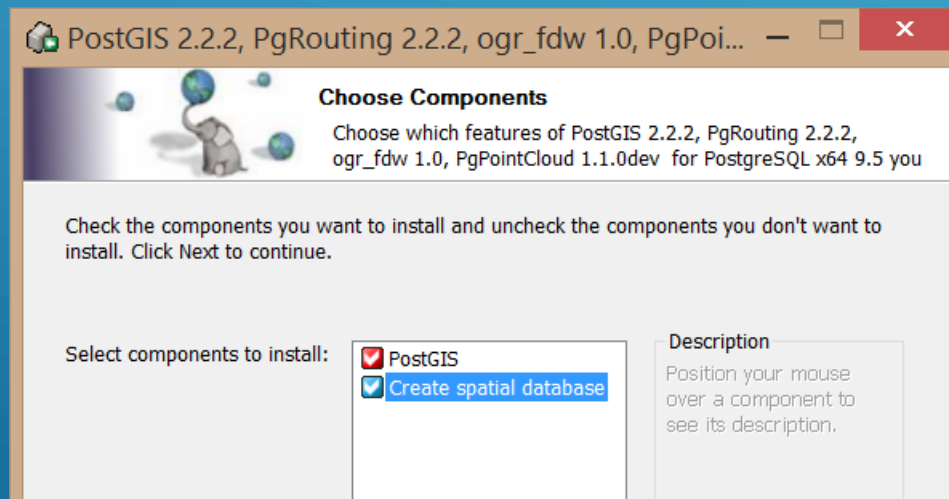
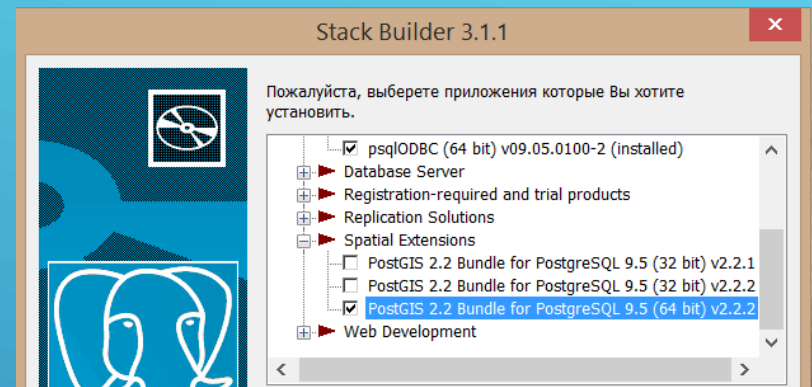
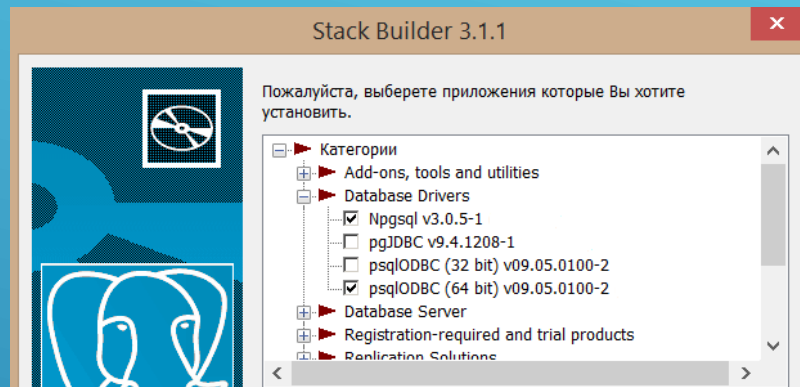
Архитектура сервера ЗКБД



Установка и настройка объектно-реляционной СУБД PostgreSQL

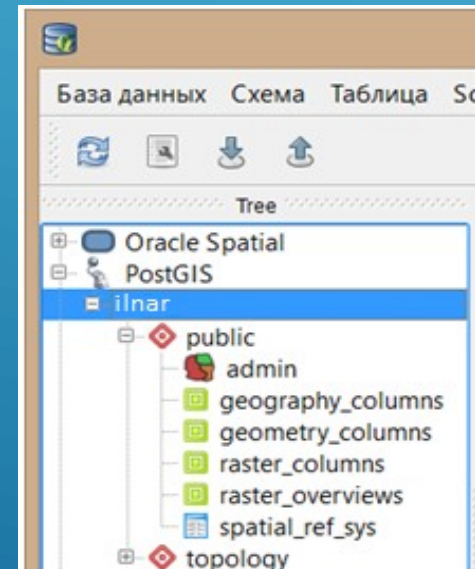
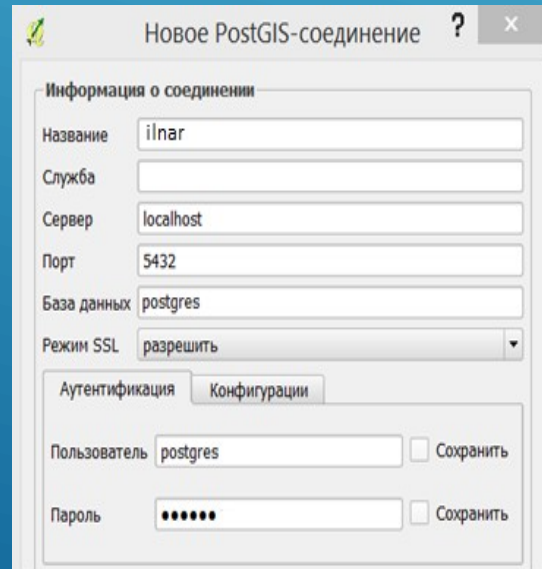
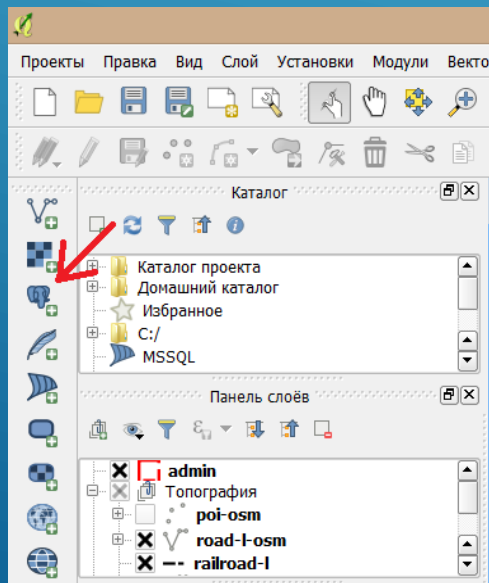


Установка и настройка объектно-реляционной СУБД PostgreSQL



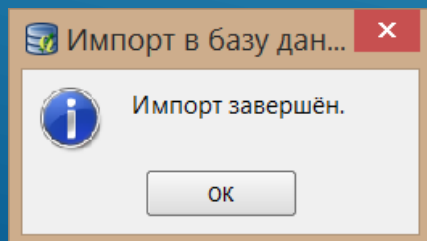
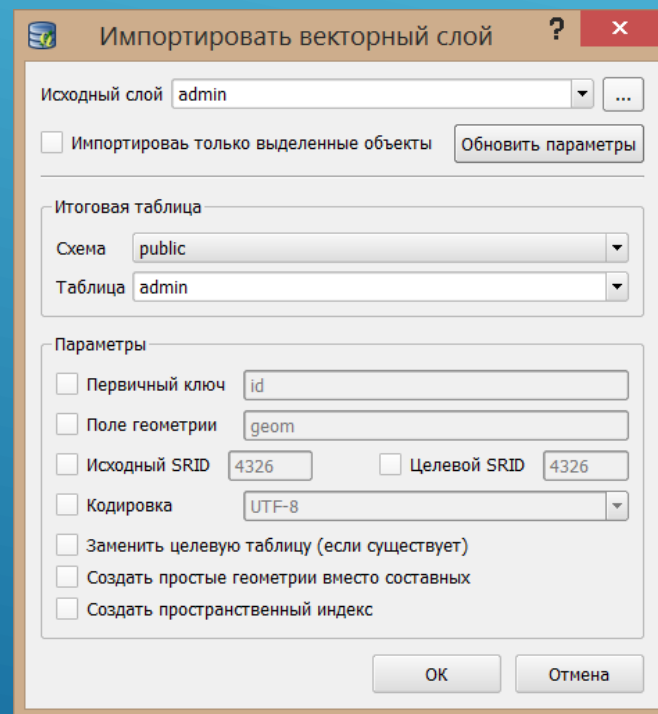
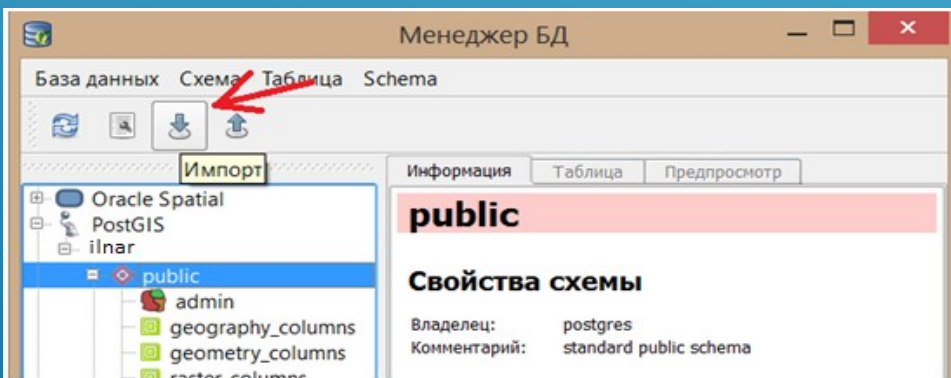
ИМПОРТ С ИСПОЛЬЗОВАНИЕМ МЕНЕДЖЕРА БД

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.

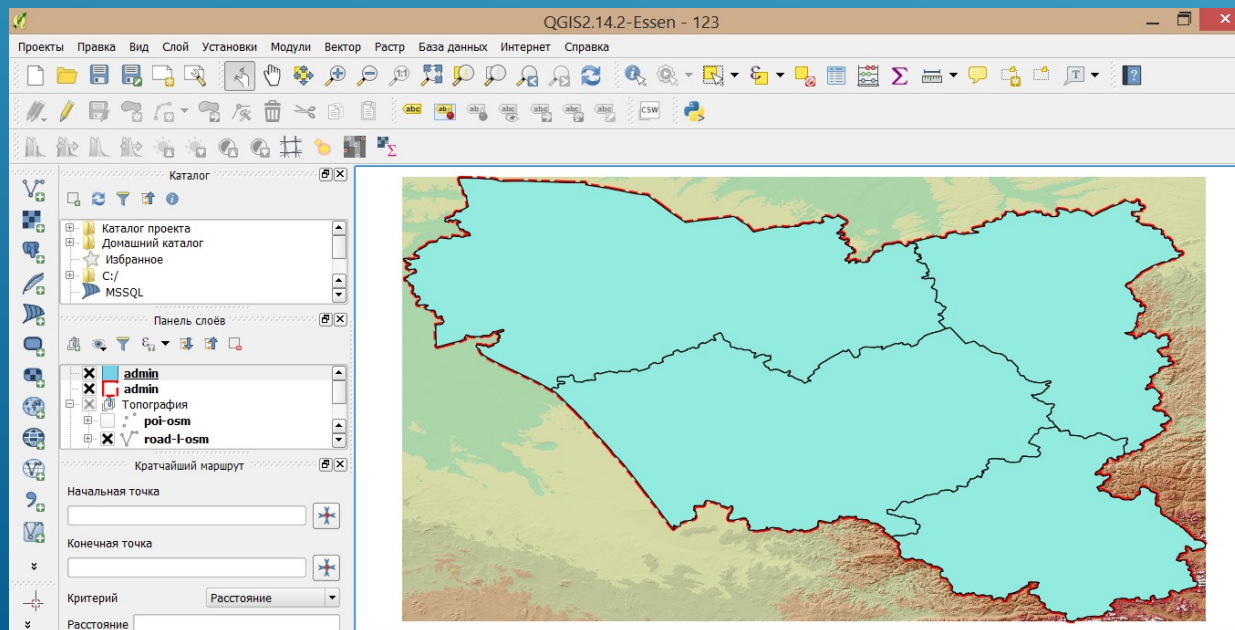
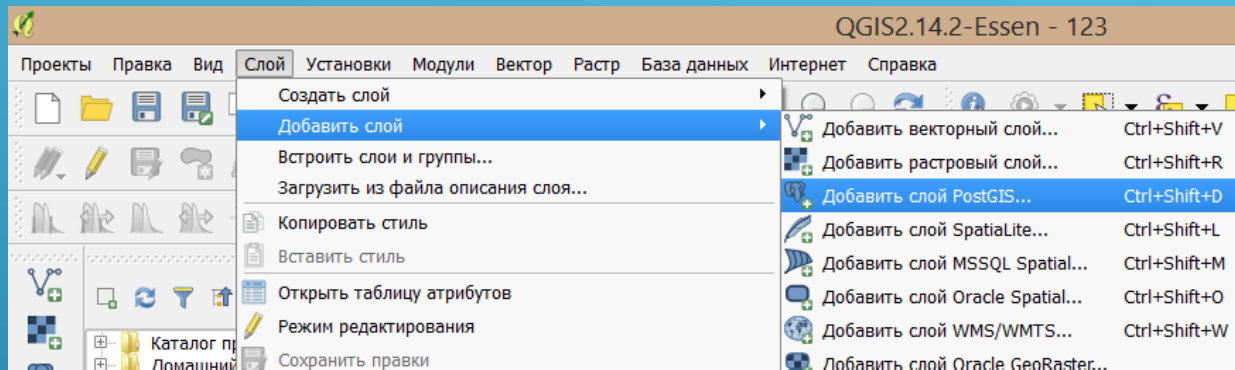


С помощью инструмента «Менеджер БД» возможно экспортировать/импортировать данные между БД и QGIS. Под «Импортом» подразумевается импорт данных из QGIS в PostgreSQL, именно эта функция вызывается.

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



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.

<code>id_0</code> integer	<code>name</code> character varying(50)	<code>id</code> integer	<code>geotext</code> text
1	Кемеровская област	0	\xc3d04070302deb8a8398ef9c3b27dd2ee01d21a8917
2	Новосибирская обла	1	\xc3d04070302fef5c3dfd78ac06376d2ee01f5312607
3	Алтайский край	2	\xc3d04070302821d24cb66bd951166d2ee019d7f187k
4	Республика Алтай	3	\xc3d040703026b57b8ee6f9606c46cd2ee01f34efb79

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

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 область	0	SRID=4326;0106000020E6100000010000000103000000010000007D060	
2	Новосибирская обла	1	SRID=4326;0106000020E610000001000000010300000001000000F1070	
3	Алтайский край	2	SRID=4326;0106000020E6100000010000000103000000010000003D070	
4	Республика Алтай	3	SRID=4326;0106000020E61000000100000001030000000100000079060	

id_0 integer	name character varying(50)	id integer	geom geometry
1	Кемеровская область	0	0106000020E6100000010000000103000000010000007D060
2	Новосибирская обла	1	0106000020E610000001000000010300000001000000F1070
3	Алтайский край	2	0106000020E6100000010000000103000000010000003D070
4	Республика Алтай	3	0106000020E61000000100000001030000000100000079060

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.

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.

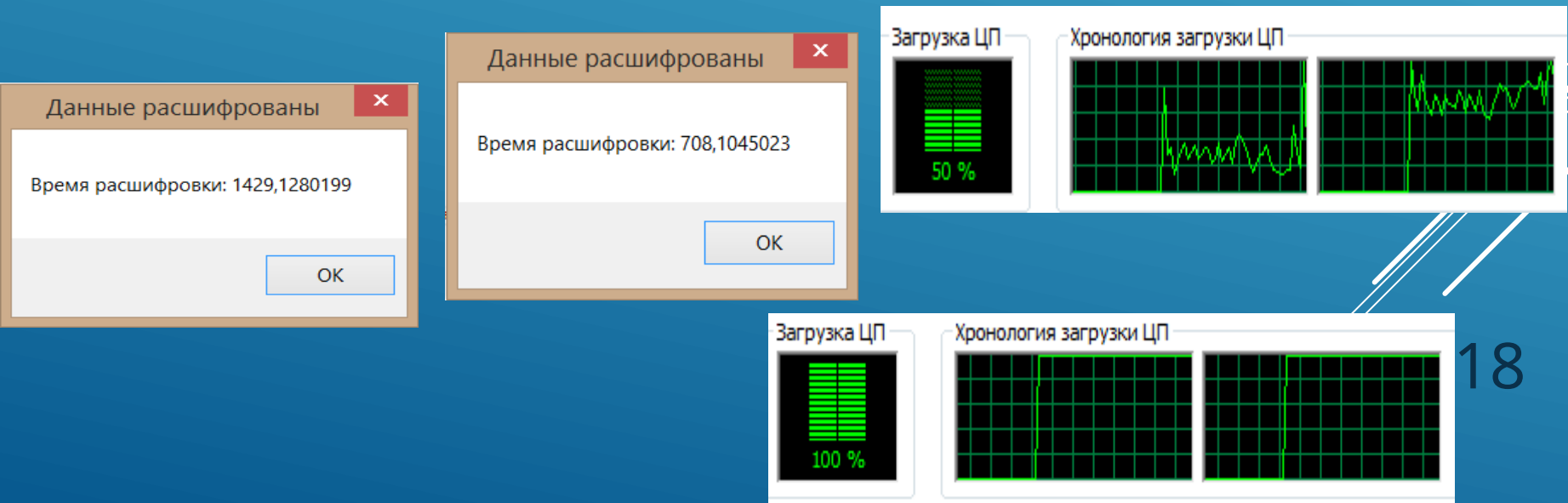
Дипломный проект

Таблица	Ключ

Расшифровать

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.



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.

Спасибо за внимание!