Санкт-Петербургский государственный университет

**НИКОЛЕНКО Светлана Владимировна**

**Выпускная квалификационная работа**

**ГЕОИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРИ СОЗДАНИИ ЕДИНОЙ КАРТОГРАФИЧЕСКОЙ ОСНОВЫ ГОСУДАРСТВЕННОГО КАДАСТРА НЕДВИЖИМОСТИ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ПРОСТРАНСТВЕННЫХ ДАННЫХ В РАМКАХ РЕАЛИЗАЦИИ ПРОЕКТОВ НАЦИОНАЛЬНОЙ ТЕХНОЛОГИЧЕСКОЙ ИНИЦИАТИВЫ**

Основная образовательная программа бакалавриата

«Картография и геоинформатика»

Профиль «Геоинформатика»

Научный руководитель: ассис.,

БЛЯХАРСКИЙ Дмитрий Петрович

Рецензент:

КАЗАКОВ Эдуард Эдуардович

Санкт-Петербург

2018

**Оглавление**

[Список принятых сокращений 4](#_Toc516165005)

[Введение 6](#_Toc516165006)

[Глава 1. Организация кадастрового учета в Российской Федерации. 9](#_Toc516165007)

[1.1. Документы, регулирующие правовые отношения в области кадастрового учета. 9](#_Toc516165008)

[1.2. Государственный кадастр недвижимости (ГКН) и единый государственный реестр недвижимости (ЕГРН), публичная кадастровая карта (ПКК), единая электронная картографическая основа (ЕЭКО). 13](#_Toc516165009)

[1.3. Кадастровые планы территорий (КПТ). Особенности составления и содержимое. Структура XML-документа. 17](#_Toc516165010)

[Глава 2. Анализ кадастровых данных. 23](#_Toc516165011)

[2.1. Примеры основных ошибок кадастровой информации. 23](#_Toc516165012)

[2.2. Преимущества подхода с использованием системы управления базами пространственных данных (СУБД). 26](#_Toc516165013)

[Глава 3. Применение СУБД для хранения, анализа и обработки кадастровых данных. 29](#_Toc516165014)

[3.1. Обзор и сравнение существующих систем управления базами пространственных данных. 29](#_Toc516165015)

[3.2. СУБД PostgreSQL - возможности и особенности 34](#_Toc516165016)

[3.3. Методика создания базы пространственных данных на примере кадастровых данных. 41](#_Toc516165017)

[Глава 4. Подготовка и анализ данных. 54](#_Toc516165018)

[4.1. Конвертация данных в формат ESRI shapefile. Способы импорта в базу данных. 54](#_Toc516165019)

[4.2. Интерфейсы подключения к базе данных: pgAdmin IV и QGIS. 56](#_Toc516165020)

[4.3. Применение методики поиска ошибок исходных кадастровых данных и варианты использования конечных данных на примере территории одного субъекта РФ. 59](#_Toc516165021)

[Заключение 63](#_Toc516165022)

[Список литературы 64](#_Toc516165023)

[Приложение 1а. Схема разработанной базы данных. 66](#_Toc516165024)

[Приложение 1б. Схема разработанной базы данных. 67](#_Toc516165025)

[Приложение 1в. Схема разработанной базы данных. 68](#_Toc516165026)

[Приложение 2. Код SQL-запроса, создающего триггерную функцию add\_to\_log, заполняющую таблицу logs. 69](#_Toc516165027)

[Приложение 3. Код SQL-запроса, создающего триггерную функцию для проверки топологических правил. 70](#_Toc516165028)

[Приложение 4. Код SQL-запроса для выявления самозахватов. 71](#_Toc516165029)

[Приложение 5. Код SQL-запроса для выделения участков с видом разрешенного использования «Личное подсобное хозяйство». 72](#_Toc516165030)

# Список принятых сокращений

1. КПТ - кадастровый паспорт территории;
2. ГКН - Государственный кадастр недвижимости;
3. ЕГРН - Единый государственный реестр недвижимости;
4. ПКК - Публичная кадастровая карта;
5. ФИАС - Федеральная информационная адресная система;
6. ФНС - Федеральная налоговая служба;
7. БТИ - бюро технической инвентаризации;
8. ОКС - объект капитального строительства;
9. ЗУ - земельный участок;
10. ЛПХ - личное подсобное хозяйство;
11. ИЖС - индивидуальное жилое строительство;
12. ЕЭКО - Единая электронная картографическая основа;
13. ОКАТО - общероссийский классификатор административно-территориального деления;
14. КЛАДР - классификатор адресов России;
15. ОКТМО - общероссийский классификатор территорий муниципальных образований;
16. МСК - местная система координат;
17. ОФП - ортофотоплан;
18. СУБД - система управления базами данных;
19. БД - база данных;
20. XML - eXtensible Markup Language (расширяемый язык разметки);
21. KML - Keyhole Markup Language (язык разметки на основе XML для представления географических данных);
22. OGC - Open Geospatial Consortium (международная некоммерческая организация, ведущая деятельность по разработке стандартов в сфере пространственных данных и сервисов);
23. WKT - Well-known Text (текстовый формат описания геометрических фигур);
24. SQL - Structured Query Language (структурированный язык запросов к базам данных);
25. SRID - Spatial Reference id (идентификатор пространственной привязки);
26. UTF-8 - Unicode Transformation Format, 8-bit (распространенный формат кодирования текста);

# Введение

Компания Геоскан принимала участие в проекте «Цифровая модель типового региона Российской Федерации», связанном с созданием единой электронной картографической основы единого государственного реестра недвижимости на территорию одного из субъектов Российской Федерации в рамках программы Национальной технологической инициативы. Национальная технологическая инициатива (НТИ) - это государственная программа, инициированная правительством Российской Федерации и реализованная при участии Агентства стратегических инициатив (АСИ), которая направлена на поддержание в России перспективных отраслей развития промышленности.

В процессе выполнения проекта выполнялись геодезические работы, съемка местности с беспилотных летательных аппаратов с последующим созданием цифровых ортофотопланов (ОФП), а также определение фактических границ землепользования на основании полученных снимков и ОФП.

Для выполнения геодезических работ были задействованы 90 пунктов государственной геодезической сети, 24 опорные базовые станции, 125 временных базовых станций и проведена съемка более 8 000 контрольных точек. Суммарное время спутниковых ГНСС наблюдений составило более 21 000 часов.

Для получения снимков и ОФП было осуществлено более 4 000 полетов с совокупной длиной маршрутов около 500 000 км. Размер всех полученных снимков составил 54 556 Гб.

Таким образом, материалами проекта являлись:

* Снимки и цифровые ортофотопланы, полученные с помощью беспилотных аэрофотосъемочных комплексов серии Геоскан и последующей обработки в ПО Agisoft Photoscan;
* КПТ - кадастровые планы территорий;
* База данных Федеральной налоговой службы (ФНС);
* Адресная база, составленная на основе данных Федеральной информационной адресной системы (ФИАС), кадастровых планов территорий и полевых сверок, а также данных сетевых компаний (водоканал, энергосбытовые компании).

Одной из задач было определение фактических границ землепользования картографическим методом на основании цифровых ортофотопланов и фотограмметрическим на основании снимков (в сложных случаях), а также выявлении ошибок исходной кадастровой информации.

Изначально работы проводились для одного района области, в качестве пилотного проекта, после успешного выполнения в работу были включены еще 3 района и встал вопрос о масштабировании работ на весь регион. Специфика данного проекта заключалась в большом объеме информации: было получено около 7 млн снимков и предстояло проанализировать более 945 000 земельных участков. Это неоспоримо требовало поиска решений по максимально возможной оптимизации и автоматизации всех этапов работ, так как все данные нуждаются в предварительной обработке и проверки по окончании процесса. А также необходимо было осуществлять контроль над ходом работ. Проводить обработку посредством ручного отбора и проверки нерационально ввиду больших временных затрат и возникновения риска ошибок, связанных со значительной ролью человеческого фактора.

Одним из путей автоматизации является использование системы управлениями базами данных (СУБД), что в целом достаточно логичный выбор при работе с большими объемами информации. Было принято решение осуществить переход на новый формат работы, что повлекло за собой возникновение следующих задач:

1. Изучений возможностей современных СУБД и выбор оптимального варианта;
2. Разработка структуры базы пространственных данных;
3. Достижение максимальной оперативности в выявлении ошибок исходных данных и оптимизация процесса их поиска с помощью запросов на языке SQL;
4. Обеспечение контроля выполнения работ;
5. Поиск способов выявления топологических ошибок, возникающих в процессе векторизации;
6. Организация дополнительных инструментов обеспечения сохранности данных.

В первой главе данной работы описываются основные аспекты организации кадастрового учета Российской Федерации. Такие как нормативные документы, формирующие правовое поле в этой сфере, порядок учета земельных участков и объектов капитального строительства, а также возможности получения данных, хранящихся в информационных базах Федеральной службы государственной регистрации, кадастра и картографии (Росреестр).

Вторая глава посвящена описанию всех ошибок, которые обнаруживаются в кадастровых данных, получаемых из Росреестра и объяснению причин, по которым устранение этих ошибок рациональнее осуществлять внутри базы данных, нежели ручной выверкой.

В третьей главе приведено обоснование выбора в качестве системы управления базами данных PostgreSQL, некоторые особенности работы с базой пространственных данных и описание практической части по разработке и созданию структуры базы данных.

Описание способов поиска ошибок кадастровой информации и применения конечных данных, включая технические аспекты работы с базой данных, а также результаты найденных путей решения поставленных задач на примере одного субъекта Российской Федерации, содержатся в четвертой главе.

# Глава 1. Организация кадастрового учета в Российской Федерации.

## 1.1. Документы, регулирующие правовые отношения в области кадастрового учета.

Основы регулирования земельно-имущественных отношений и особенности, связанные с различными видами недвижимого имущества, в Российской Федерации определяются нормативными документами различных уровней: Конституция Российской Федерации, федеральные законы, указы Президента РФ, постановления Правительства РФ, акты федеральных органов исполнительной власти РФ. Все законодательные и нормативно-правовые акты можно разделить на 2 категории: прямо и косвенно регулирующие кадастровую деятельность РФ. Основными документами являются:

1. Конституция РФ (от 12.12.1993 с учетом поправок, внесенных Законами Российской Федерации о поправках к Конституции Российской Федерации от 30.12.2008 N 6-ФКЗ, от 30.12.2008 N 7-ФКЗ, от 05.02.2014 N 2-ФКЗ, от 21.07.2014 N 11-ФКЗ);
2. Гражданский кодекс РФ (от 30.11.1994 № 51-ФЗ) ;
3. Земельный кодекс РФ (от 25.10.2001 № 136-ФЗ);
4. Лесной кодекс РФ (от 04.12.2006 № 200-ФЗ);
5. Водный кодекс РФ (от 03.06.2006 № 74-ФЗ);
6. Градостроительный кодекс РФ (от 29.12.2004 № 190-ФЗ);
7. Жилищный кодекс РФ (от 29.12.2004 № 188-ФЗ);
8. Федеральный закон №218 «О государственной регистрации недвижимости» (от 01.01.2017);
9. Федеральный закон №221 «О кадастровой деятельности» (от 24.07.2007).

***Конституция РФ***

Конституция - это основной закон страны, который сообщает, что земля и другие природные ресурсы - это основа жизни и деятельности народов, подлежит охране и может находиться в частной, государственной, муниципальной и иных формах собственности (Конституция РФ, ст. 9, 1993). Право частной собственности охраняется законом и каждый гражданин РФ вправе иметь имущество в собственности, владеть, пользоваться и распоряжаться им как единолично, так и совместно с другими лицами (Конституция РФ, ст. 35, 1993). Граждане РФ и их объединения вправе также иметь в частной собственности землю, а условия и порядок пользования землей определяются на основе федерального закона (Конституция РФ, ст. 36, 1993).

***Гражданский кодекс РФ***

Гражданский кодекс регулирует правовое положение участников гражданского оборота, их договорные обязательства и личные имущественные и неимущественные отношения, определяет порядок осуществления права собственности, а также интеллектуальных прав. Особое значение для кадастровой деятельности имеют статьи 130 и 131. В статье 130 установлено, что к недвижимым вещам (недвижимое имущество, недвижимость) относятся земельные участки, участки недр и все, что прочно связано с землей, то есть объекты, перемещение которых без несоразмерного ущерба их назначению невозможно, в том числе здания, сооружения, объекты незавершенного строительства, а также жилые и нежилые помещения и предназначенные для размещения транспортных средств части зданий или сооружений (машино-места), если границы таких помещений, частей зданий или сооружений описаны в установленном законодательством о государственном кадастровом учете порядке. Статья 131 определяет, что право собственности и другие вещные права на недвижимые вещи, ограничения этих прав, их возникновение, переход и прекращение подлежат государственной регистрации в едином государственном реестре органами, осуществляющими государственную регистрацию прав на недвижимость и сделок с ней.

***Земельный кодекс РФ***

Является основным источником земельного права. В земельном кодексе вводится понятие «земельный участок», описываются способы образования земельных участков как объектов кадастровой деятельности. Провозглашается приоритет сохранения особо ценных земель и земель особо охраняемых территорий, а также принцип деления земель по целевому назначению на категории, согласно которому правовой режим земель определяется исходя из их принадлежности к определенной категории и разрешенного использования в соответствии с зонированием территорий и требованиями законодательства (Земельный кодекс РФ, ст. 1, 2001). В статье 7 настоящего кодекса уточняется состав земель в Российской Федерации, приводится перечень категорий. Главы с XIV по XVIII посвящены подробным разъяснениям особенностей осуществления земельных отношений с каждой категорией в отдельности.

Порядок осуществления прав собственности и вещных прав на землю отражены в главах III-V, а порядок возникновения и прекращения прав в V и VII главах.

***Лесной кодекс РФ***

Регулирует отношения, связанные с использованием лесных участков, являющихся недвижимым имуществом. Данный кодекс определяется лесной участок как земельный участок, который расположен в границах лесничеств, лесопарков и образован в соответствии с требованиями земельного законодательства и настоящего кодекса (Лесной кодекс РФ, ст. 7, 2006).

***Водный кодекс РФ***

Устанавливает, что водные объекты находятся в федеральной собственности (принадлежат Российской Федерации), за исключением следующих случаев: пруд, обводненный карьер, расположенные в границах земельного участка, принадлежащего на праве собственности субъекту Российской Федерации, муниципальному образованию, физическому лицу, юридическому лицу, находятся соответственно в собственности субъекта Российской Федерации, муниципального образования, физического лица, юридического лица, если иное не установлено федеральными законами (Водный кодекс РФ, ст. 8, 2006).

***Градостроительный кодекс РФ***

Законодательство о градостроительной деятельности регулирует отношения по территориальному планированию, градостроительному зонированию, планировке территории, архитектурно-строительному проектированию, отношения по строительству объектов капитального строительства, их реконструкции, капитальному ремонту, а также по эксплуатации зданий, сооружений (Градостроительный кодекс РФ, ст. 4, 2004). Также кодекс определяет виды градостроительной документации (Градостроительный кодекс РФ, ст. 9, ст. 26, 2004), в частности, устанавливающие характеристики земельных участков и содержит информацию о видах разрешенного использования земельных участков и объектов капитального строительства (Градостроительный кодекс РФ, ст. 37, 2004).

***Жилищный кодекс РФ***

Объектами жилищных прав являются жилые помещения. Жилым помещением признается изолированное помещение, которое является недвижимым имуществом и пригодно для постоянного проживания граждан (Жилищный кодекс РФ, ст. 15, 2004). Кодекс устанавливает порядок предоставления сведений о жилых помещениях органам кадастрового учета.

***Федеральный закон №218 «О государственной регистрации недвижимости»***

Настоящий Федеральный закон регулирует отношения, возникающие в связи с осуществлением на территории Российской Федерации государственной регистрации прав на недвижимое имущество и сделок с ним, государственного кадастрового учета недвижимого имущества, а также ведением Единого государственного реестра недвижимости и предоставлением предусмотренных настоящим Федеральным законом сведений, содержащихся в Едином государственном реестре недвижимости (ФЗ-218, ст. 1, 2017). Определяет орган, осуществляющий кадастровый учет на территории РФ (ФЗ-218, ст. 3, 2017), геодезическую и картографическую основу Государственного реестра недвижимости (ФЗ-2018, ст. 6, 2017).

В главах 4-5 описывается порядок внесения сведения в ГРН, а в главе 8 порядок предоставления и плата за сведения, содержащиеся в ГРН.

Статья 69 устанавливает понятие ранее учтенного объекта недвижимости.

***Федеральный закон №221 «О кадастровой деятельности»***

Предметом регулирования закона являются отношения, возникающие в связи с осуществлением кадастровой деятельности, деятельности саморегулируемых организаций кадастровых инженеров, национального объединения саморегулируемых организаций кадастровых инженеров.

Многие статьи утратили силу после вступления в силу ФЗ №218 01.01.2017г.

## 1.2. Государственный кадастр недвижимости (ГКН) и единый государственный реестр недвижимости (ЕГРН), публичная кадастровая карта (ПКК), единая электронная картографическая основа (ЕЭКО).

*Государственный кадастр недвижимости* (ГКН) - система, включающая в себя сведения обо всех объектах недвижимого имущества, которые были поставлены на кадастровый учет (земельные участки, квартиры, дома, здания любого назначения). А также сведения о прохождении Государственной границы Российской Федерации, о границах между субъектами Российской Федерации, муниципальных образований, населенных пунктов, о территориальных зонах и зонах с особыми условиями использования территорий (ФЗ-221, 2007).

Основное назначение ГКН - организация оборота прав, зарегистрированных в Едином государственном реестре прав на недвижимое имущество и сделок с ним. Ведение ГКН осуществляется Федеральной службой государственной регистрации, кадастра и картографии, подведомственным учреждением Росреестра.

С 1 января 2017 года сведения, содержавшиеся в Государственном кадастре недвижимости, а также Едином государственном реестре прав на недвижимости (ЕГРП), вошли в *Единый государственный реестр недвижимости* (ЕГРН). Это государственный информационный ресурс, содержащий достоверные данные об объектах недвижимости, находящихся на территории РФ. Государственная регистрация права в ЕГРН является единственным доказательством существования зарегистрированного права. Данные ГКН и ЕГРН являются общедоступными и могут быть получены заинтересованным лицом в письменном или электронном виде после внесения установленной государством пошлины. Заказать их можно через сайт госуслуг Росреестра (https://rosreestr.ru/). Существует несколько видов кадастровых данных, которые можно получить, используя этот информационный ресурс:

1. Выписка из ЕГРН. Документ, содержащий наиболее полные сведения об объекте недвижимости, на основании которого подтверждается право собственности гражданина на недвижимое имущество. В ЕГРН содержатся следующие сведения:
   1. Уникальный кадастровый номер;
   2. Данные о собственнике объекта недвижимости или о распределении долей между собственниками, если у объекта несколько владельцев;
   3. Основания возникновения правомочий владельцев, изменение и переход прав или прекращение правомочий;
   4. Площадь объекта, координаты границ;
   5. Адрес;
   6. Геометрия;
   7. Категория земель;
   8. Вид разрешенного использования;
   9. Кадастровая стоимость;
   10. Дата постановки на учёт.

Существует несколько разновидностей форм выписок:

* Об объекте недвижимости (основных характеристиках и зарегистрированных на него прав);
* О переходе прав на объект недвижимости;
* О правомочиях отдельного лица на имевшиеся (имеющиеся) у него объекты недвижимости;
* О признании правообладателя ограниченно дееспособным или недееспособным.

Выписка из ЕГРН необходима при подготовке документов о межевании, приватизации жилья, судебных спорах и т.д. Предоставляется на платной основе.

1. Кадастровый паспорт объекта недвижимости. Это информация об объекте недвижимости. В отличии от кадастровой выписки, не содержит пространственных данных объекта и является обязательным документом при регистрации права на недвижимое имущество.

Может быть выдан на квартиру или на земельный участок, в зависимости от чего будет содержать различные сведения. Кадастровый паспорт на квартиру имеет следующие пункты:

* 1. Предмет назначения жилого или нежилого помещения;
  2. Адрес, этажность или указание этажа, на котором находится помещение;
  3. Тип помещения и площадь;
  4. Инвентаризационный и кадастровый номера;
  5. Дата проведения технической экспертизы;
  6. Орган БТИ, выдавший документ.

Кадастровый паспорт на земельный участок, в свою очередь, включает информацию об общих данных на земельный участок, схему с планировкой земельного участка и раздел об ограничении использования данного имущества.

1. Выписка о кадастровой стоимости объекта недвижимости. Особая форма выписки, содержит только кадастровый номер объекта и его кадастровую стоимость. Необходима для расчета величины налога при оформлении кредита и наследства и при выкупе объекта недвижимости. Предоставляется бесплатно.
2. Кадастровый план территории. Это план кадастрового квартала, состоящий из графической и семантической частей и содержащий информацию обо всех объектах недвижимости, находящихся в пределах кадастрового квартала (здания, сооружения, инженерные коммуникации и др.). Требуется при выполнении различного рода кадастровых работ. Предоставляется платно. Подробнее о сведениях, содержащихся в КПТ, будет рассказано в следующих разделах.

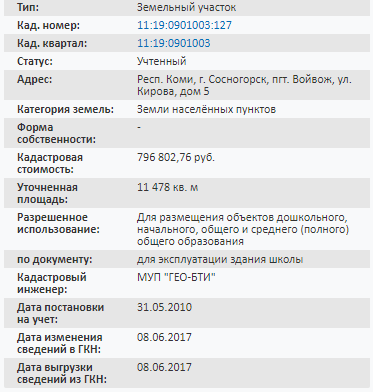
*Публичная кадастровая карта* (ПКК) - это всероссийский справочно-информационный интернет-ресурс, предназначенный для получения гражданами РФ информации об объектах недвижимости, стоящих на кадастровом учете (https://pkk5.rosreestr.ru/). Позволяет осуществлять поиск по кадастровым номерам ОКС, земельных участков, кварталов, районов и т.д. А также поиск по адресу. Пример характеристик земельного участка, предоставляемых на ПКК, представлен на Рисунке 1. Дополнительно можно посмотреть план объекта недвижимости или провести поиск объектов, находящихся в его границах.

Общедоступные кадастровые сведения, содержащиеся на ПКК - Рисунок 2.

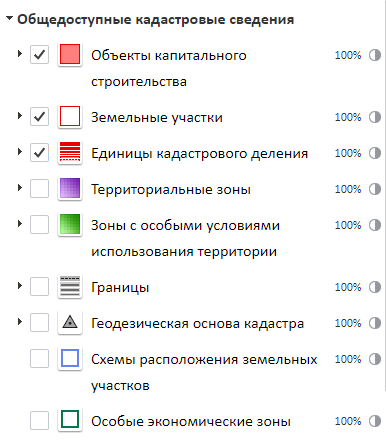
В качестве дополнительных сведений можно добавить на карту границы лесничеств и лесопарков, красные линии, а также нанести какие-либо объекты самостоятельно, используя инструменты редактирования или загрузив их из файла формата kml.

В качестве подложки для кадастровых данных можно использовать космические снимки, единую электронную картографическую основу или одну из следующих тематических карт:

* форма собственности на земельных участках;
* кадастровая стоимость ЗУ;
* кадастровая стоимость ЗУ за 1 кв. м;
* категории земель;
* разрешенное использование ЗУ;
* срок давности выгрузки сведений из ГКН.



*Рисунок 1. Пример характеристик земельного участка на ПКК.*

****

*Рисунок 2. Общедоступные кадастровые сведения, содержащиеся на ПКК.*

*Единая электронная картографическая основа* (ЕЭКО) федерального, регионального, муниципального назначения состоит из слоев цифровых государственных топографических карт или планов в векторном формате либо, в случае их отсутствия, растровых геокодированных материалов дистанционного зондирования Земли, а также метаданных и создается в масштабах 1:2 000, 1:5 000, 1:10 000, 1:25 000, 1:50 000, 1:100 000, 1:200 000, 1:500 000, 1:1 000 000 (Приказ Минэкономразвития России №467, 2008). Создана для обеспечения органов государственной власти, органов местного самоуправления, физических и юридических лиц пространственными данными. Об этом сообщается в 20-й статье Федерального закона от 30.12.2015 №431 (ред. от 03.07.2016) «О геодезии, картографии и пространственных данных и о внесении изменений в отдельные законодательные акты Российской Федерации». В целях обновления сведений единой электронной картографической основы используются, в том числе сведения, содержащиеся в Едином государственном реестре недвижимости. Перечень сведений, содержащихся в ЕЭКО, определяется органом исполнительной власти, уполномоченным осуществлять функции по выработке государственной политики и нормативно-правовому регулированию в сфере геодезии и картографии.

## 1.3. Кадастровые планы территорий (КПТ). Особенности составления и содержимое. Структура XML-документа.

Как уже было отмечено, кадастровый план территории - это тематический план кадастрового квартала, включающий все объекты недвижимого имущества, стоящие на кадастровом учете и находящиеся в пределах данного квартала. В КПТ можно найти не только атрибутивную информацию об объектах, но и описания границ участков и ОКС. По проекту КПТ являлись основным источником кадастровой информации

КПТ можно получить как в письменном, так и в электронном виде. В электронном виде они поставляются в формате XML-документа при заказе через портал Росреестра.

XML - расширяемый язык разметки. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: не существует строгого набора дескрипторов (тегов), они определяются разработчиком в зависимости от предметной области. Необходимо лишь соблюдать синтаксические правила. Таким образом, язык оказывается удобен как для обработки и создания программами, так и для создания и чтения человеком (Спецификация Extensible Markup Language (XML) 1.0, 2000).

Схемы XML-документов утверждаются приказами Федеральной службы государственной регистрации, кадастра и картографии. На данный момент актуальной является 10-я версия схемы, утвержденная приказом Росреестра от 18.08.2016 №390.

КПТ-схема определяет наименования элементов, обязательность или необязательность наличия атрибутов, формат полей (текст, число, дата, булево выражение, пустое выражение, вложенный элемент) и их максимальную длину, а также карту значений для атрибутов с ограниченным набором возможных значений.

В общем смысле состоит из двух частей:

1. Описание кадастрового квартала и входящих в него объектов:
   1. Земельные участки;
   2. Здания, сооружения, объекты незавершенного строительства;
   3. Границы, объекты землеустройства, единицы кадастрового деления;
2. Сведения об удостоверении документа:
   1. Наименование органа кадастрового учета;
   2. Дата;
   3. Номер документа;
   4. Должностное лицо:
      1. Должность удостоверяющего;
      2. ФИО удостоверяющего:
         1. Фамилия;
         2. Имя;
         3. Отчество.

Далее подробнее рассмотрим атрибуты, определенные для объектов недвижимого имущества по отдельности.

Земельные участки:

1. Площадь земельного участка:
   1. Значение площади;
   2. Единицы измерения;
   3. Погрешность измерения;
2. Наименование участка (по справочнику «Вид земельного участка»)
3. Уточнение местоположения (заполняется в отсутствии адреса):
   1. Наименование ориентира;
   2. Расстояние до ориентира;
   3. Направление на ориентир;
   4. Расположение ориентира относительно границ участка;
4. Адрес:
   1. ОКАТО;
   2. КЛАДР;
   3. ОКТМО;
   4. Почтовый индекс;
   5. Код региона;
   6. Район;
   7. Муниципальное образование;
   8. Городской район;
   9. Сельсовет;
   10. Населенный пункт;
   11. Улица;
   12. Дом;
   13. Корпус;
   14. Строение;
   15. Квартира;
   16. Иное;
   17. Неформализованное описание;
5. Категория земель;
6. Разрешенное использование участка:
   1. Вид разрешенного использования по классификатору «Классификатор видов использования земель»;
   2. Вид использования участка по документу;
   3. Вид разрешенного использования земельного участка в соответствии с классификатором, утвержденным приказом Минэкономразвития России от 01.09.2014 № 540;
7. Кадастровый номер земельного участка;
8. Контур многоконтурного земельного участка;
9. Кадастровая стоимость:
   1. Значение кадастровой стоимости;
   2. Единицы измерения;
10. Статус записи о ЗУ по справочнику «Статус объекта недвижимости»;
11. Дата постановки на учет;
12. Описание местоположения границ:
    1. Сведения о системе координат;
    2. Координаты точек границ:
       1. Координата X;
       2. Координата Y.

Здания, сооружения, объекты незавершенного строительства:

1. Здание:
   1. Вид объекта недвижимости (здание);
   2. Назначение здания;
   3. Площадь в квадратных метрах;
   4. Адрес (с таким же подразделением как у ЗУ);
   5. Кадастровая стоимость (с таким же подразделением как у ЗУ);
   6. Кадастровый номер;
   7. Описание местоположения границ (с таким же подразделением как у ЗУ).
2. Сооружение:
   1. Вид объекта недвижимости (сооружение);
   2. Назначение сооружения;
   3. Основные характеристики сооружения:
      1. Тип характеристики по справочнику «Тип основного параметра»;
      2. Значение характеристики;
   4. Адрес (с таким же подразделением как у ЗУ);
   5. Кадастровая стоимость (с таким же подразделением как у ЗУ);
   6. Кадастровый номер;
   7. Описание местоположения границ (с таким же подразделением как у ЗУ);
3. Объект незавершенного строительства:
   1. Вид объекта недвижимости (ОНС);
   2. Проектируемое назначение объекта незавершенного строительства;
   3. Основные характеристики сооружения:
      1. Тип характеристики по справочнику «Тип основного параметра»;
      2. Значение характеристики;
   4. Адрес (с таким же подразделением как у ЗУ);
   5. Кадастровая стоимость (с таким же подразделением как у ЗУ);
   6. Кадастровый номер;
   7. Описание местоположения границ (с таким же подразделением как у ЗУ).

Границы, объекты землеустройства, единицы кадастрового деления:

1. Местоположение квартала:
   1. Описание местоположения границ (с таким же подразделением как у ЗУ);
   2. Документы-основания:
      1. Код документа;
      2. Наименование документа;
      3. Серия документа;
      4. Номер документа;
      5. Организация, выдавшая документ;
      6. Особые отметки;
2. Границы между субъектами РФ, границы населенных пунктов, муниципальных образований, расположенных в кадастровом квартале:
   1. Наименование объекта землеустройства;
   2. Учетный номер;
   3. Описание местоположения границ (с таким же подразделением как у ЗУ);
   4. Документы-основания (см. местоположение квартала);
   5. Наименование смежного субъекта РФ (для границ между субъектами РФ);
   6. Наименование муниципального образования;
   7. Наименование населенного пункта;
3. Границы территориальных зон, зон с особыми условиями использования территорий:
   1. Наименование объекта землеустройства;
   2. Учетный номер;
   3. Описание местоположения границ (с таким же подразделением как у ЗУ);
   4. Документы-основания (см. местоположение квартала);
   5. Территориальная зона:
      1. Вид разрешенного использования;
      2. Разрешенное использование в соответствии с классификатором, утвержденным приказом Минэкономразвития России от 01.09.2014 № 540;
      3. Разрешенное использование в соответствии с ранее использовавшимся классификатором;
   6. Зона с особыми условиями использования территорий:
      1. Содержание ограничений использования объектов недвижимости.

В процессе работы над проектом также пришлось столкнуться с КПТ, составленными по еще не утвержденной схеме 2017 года. Одним из главных отличий было новое поле «Адрес в соответствии с ФИАС», что оказалось очень полезным нововведением, т.к. до этого для выверки адресной базы и поиска ранее учтенных участков приходилось корректировать некоторые написания наименований населенных пунктов и улиц. А также в новую схему добавлены несколько новых разновидностей поля «Вид разрешенного использования», а именно:

* Вид разрешенного использования по классификатору (числовой код);
* Вид разрешенного использования по справочнику (текстовый формат);
* Вид разрешенного использования по старому классификатору (числовой код);
* Вид разрешенного использования по старому справочнику (текстовый формат).

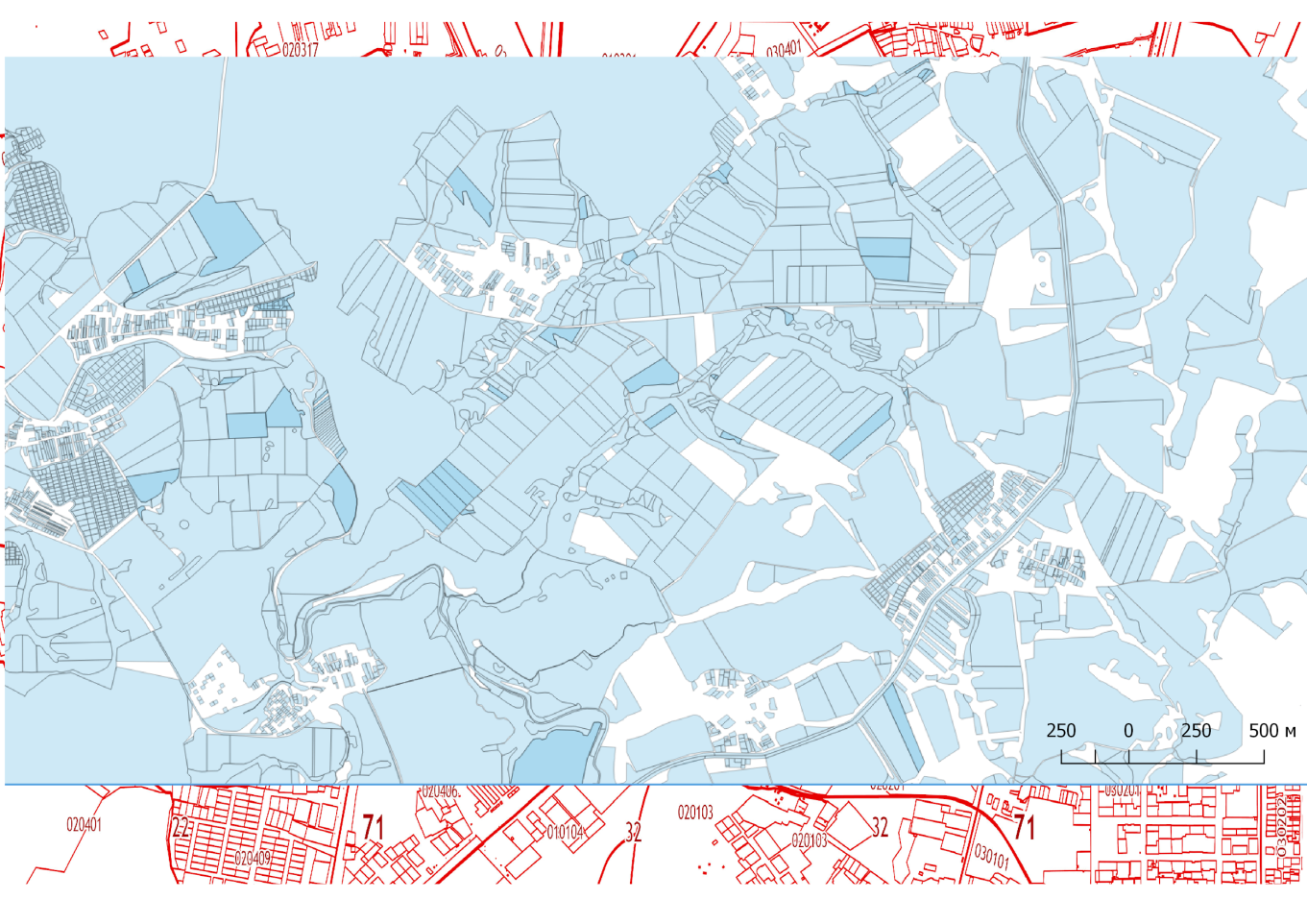
Единицы измерения были добавлены непосредственно в названия полей, содержащих площади. Также возникли небольшие трудности при сопоставлении переименованных атрибутов.

# Глава 2. Анализ кадастровых данных.

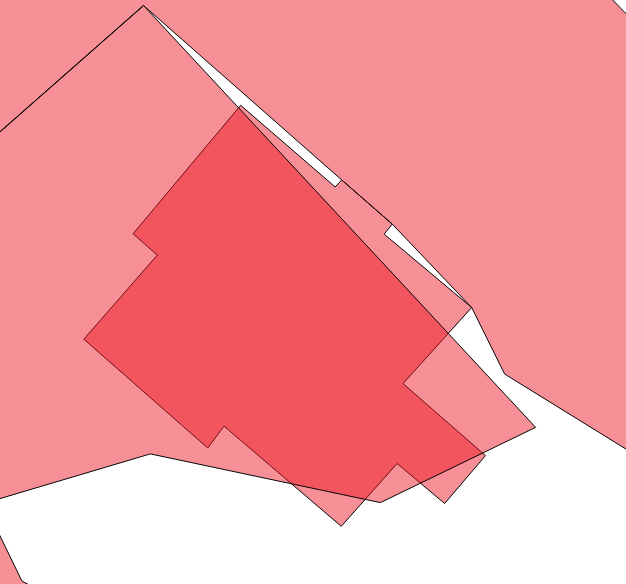
## 2.1. Примеры основных ошибок кадастровой информации.

Изначально, целью проекта была разработка структуры базы данных для определения фактических границ землепользования земельных участков и объектов капитального строительства. В качестве исходных данных использовались кадастровые планы территорий (КПТ). Однако, в процессе работы выяснилось, что исходные кадастровые данные включают в себя некоторые виды технических ошибок, которые затрудняют рабочий процесс на всех его этапах. Основные виды ошибок:

1. *Дубли геометрии*. При формировании базы данных Росреестра осуществляется проверка уникальности кадастрового номера, но не геометрии. Иногда один и тот же участок стоит на учете под разными кадастровыми номерами. Такие ситуации возникают из-за того, что до 2013 года государственная регистрация прав на недвижимость проводилась не в единой базе данных, а сначала в органах местного самоуправления, а затем в органах государственной регистрации. Пунктом 3 статьи 70 Закона №218 предусмотрено, что орган регистрации прав в соответствии с установленными правилами ведения ЕГРН снимает с государственного кадастрового учета земельный участок, учтенный в установленном законодательством Российской Федерации порядке до 1 марта 2008 года. Участки с дублирующейся геометрией в рамках проекта выделяются в отдельную таблицу. На Рисунке 3 показана схема исходных кадастровых данных, на которой темно-синим цветом выделены дублирующиеся участки.
2. *Пересечения земельных участков*. Возникают вследствие многих причин: недостаточная точность геодезических измерений; определение координат характерных точек границ нового участка без учета границ смежного; линейно-угловые ошибки, возникшие при измерении земельного участка. В кадастровой практике существуют различные способы урегулирования подобных конфликтов. В рамках проекта определяются фактические границы землепользования картографическим способом по цифровым ортофотопланам. Участки, между которыми существуют спорные границы, выделяются отдельно. На Рисунке 4 представлен пример пересекающихся земельных участков, на котором темно-красным цветом показана область пересечения.

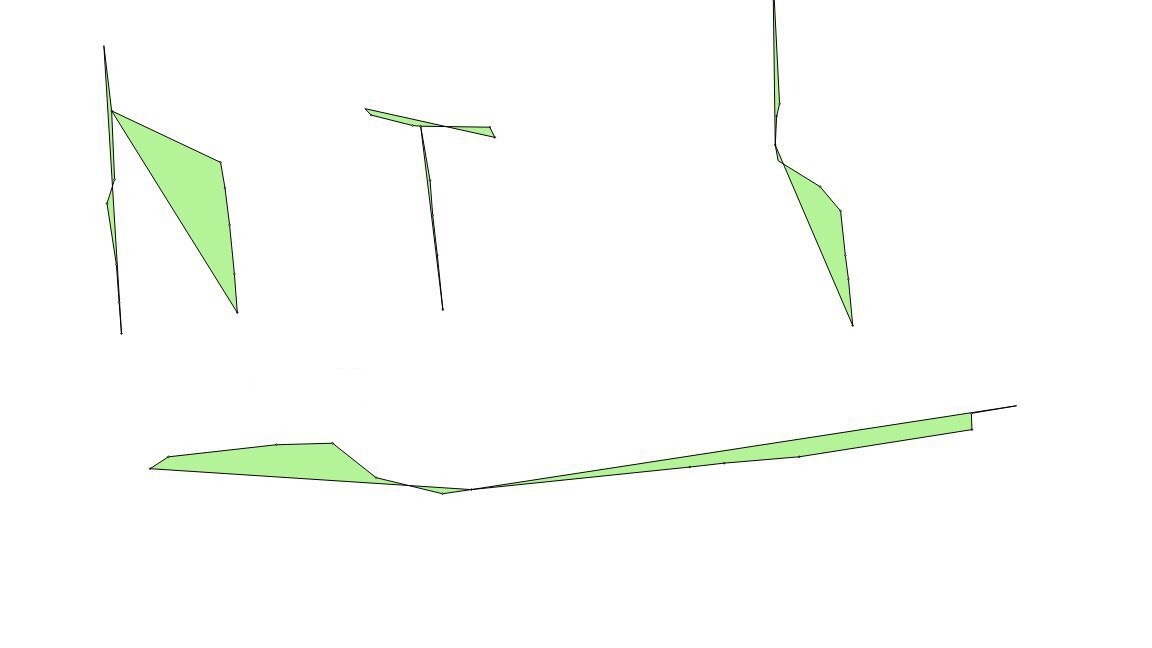


*Рисунок 3. Схема исходных кадастровых данных с дублирующимися участками.*



*Рисунок 4. Пример пересечения земельных участков.*

1. *Самопересечение объекта.* Возникает при конвертации в векторный формат из-за нарушений в последовательности пар координат в исходных КПТ. Такие ошибки исправляются в процессе векторизации. Примеры самопересекающихся объектов представлены на Рисунке 5.



*Рисунок 5. Примеры самопересечений объектов.*

1. *«Отлетевшие» объекты*. Так обозначаются объекты, чье местонахождение не соответствует истинному положению участка или здания. Такие ошибки могут возникнуть из-за неверного определения или внесения в ГКН координат характерных точек границ объекта. А также в результате допущенной технической ошибки при пересчете координат из местной или условной системы в установленную законом. При наличии границ кварталов можно определить приблизительное местоположение объекта, уточнить которое можно с помощью сопоставления адресной базы и кадастровой информации объекта недвижимости.
2. *Ошибки категории земель.* Попадание земель населенных пунктов в категорию сельскохозяйственных земель и наоборот. Это один из видов ошибок семантической информации, который можно обнаружить по косвенным признакам, информации из других атрибутов.
3. *Опечатки.* Очень распространенный вид ошибок в атрибутах, связанных с адресом участка, а также в поле «Вид использования участка по документу». Вызваны, вероятно, «ручным» заполнением данных полей. Исправить адрес участка или здания можно, сопоставив кадастровые данные с написанием в базе ФИАС. Что касается поля «Вид использования участка по документу», можно сравнить значение с классификатором.

## 2.2. Преимущества подхода с использованием системы управления базами пространственных данных (СУБД).

Одно из основных преимуществ работы с использованием СУБД - возможность распределенной одновременной работы над проектом, в котором задействовано большое количество сотрудников. Альтернативная схема организации работ, которая использовалась в самом начале работы над проектом - обмен файлами.

Сотрудников, работавших над определением фактических границ землепользования, было 3-4 человека. Каждый работал со своим набором файлов. После завершения векторизации файлы передавались одному человеку, который занимался сборкой всех частей работы в один файл, параллельно проверяя стыковку на границах работ разных сотрудников, исправляя возникающие наложения и т.д. При увеличении числа сотрудников, такая схема становилась трудоемкой и нецелесообразной. Еще одной проблемой являлась подготовка исходных данных. Перечисленные выше ошибки требовали поиска, выделения и, по возможности, исправления. Однако, результаты выполнения этих задач «вручную», не всегда удовлетворяли предъявляемым требованиям или отнимали слишком много ресурсов. Также большую роль играл человеческий фактор.

При увеличении объема работ неизбежно растет количество сотрудников. Для повышения производительности необходимо максимально оптимизировать и автоматизировать весь процесс достижения конечного результата. Хранение информации должно быть систематизировано и единообразно. Копирование, передача и сборка файлов также могут нарушить версионность и внести лишнюю путаницу.

Логично доверить хранение данных системе управления базами данных, т.к. это позволит:

1. *Организовать распределенную работу.* При использовании клиент-серверной структуры работы с БД появляется возможность подключения и одновременного редактирования различными пользователями одних и тех же таблиц. При этом все задействованные сотрудники видят изменения друг друга в онлайн-режиме. Соответственно пропадает необходимость в сборке и проверке корректной стыковки объектов, созданных разными пользователями.
2. *Безопасность информации.* Доступ к базе данных осуществляется по логину и паролю. Пользователей можно наделять правами или же запрещать проводить им какие-либо действия с данными. Правила доступа можно установить на разных уровнях: для каждой конкретной таблицы, отдельной схемы или базы данных целиком. Можно разрешить только просматривать данные, без возможности внесения изменений, или сделать таблицу невидимой для одного пользователя или группы пользователей.
3. *Обработка атрибутивной информации.* Работа через СУБД позволяет упростить все операции с семантическими данными, начиная от выборок по значению какого-либо атрибута и заканчивая от генерирования производных полей, объединения разных таблиц по общим атрибутам и создания новых.
4. *Геометрическая обработка.* Как говорилось выше, исходные кадастровые данные содержат различного рода ошибки, такие как дубли и пересечения земельных участков. Настольные ГИС позволяют выявить их как топологическую ошибку. Но работая через СУБД, мы можем не только выделить подобные участки, содержащие ошибки, но и очистить таблицу от лишней информации, задав критерий отбора этих участков.
5. *Централизованная установка единой системы координат.* Система координат - это основополагающий фактор любых пространственных данных. Кадастровые данные в РФ поставляются, как правило, в местных системах координат регионов. Параметров МСК нет в библиотеках, которые используют крупные настольные ГИС (например, библиотека proj4[[1]](#footnote-1)). Следовательно, вносить эти параметры необходимо самостоятельно. Из-за некорректно установленных параметров объекты могут сместиться относительно их реального положения. При работе через СУБД такие ситуации невозможны, т.к. параметры СК устанавливаются централизованно и не зависят от ПО-клиента.
6. *Оптимизация пространства.* СУБД позволяет создавать представления - таблицы, оперативно создаваемые на основе SQL-запроса к другим таблицам. Если какая-либо таблица является производной от других (подсчет статистики, расчет площадей, выделение самозахватов и т.д.), нам не обязательно создавать статичные таблица и затем их обновлять при обновлении исходных данных. Мы имеем возможность вынести эту информацию в представление, которое во-первых не будет занимать физически пространства на сервера, а во-вторых будет оперативно обновляться вслед за основными данными.

# Глава 3. Применение СУБД для хранения, анализа и обработки кадастровых данных.

## 3.1. Обзор и сравнение существующих систем управления базами пространственных данных.

Система управления базами данных - это комплекс языковых и программных средств, предназначенных для создания, администрирования и использования базы данных (C.J.Date, 2004). Пространственные базы данных отличаются от обычных тем, что предоставляют возможность хранения пространственных объектов и дополнительный функционал для работы с такими объектами. Это осуществляется, как правило, за счет расширений под каждую конкретную СУБД.

Перед тем как приступить к реализации идеи, необходимо было определиться, какая СУБД будет использоваться. Существуют как коммерческие, так и свободные решения для работы с пространственными данными для различных СУБД. Несмотря на то, что все крупные системы управления базами пространственных данных стараются соответствовать стандартам OGC (Open Geospatial Consortium) и OGIS (Open GIS Consortium), между ними все-таки есть различия. Рассмотрим некоторые наиболее популярные варианты.

***MySQL Spatial***

MySQL - это свободная реляционная система управления базами данных с открытым кодом, разработку и поддержку которой осуществляет корпорация Oracle. Для работы с пространственными данными никакие дополнительные расширения не требуются. Весь функционал включен в стандартный набор. Поддерживаемые WKT-форматы данных:

* Point;
* LineString;
* Polygon;
* Multipoint;
* MiltiLineString;
* MultiPolygon;
* GeometryCollection.

Все в соответствии со стандартом OGC 06-103r4 Simple Feature Access, за исключением того, что в MySQL поддерживаются только двумерные пространственные данные.

При создании таблиц в MySQL существует возможность выбора схемы хранения данных (storage engines). Пространственные типы поддерживаются в [MyISAM](https://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html), [InnoDB](https://dev.mysql.com/doc/refman/5.7/en/innodb-storage-engine.html), [NDB](https://dev.mysql.com/doc/refman/5.7/en/mysql-cluster.html), и [ARCHIVE](https://dev.mysql.com/doc/refman/5.7/en/archive-storage-engine.html). Возможность построения пространственных индексов есть только в MyISAM и InnoDB. Без пространственных индексов заметно увеличивается время обработки запросов к базе данных, поэтому имеет смысл выбирать из этих двух схем хранения. Из преимуществ MyISAM - высокая производительность при операциях типа INSERT и SELECT, из недостатков - блокировка на уровне таблиц, в то время как в InnoDB - на уровне строк. Так называемая, row-level блокировка позволяет нескольким пользователям параллельно работать с одной таблицей, т.к. при редактировании СУБД блокирует только редактируемую строку, а не всю таблицу (MySQL 5.7 Reference Manual, 2018).

Что касается пространственных функций, в данной СУБД реализованы все основные методы для работы с пространственными данными, за исключением функций, связанных с z, m - координатами. Координата Z - это координата, обозначающая высоту (глубину), а M - это, так называемая, «измеренная» координата (от англ. measurement), в качестве которой могут выступать численные параметры объекта, например, температура или давление воздуха в точке.

***Oracle Spatial and Graph***

Oracle Database (Oracle RDBMS) - проприетарная объектно-реляционная система управления базами данных компании Oracle. Spatial and Graph - отдельно лицензируемый компонент данной СУБД, позволяющий работать с пространственными данными. Позволяет хранить как векторные, так и растровые данные. Последний пункт представляет особый интерес, так как в базе данных хранится не просто картинка, а полноценные пространственные растровые данные, для которых существует специальный тип объектов SDO\_RASTER. Метаданные описываются в виде XML-схемы (GeoRaster Metadata XML Schema). В первую очередь, это информация о геопривязке растра и о системе координат, в которой эта привязка осуществлялась. Также для данных дистанционного зондирования есть возможность хранить все каналы вместе, как это происходит, например, при использовании формата hdf.

Более того, в Spatial and Graph мы можем не просто хранить свои растровые данные, но также обрабатывать при помощи функций растровой алгебры, в том числе проводить классификацию по растровым изображениям и строить поверхности уклонов, экспозиции по цифровым моделям рельефа.

Что касается векторной информации, в соответствии со спецификацией OGIS поддерживаются следующие геометрические примитивы:

* Unknown geometry;
* Point;
* Line or Curve;
* Polygon or Surface;
* Collection;
* MultiPoint;
* MultiLine or MultiCurve;
* MultiPlygon or MultiSurface;
* Solid$
* MultiSolid.

Объекты в модели данных этой СУБД могут быть 2-х, 3-х или 4-х - мерными. Если с двумя или тремя координатами все понятно, то четвертая координата (в других СУБД или ГИС ее еще называют M-dimension, от англ. measurement) - это «измеренная» координата. В качестве нее может служить какая-то численная характеристика объекта, например, высота или температура в данной точке.

Стоит отметить, что существует отдельный класс объектов SDO\_TIN, описывающий TIN-поверхности, а также SDO\_PC для облака точек. Расширение поддерживает создание пространственных индексов.

Oracle Spatial and Graph поддерживает все описанные стандартами функции для анализа и обработки пространственных данных (Oracle Spatial and Graph Developer’s Guide, 2017).

***MS SQL Server 2017***

Это еще одна коммерческая система управления реляционными базами данных, разработанная компанией Microsoft. Поддерживает два пространственных типа данных:

1. Geometry - данные в плоской системе координат;
2. Geography - данные в системе координат для сферической Земли.

Первый тип данных соответствует стандарту OGC и поддерживает следующие виды объектов:

* Point;
* MultiPoint;
* LineString;
* CircularString;
* MultiLineString;
* CompoundCurve;
* Polygon;
* MultiPolygon;
* CurvePolygon;
* GeometryCollection.

В большинстве случаев поведение типа geography не отличается от geometry, но есть исключения. Например, при измерениях. При использовании geometry измерения проводятся в тех же единицах, в которых заданы координаты объектов. Чаще всего, это метры, однако, если координаты объектов необходимо хранить в градусах, а площадь или длина все-таки нужны в метрах, целесообразно пользоваться типом geography. Также второй тип данных имеет ряд некоторых ограничений:

1. Не допускается хранение объектов, превышающих размеры полушария;
2. Если результат выполнения функции, для которой указываются два аргумента (пересечение, буфер и т.д.), превышает размеры полушария, функция вернет NULL.

В MS SQL Server имеется специальный объект FullGlobe, который является разновидностью типа Polygon и описывает весь земной шар, имеет площадь, но не имеет границ и вершин.

В СУБД реализована поддержка пространственных индексов для обоих типов пространственных данных, а также всех основных функций для работы с пространственными данными (Документация MS SQL Server 2017).

***PostgreSQL PostGIS***

Свободная объектно-реляционная система управления базами данных, практически не уступающая по мощности различным проприетарным аналогам. PostGIS - расширение для работы с пространственными данными. В рамках типа пространственных данных geometry PostGIS сейчас поддерживает все форматы данных, описанные в стандарте OGC:

* Point;
* Line, LineString, LinearRing;
* Curve;
* Polygon, Triangle;
* Multipoint;
* MiltiLineString;
* MultiPolygon;
* MultiCurve;
* GeometryCollection;
* Surface;
* PolyhedralSurface;
* MultiSurface.

Для всех перечисленных типов реализована поддержка не только «плоских» объектов, но также существует возможность работы с Z- и M-координатами.

Также PostGIS позволяет хранить данные в различных системах координат, используя библиотеку proj4 для преобразования систем координат. Пользователь может, как выбрать нужную из таблицы spatial\_ref\_sys (поддержка стандартов OpenGIS), так и добавить туда свою, если у него имеются ее параметры в формате proj4.

СУБД PostgreSQL по умолчанию поддерживает 3 вида индексации: B-tree, R-tree и GiST. PostGIS для реализации пространственных индексов использует R-tree поверх GiST.

В пространственной БД под управлением PostgreSQL (PostGIS) могут храниться пространственные данные типа geography с координатами формата широта/долгота, и пользователь может проводить измерения на эллипсоиде с указанием его параметров.

В PostGIS реализованы все функции для работы с пространственными данными, описанные в стандарте OGC.

Если пользователю необходимо хранить растровые данные, у PostGIS есть особый тип данных raster, в котором хранятся пространственные растровые данные, а также несколько сопутствующих типов, содержащих результаты работы растровых функций. Исходные растровые данные могут быть форматов jpeg, tiff, png, а также цифровые модели рельефа. Могут включать 1 или более каналов. Тип данных raster хранит информацию о геопривязке растровых изображений, если таковая имеется.

С данными типа raster в PostGIS можно работать с помощью функций растровой алгебры. Среди прочих, строить поверхности уклонов, экспозиции, создавать векторные представления типа geometry на основе значений пикселей. Все функции работы с растровыми изображениями в PostGIS реализованы при помощи библиотеки GDAL (PostGIS 2.4.5dev Manual, 2018).

## 3.2. СУБД PostgreSQL - возможности и особенности.

В результате сравнения и анализа существующих систем управления базами данных, позволяющих хранить и обрабатывать пространственные данные, выбор пал на СУБД PostgreSQL с расширением PostGIS, т.к. упор делался на использование в работе над проектом открытого программного обеспечения. Но, даже отбросив этот факт, функционал данной СУБД полностью удовлетворяет задачам и целям проекта.

Далее будут рассмотрены все основные аспекты создания базы данных под управлением PostgreSQL, а также вопросы, неизбежно возникающие и не всегда очевидно разрешаемые.

***Начало работы***

Структура любой базы данных в PostgreSQL организована следующим образом:

1. Сервер:
   1. Пользователи или группы пользователей
   2. База данных:
      1. Расширения
      2. Схемы:
         1. Таблицы:
            1. Колонки
            2. Триггеры
         2. Триггерные функции
         3. Представления:
            1. Колонки

Безусловно, это усеченное представление, в действительности структура базы данных значительно шире и включает в себя большее количество элементов. Однако, для реализации наших целей этого пока что достаточно.

Таким образом, каждая БД может включать в себя одну или несколько схем. По стандарту всегда существует схема public. Если пользователь не указывает принадлежность таблицы к какой-либо схеме явно (schema.table), СУБД будет искать/создавать эту таблицу в public.   
 Схемы используются для логического объединения данных в кластеры, либо для их разделения под использование конкретными пользователями. В разных схемах могут существовать объекты с одинаковыми именами. Перемещение таблиц из одной схемы в другую происходит легко, с помощью одного SQL-запроса, в отличии от перемещения между разными базами данных: не используя дополнительных изменений конфигураций, в рамках одного подключения к серверу пользователь может обращаться только к одной базе данных.

Как же правильно организовать структуру хранения, создавать несколько баз данных в рамках одного сервера или несколько схем в рамках одной базы данных? К сожалению, однозначного ответа на этот вопрос нет, и у того, и у другого метода есть свои преимущества и недостатки. При наличии нескольких баз данных, необходимо для каждой отдельно добавлять расширение PostGIS, а значит указывать свои параметры СК (если нужной нет в списке), создавать триггерные функции. Что такое триггерные функции, где и как указываются параметры СК, будет описано ниже. На данном этапе необходимо понимать, что все эти действия происходят на уровне базы данных. Во втором случае, при разделении данных по схемам, база данных может оказаться слишком громоздкой, что ухудшит инфраструктуру хранения данных. А также как следствие, может замедляться процесс доступа к данным из-за их большого объема.

***Установка расширения, системы координат***

Чтобы иметь возможность полноценно работать с пространственными данными в созданной БД необходимо загрузить в нее расширение PostGIS. Это делается посредством выражения CREATE EXTENSION, если пользователь работает через командную строку или поле запросов в каком-либо клиентском ПО.

После загрузки расширения, в схеме public автоматически появляется таблица spatial\_ref\_sys, которая содержит данные о системах координат. Она содержит следующие данные:

* srid - порядковый номер системы координат;
* auth\_name - наименование СК;
* auth\_srid - общепринятый код СК, EPSG (совпадает с колонкой srid);
* srtext - описание параметров СК в формате WKT;
* proj4text - описание параметров СК в формате proj4.

Как говорилось выше, для преобразования систем координат, PostGIS использует библиотеку proj4, соответственно, поддерживает все СК, реализованные в данной библиотеке. Но, если пользователю необходимо хранить данные в своей системе координат (например, местная СК субъекта РФ), он может добавить новую строку в spatial\_ref\_sys, указав, в первую очередь, атрибуты SRID и proj4text.

Чтобы присвоить или обновить систему координат какой-либо таблицы, в PostGIS предусмотрены функции:

ST\_SetSRID(geometry geom, integer srid),  
UpdateGeometrySRID(<**schema**>, <**table**>, <geometryColumn>, integer srid).

Необходимо понимать, что при обновлении системы координат таблицы, хранимые координаты не пересчитываются с учетом новых параметров. Обновление происходит лишь на уровне метаданных. Для пересчета координат существует функция ST\_Transform.

Также после установки расширения, появятся 2 новых представления: geometry\_columns и geography\_columns, для каждого типа данных соответственно. В них хранятся данные о размерности, системе координат, типе геометрии и т.д. таблиц, содержащих пространственные данные. Так как это представления, изменять эти данные, как таблицу spatial\_ref\_sys нельзя, они сугубо информационные. Чтобы изменить размерность данных или тип геометрии, необходимо воспользоваться специальными функциями, например ST\_Force2D (создает двумерное представление данных) или рассмотренные выше функции для установки и изменения системы координат. Помимо этих двух представлений для полей геометрии, расширение добавит еще два для растровых данных: raster\_columns и raster\_overviews, которые также содержат метаданные, только уже о растровых данных. В проекте они нам не понадобятся, однако о такой возможности знать стоит.

***Таблицы, последовательности, ключи, индексы, представления***

Добавить таблицу с пространственными данными в нашу базу данных мы можем двумя способами:

1. Импортировать уже имеющиеся данные;
2. Создать путем SQL-запроса.

Первый способ будет подробнее рассмотрен в следующей главе, с готовыми слоями иметь дело гораздо проще. О втором же способе поговорим сейчас. Создание таблиц с геометрией происходит в 2 этапа:

1. Создаем обыкновенную таблицу, с помощью выражения CREATE TABLE, с полями, соответствующими нужным атрибутам;
2. Добавляем колонку для хранения геометрии с помощью функции AddGeometryColumn( <**table**>, <**column**>, <srid>, <**type**>, <dimension>). В аргументах функции сообщаем имя таблицы, имя колонки, номер системы координат, тип хранимой геометрии и размерность геометрии.

Далее для созданной таблицы необходимо создать последовательность (sequence). Последовательность - это объект базы данных, генерирующий уникальную последовательность чисел, другими словами, автоинкрементное поле. Новые значения генерируются с помощью функции nextval(regclass). Не путать с индексом. При импорте уже существующей таблицы, последовательность создается автоматически. При создании пространственной таблицы путем запроса, ее необходимо создать самому. Это также можно сделать двумя способами:

1. Сначала создать последовательность с необходимыми параметрами: шаг счетчика (по умолчанию 1), минимальные и максимальные значения (по умолчания NULL) и др. После чего создать таблицу, в том числе с полем id, у которого будет определено значением по умолчанию функция nextval(regclass), и назначить это поле первичным ключом.

**CREATE** SEQUENCE <id\_sequence>  
**INCREMENT** **BY** **1**  
**NO** **MAXVALUE**

**NO** **MINVALUE**

**CACHE** **1**;  
**CREATE** **TABLE** <**table**> (  
id integer **DEFAULT** nextval('<id\_sequence>') **NOT** **NULL**,  
 ... );

1. При создании таблицы добавить поле id с типом SERIAL/BIGSERIAL и назначить его первичным ключом. Дело в том, что создание поля с этим типом данных инициирует автоматическое создание последовательности.

Важное замечание: если у таблицы, поддерживающей хранение пространственных данных, отсутствует последовательность, ее будет невозможно редактировать через геоинформационные клиенты.

Что означает «назначить первичным ключом»? Первичный ключ - это уникальный идентификатор объекта в пределах одной таблицы. Хотя сам PostgreSQL позволяет создать таблицу без первичного ключа, по условию соблюдения 2-й нормальной формы его наличие обязательно. В качестве ПК может выступать один из атрибутов (или комбинация), если он поддерживает условие уникальности и не содержит значений типа NULL. Однако, гораздо чаще создается инкрементное поле, которое и выступает в роли первичного ключа. У этого есть вполне объективная причина: осуществлять операции с простыми числами гораздо проще и быстрее для СУБД.

Помимо первичного ключа, существуют внешние ключи, связывающие разные таблицы между собой и обеспечивающие ссылочную целостность. Это означает, что значения столбца (или группы столбцов) одной таблицы не могут принимать значения, которые не содержатся в другой таблице. Соответственно существуют ссылающаяся (referencing) и ссылочная (referenced) таблицы. Внешних ключей может быть несколько для одной таблицы. Главные преимущества использования внешних ключей - соблюдение связанности информации и оптимизация ее хранения.

Следующим этапом для успешной работы с таблицей, в которой хранятся пространственные данные, должно стать создание пространственного индекса. Индексы в PostgreSQL - это объекты базы данных, предназначенные для ускорения доступа к данным. Опять же, при импорте через какие-то сторонние источники, ПО, как правило, предлагает пользователю сразу создать пространственный индекс. Для чистой таблицы, созданной в базе данных, это можно сделать с помощью с помощью запроса CREATE INDEX. Пространственные индексы в PostGIS реализованы при помощи метода GiST (Generalized Search Tree). Не углубляясь в технические подробности реализации построения индекса, можно сказать, что данный индекс создается на основе взаимного расположения объектов геометрии, путем последовательного разбиения пространства на более мелкие части до тех пор, пока в одну часть не будет попадать лишь один объект. Без пространственных индексов даже простейший запрос будет выполняться долго, т.к. СУБД придется сканировать все строки таблицы подряд, пока не будет найдена строка, удовлетворяющая заданному условию.

Последний логический пункт данного раздела - представления. Представления в PostgreSQL - это таблицы, создаваемые путем SQL-запроса, «на лету». Они не занимают физического места на диске и не имеют никаких собственных данных. Позволяют создавать производную информацию на основе базовых таблиц (одной или несколько). Основное преимущество: данные в представлении изменяются вслед за изменениями в базовых таблицах. При желании результат запроса можно сохранить в базовую таблицу. Недостатки возникают при попытке изменения имени или типа колонки в представлении, требуется пересоздавать представление, что может вызывать некоторые неудобства.

***Пользователи, группы пользователей, разграничение доступа***

Допустим, у нас есть некоторое количество пользователей базы данных. Они используют разные таблицы в разном объеме. В целях обеспечения безопасности и сохранения данных, имеет смысл ограничить доступ пользователей к различным схемам и таблицам. Изначально в базе существует один пользователь - создатель базы данных, он определяется как «суперпользователь» и наделен всеми возможными правами доступа и управления данными, а также правом создания новых пользователей. Для этого предусмотрено выражение CREATE ROLE. Доступ к базе данных осуществляется по логину и паролю, которые задаются при создании, так называемой, роли. Параметры, которые можно указать непосредственно при создании роли:

1. *Суперпользователь* (SUPERUSER/ NOSUPERUSER)*.* Мы можем создать еще одну роль в базе, для которой не будет существовать ограничений. Суперпользователя может создать только другой суперпользователь. Следует понимать, чем больше таких пользователей в базе, тем больше угрозы для данных;
2. *Право создавать новые базы данных* (CREATEDB/NOCREATEDB);
3. *Право создавать новые роли* (CREATEROLE/NOCREATEROLE);
4. *Наследование прав* (INHERIT/NOINHERIT)*.* Подразумевается наследование прав группы ролей, если создаваемая роль является частью какой-либо группы;
5. *Обеспечение входа на сервер* (LOGIN/NOLOGIN)*.* Если значение этого атрибута не установлено дополнительно, он принимается равным NOLOGIN. В таком случае создается не пользователь в обычном понимании, а «группа пользователей», которую мы можем использовать для администрирования базы данных, но заходить на сервер под ее именем и паролем нельзя. Существует команда CREATE USER, для которой этот параметр по умолчанию равен LOGIN;
6. *Лимит подключений* (CONNECTION LIMIT)*.* Определяет количество параллельных подключений для роли. Значение «-1» снимает все ограничения.
7. *Отношения* (IN ROLE, ROLE)*.* Можно сделать роль членом какой-либо группы или добавить новых пользователей в создаваемую роль;
8. *Срок действия пароля* (VALID UNTIL);

Существуют некоторые другие параметры (например, роль репликации), но на данном этапе перечисленных более чем достаточно.

Раздача права на определенные действия с объектами производятся на уровне объектов. Это, так называемые, привилегии. Они бывают следующих видов, в зависимости от типа объекта:

1. SELECT. Доступ на чтение;
2. INSERT. Добавление новых строк в таблицу;
3. UPDATE. Обновление данных;
4. DELETE. Удаление строк из таблицы;
5. REFERENCES. Создание внешнего ключа;
6. TRUNCATE. Опустошение таблицы;
7. TRIGGER. Создание триггеров для таблицы;
8. CREATE. Создание схем, новых объектов в схеме, таблиц, последовательностей, индексов и т.д.;
9. CONNECT. Подключение к базе данных;
10. TEMPORARY. Создание временных таблиц;
11. EXECUTE. Выполнение функции;
12. USAGE. Имеет много вариаций в зависимости от объекта. Важно: применительно к последовательностям дает право на вызов функции nextval;

Пример набора возможных привилегий для таблиц: SELECT, INSERT, UPDATE, DELETE, REFERENCES, TRUNCATE, TRIGGER.

Пример набора возможных привилегий для последовательностей: SELECT, UPDATE, USAGE.

Таким образом, мы определяем, какими привилегиями наделить каждого конкретного пользователя или группу пользователей, в зависимости от того, как они используют эти данные. Тем не менее, существует ряд рекомендаций, которые необходимо соблюсти для корректной работы с пространственной базой данных:

1. Для возможности чтения параметров систем координат и корректного отображения пространственных данных необходим доступ на SELECT для таблицы spatial\_ref\_sys. Похожая ситуация с таблицей layer\_styles, которая хранит стили слоев для отображения в геоинформационной системе;
2. Также пользователь должен иметь привилегию SELECT для представления geometry\_columns, о котором мы говорили ранее, чтобы СУБД владела информацией, в какой именно системе координат хранятся данные, объекты какого типа геометрии и какой размерности находятся в таблице. В противном случае, таблицы будут недоступны;
3. Привилегии SELECT и USAGE для последовательностей, иначе в геоинформационной системе слои можно будет открыть, но не будет возможности редактировать. Происходит это потому, что последовательность обеспечивает генерирование первичного ключа с помощью функции nextval.

***Триггеры, триггерные функции***

Команда CREATE TRIGGER в PostgreSQL определяет операцию, которая будет выполняться при добавлении, обновлении или удалении строк в таблице, которой принадлежит триггер. Триггер существует на основе триггерных функций, то есть инициирует их выполнение. В свою очередь, синтаксис создания триггерной функции следующий:

**CREATE** **FUNCTION** <**function**>() **RETURNS** <**trigger**>**AS** $$  
**DECLARE**  
<variables>;  
**BEGIN**  
<comands>;  
**END**;   
**LANGUAGE** plpgsql;

Существуют специальные переменные, с помощью которых функция работает с данными таблиц, к которым обращается. С помощью различных функций и триггеров, мы можем вводить дополнительные проверки корректности данных (например, проверка топологии) или вести таблицы учета изменений, хранящие данные обо всех действиях над объектами, совершенными разными пользователями в базе данных.

## 3.3. Методика создания базы пространственных данных на примере кадастровых данных.

Исходными кадастровыми данными для проекта являлись кадастровые паспорта территорий (КПТ), о составе которых рассказывалось в предыдущих главах. Так как КПТ поставляются в формате XML, их необходимо конвертировать в привычные нам векторные файлы. В качестве такого векторного формата был выбран формат ESRI shapefile по нескольким причинам: во-первых, этот формат позволяет уйти от совместного хранения различных видов геометрий в одном файле, а во-вторых, для него имеется готовый набор утилит для последующей загрузки в БД. Конвертация проводилась с помощью ПО «Спутник Кадастр», разработка и поддержка которого осуществляется ГК Геоскан. Формат XML поддерживает хранение всех типов геометрий, а также объектов, не имеющих данных о геометрии, в одном файле. При переходе к формату shp автоматически формируются 4 части:

1. Shp-файл с точечными объектами;
2. Shp-файл с линейными объектами;
3. Shp-файл с полигональными объектами;
4. Csv-таблица с объектами без геометрии.

Точечные объекты в рамках проекта не использовались. В файл с линейными объектами попадали границы районов, населенных пунктов, кварталов. Полигональный слой состоит из земельных участков(различных категорий) и объектов капитального строительства(зданий и сооружений). В csv-файл попадают объект всех категорий, у которых в исходном XML-файле не содержалось координат границ.

Также исходными данными являлся точечный shp-файл с адресной базой, которая составлялась на основании данных администраций районов, полевых выверок и корректировалась в соответствии с базой данных Федеральной информационной адресной системы (ФИАС). Для каждой точки 3 атрибута: населенный пункт, улица и дом.

Дополнительно, из Федеральной налоговой службы, заказывались данные о налогах за земельные участки, которые содержали кадастровые номера участков, сумма исчисленного налога, сумма налога, предъявленного к уплате и т.д.

Помимо этого, в работе использовались ортофотопланы и снимки. Они разбиты по квадратам для удобства. Так как объем этих материалов был очень большим, для них также была организована БД на основе PostgreSQL, к которой можно было подключаться, тем самым получая доступ к снимкам и ортофотопланам. С помощью специального модуля для QGIS создается shp-файл с разграфкой тайлов.

Итак, чтобы начать работу, в первую очередь, необходимо подключиться к серверу и создать базу данных. Для этого существует команда CREATE DATABASE. Дополнительно можно определить владельца создаваемой БД, кодировку таблиц, количество подключений. По умолчанию, владельцем является пользователь, выполняющий запрос, кодировка для всех таблиц - UTF-8, количество подключений не ограничено. Таким образом, значение параметров можно оставить дефолтным.

Однако, с количеством подключений может возникнуть проблема. Несмотря на то, что количество подключений к базе не ограничено, существует ограничение на количество подключений к серверу. Оно прописывается в файле конфигурации postgresql.conf в корневой папке PostgreSQL. Учитывая то, что под «подключением» подразумевается «запрос» к базе данных, один пользователь - не равно одно подключение. И при более-менее достаточном количестве пользователей БД может возникнуть следующая ошибка:

Arval **SQLException**: FATAL: sorry, too many clients already.

Это означает, что в данный момент превышено количество подключений к серверу. Чтобы это исправить, необходимо в postgresql.conf изменить строку max\_connections, которая по умолчанию равна 100. Также было бы полезно изменить значение параметра shared\_buffers, который определяет максимальный объем оперативной памяти, отведенный СУБД. Однако с этим параметром следует быть осторожными и рассчитывать его, исходя из мощностей ПК и его загруженности.

Отслеживать подключения можно с помощью таблицы pg\_stat\_activity. Она не отображается в общей схеме БД, но ее можно получить по запросу. В экстренных случаях подключения можно сбрасывать принудительно. Например, при неаккуратном обращении с подключениями возникает большое количество незавершенных, но неактивных сеансов. Тогда можно выполнить следующий запрос:

**SELECT** pg\_terminate\_backend(pid)   
 **FROM** pg\_stat\_activity   
 **WHERE** pid <> pg\_backend\_pid()  
 **AND** datname = 'database\_name' ;

Функция администрирования pg\_terminate\_backend() - завершает обслуживающие процессы пользователей, а pg\_backend\_pid() - определяет процесс, обслуживающий текущий сеанс, мы исключаем его из числа завершаемых процессов, чтобы не прервать выполнение запроса.

После создания самой базы, необходимо подключить расширение PostGIS, создать необходимые схемы и импортировать наши исходные данные. О способах импорта будет подробно рассказано в следующей главе, поэтому сразу перейдем к следующим пунктам. Теперь в базе имеются 5 таблиц со следующими названиями:

* polygons\_source - таблица, содержащая полигональные объекты, полученные из КПТ;
* lines\_source - таблица, содержащая линейные объекты, полученные из КПТ;
* without\_geom - таблица, содержащая объекты без геометрии;
* address\_source - адресная база с точками
* taxes\_source - данные ФНС;
* tiles - разграфка ортофотопланов.

На этом работа с исходными данными еще не закончена. Периодически в КПТ одного муниципального района попадают участки из соседних или даже из соседних регионов. Справиться с этим можно двумя путями: в первом случае, выделив уникальные значения и осуществив фильтрацию по полю «Район», а во втором по полю «Код региона». Например:

**SELECT** **DISTINCT** "Код региона" **FROM** polygons\_source;  
 **DELETE** \* **FROM** polygons\_source **WHERE** "Код региона"<>**71**;

Следующим пунктом исходные данные необходимо очистить от, так называемых, легальных дублей. Это полностью идентичные объекты, в том числе с одинаковой геометрией, которые были продублированы несколько раз в разных КПТ. Это не ошибка, обычно это объекты, относящиеся к нулевому кварталу. Нулевой кадастровый квартал - это условное обозначение, которое используется для участков, располагающихся в некоторых кадастровых кварталах. Так как нам не имеет значения, какой именно из двух объектов оставить для работы, их можно удалить следующим запросом:

**DELETE** **FROM** <**table\_name**> **USING**  
(**SELECT** t1.gid

**FROM**  
(**SELECT** **COUNT**(geom) **AS** count\_, **MAX**(gid) **AS** gid, geom, "Кадастровый номер земельного участка", "Дата постановки на учет"

**FROM** test

**GROUP** **BY** geom, "Кадастровый номер земельного участка", "Дата постановки на учет" )t1  
**WHERE** count\_>**1**)t2  
**WHERE** <**table\_name**>.gid=t2.gid;

Для удобства и корректности, из polygons\_source необходимо выделить в отдельную таблицу объекты капитального строительства. Это можно сделать, руководствуясь следующим принципом: согласно структуре КПТ, есть два поля с кадастровыми номерами: «Кадастровый номер» и «Кадастровый номер земельного участка». Первое предназначено для хранения кадастровых номеров ОКС. Следовательно, мы можем выделить все строки таблицы, в которых это поле ненулевое и перенести в таблицу buildings, в которой будем хранить здания и сооружения:

**SELECT** \* **INTO** buildings

**FROM** polygons\_source

**WHERE** "Кадастровый номер" **IS** **NOT** **NULL**;  
**DELETE** **FROM** polygons\_source

**WHERE** "Кадастровый номер" **IS** **NOT** **NULL**;

При использовании команды SELECT...INTO...FROM новая таблица создается автоматически, нам остается лишь определить для нее первичный и внешний ключи, права доступа.

Продолжаем работу с таблицей polygons\_source, в которой теперь остались лишь земельные участки. Однако, по условиям проекта, необходимо было определить фактические границы землепользования только для земель двух категория: населенных пунктов и сельскохозяйственного использования. Согласно приказу Росреестра от 12.10.2011 N П/389 (ред. от 21.10.2014) в Российской Федерации существует 8 категорий земель со специальными кодами. Они представлены в Таблице 1.

|  |  |  |
| --- | --- | --- |
| Номер | Классификационный код | Значение |
| 3.1 | 003001000000 | Земли сельскохозяйственного назначения |
| 3.2 | 003002000000 | Земли населенных пунктов |
| 3.3 | 003003000000 | Земли промышленности, энергетики, транспорта, связи, радиовещания, телевидения, информатики, земли для обеспечения космической деятельности, земли обороны, безопасности и земли иного специального назначения |
| 3.4 | 003004000000 | Земли особо охраняемых территорий и объектов |
| 3.5 | 003005000000 | Земли лесного фонда |
| 3.6 | 003006000000 | Земли водного фонда |
| 3.7 | 003007000000 | Земли запаса |
| 3.8 | 003008000000 | Категория не установлена |

*Таблица 1. Классификатор категорий земель.*

Выделим нужные категории в два различных представления с помощью следующих SQL-запросов:

1. **CREATE** **VIEW** cadastre\_np **AS**  
   **SELECT** \* **FROM** polygons\_source **WHERE** "Категория земель"='003002000000' **OR** "Категория земель"='003008000000';
2. **CREATE** **VIEW** cadastre\_agro **AS**  
   **SELECT** \* **FROM** polygons\_source **WHERE** "Категория земель"='003001000000';

В первом случае мы добавили еще код для неустановленной категории, т.к. опытным путем было выяснено, что в нее часто попадают земли населенных пунктов, которые необходимо учитывать.

Существует еще один нюанс: земли населенных пунктов - это, в том числе, ЛЭП, газопроводы, водопроводы, автомобильные дороги, железные дороги и т.д. Но по проекту нам необходимы лишь земли, используемые для личного подсобного хозяйства и индивидуального жилого строительства. Определить, для чего именно используется участок, можно по полям «Вид разрешенного использования» и «Вид использования по документу». В первом случае, поле заполнено кодами в соответствии с классификатором видов разрешенного использования, а второе является текстовым полем с описанием вида использования земельного участка. Как правило, второе более достоверно отражает действительность, хотя запись не унифицирована, что вызывает некоторые сложности. Число уникальных значений по этому полю может достигать нескольких тысяч, включая не только семантически разные виды, но и опечатки. Вариаций написания нужных нам видов использования (личное подсобное хозяйство) гораздо больше, чем тех, что следует исключить, поэтому отфильтруем созданное представление cadastre\_np по ключевым словам, используя нечеткие соответствия:

**CREATE** **VIEW** cadastre\_np **AS**  
**SELECT** \* **FROM** polygons\_source

**WHERE** ("Категория земель"='003002000000' **OR** "Категория земель"='003008000000')   
**AND** ("Вид использования по документу" **NOT** **LIKE** '%ЛЭП%' **OR** "Вид использования по документу" **NOT** **LIKE** '%газопровод%' **OR** "Вид использования по документу" **NOT** **LIKE** '%водопровод%' **OR** "Вид использования по документу" **NOT** **LIKE** '%дорога%' **OR** "Вид использования по документу" **NOT** **LIKE** '%улица%');

Не исключено, что в некоторых районах подобные ограничения придется вводить и для представления cadastre\_agro, т.к. объекты коммунальной инфраструктуры нередко попадают и в категорию земель сельскохозяйственного назначения.

Из таблицы lines\_source нужно извлечь границы кварталов в quarters, границы населенных пунктов в towns и границу самого района в boundary. Тип границы указывается в поле «Вид объекта реестра границ», по значению которого мы можем выделить линии, соответствующие границам кварталов, границы населенных пунктов и т.д.

Таблицы without\_geom и address\_source оставляем без изменений.

Следующим шагом необходимо создать таблицы, в которых будет вестись векторизация. Схема векторизации следующая:

* zu\_cad - таблица, содержащая скорректированные по ОФП границы участков, которые уже стоят на кадастровом учете и имеются в представлении cadastre\_np с исходными данными;
* zu - таблица, в которой отрисовываются границы участков, не стоящих на кадастровом учете, но видных на ОФП;
* zu\_agro\_1 - участки, принадлежащие категории сельскохозяйственных земель, которые используются по назначению. Как стоящие на кадастровом учете, так и отрисованные по фактическим границам;
* zu\_agro\_2 - участки, принадлежащие категории сельскохозяйственных земель, но используемые для каких-либо иных целей или заброшенные. Как стоящие на кадастровом учете, так и нет;
* ru - ранее учтенные участки
* zu\_cad\_oks - объекты капитального строительства, стоящие на кадастровом учете;
* zu\_oks - объекты капитального строительства, не стоящие на кадастровом учете.

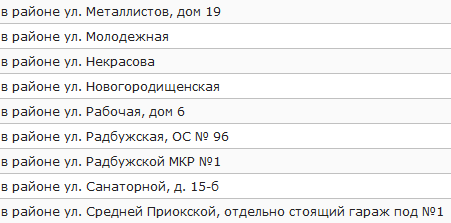
В таблицах zu\_cad, zu\_agro\_1, zu\_agro\_2 и zu\_cad\_oks содержатся земельные участки, стоящие на кадастровом учете, следовательно, они должны содержать некоторые поля, идентичные полям таблиц с исходными данными и хранящие информацию о земельных участках. А именно:

* Кадастровый номер (Тип поля: text);
* Контур (Тип поля: text);
* Неформализованное описание (Тип поля: text);
* Вид разрешенного использования (Тип поля: integer);
* Вид использования участка/ОКС по документу (Тип поля: text);
* Дата постановки на учет (Тип поля: text);
* Площадь земельного участка/ОКС (Тип поля: double);

Помимо приведенного списка, должно присутствовать поле uid с уникальным идентификатором объекта в пределах таблицы, а также gid с глобальным идентификатором объекта, ссылающимся на таблицу polygons\_source с исходными данными, по которому можно восстановить прочие поля в случае необходимости. А также поле geom, содержащее геометрию объектов. Для таблицы zu\_agro\_2 дополнительное поле state типа text, с указанием характеристики угодий, используемых не по назначению: деревья и кустарники, многолетние насаждения, луговая растительность или нецелевое использование (застройка).

Для таблицы zu и zu\_oks достаточно полей gid и geom.

Что касается таблицы ru: она содержит ранее учтенные участки. Ранее учтенные участки - это земельные участки, стоящие на кадастровом учете, но не имеющие информации о границах и, соответственно, геометрии. Единственной геопривязкой, которая у них имеется, является адрес. Для этого нам и нужна адресная база. С помощью специального ПО «Спутник Кадастр», разработанного в компании Геоскан, сопоставляются адреса участков содержащихся в таблице without\_geom и точек из таблицы address\_source. При нахождении совпадений, точкам из адресной базы присваивается кадастровый номер и вся атрибутивная информация земельного участка, после чего точки пересекаются с участками, отрисованными в слое zu и присваивается их геометрия. В результате создаются 9 таблиц с разной степенью достоверности, которые требуют проверки и являются исходными слоями для таблицы ru. Следует отметить, что в исходных КПТ очень много адресов, использовать которые не представляется возможным, т.к. они обозначают не точное местоположение объекта недвижимости, а лишь ориентир. Пример таких адресов представлен на рисунке 6.



*Рис.6. Пример адресов участков из исходных КПТ.*

Таким образом, для ранее учтенных участков требуются поля:

* gid;
* Кадастровый номер земельного участка;
* Контур;
* Площадь земельного участка;
* Неформализованное описание;
* Категория земель;
* Дата постановки на учет;
* Кадастровая стоимость;
* Вид разрешенного использования;
* Вид использования участка по документу.

Общая схема разработанной базы данных представлена в Приложениях 1а-в. Синим цветом обозначены таблицы, содержащие исходные данные, зеленым цветом - представления, фиолетовым - таблицы, в которых велась векторизация и серым - для административного использования.

***Использование триггерных функций и триггеров***

При работе с большими объемами информации, возникает необходимость поиска путей обеспечения сохранности данных и защиты их от случайностей и влияния человеческого фактора. Самый простой пример такого влияния: случайное удаление данных.

С этим можно бороться периодическим созданием резервных копий, как всей базы целиком, так и выгрузкой отдельных слоев в векторные файлы. Однако при использовании таких методов, можно пропустить «критическую» ошибку и потерять определенное количество работы, проделанной большим количеством сотрудников. Поэтому для всех таблиц, в которых велась векторизация, была создана таблица учета изменений, в которую записывалось каждое изменение, вносимое в таблицы. Таблицы учета изменений имеют четкую структуру и содержат следующие поля:

* object\_gid - идентификатор измененного объекта;
* action - вид действия: insert(добавление), update(обновление), delete(удаление);
* user\_name - логин пользователя, внесшего изменение;
* time - время изменения;
* geom - геометрия объекта, над которым произведено изменение;
* table\_name - название таблицы.

Эти таблицы заполняются автоматически с помощью триггерной функции и триггера. В триггерной функции add\_to\_log прописано, какими именно значениями заполнить поля таблицы, а триггер инициирует вызов этой функции при каких-либо действия с исходной таблицей. Код запроса, создающего триггерную функцию, приводится в Приложении 2.

TG\_TABLE\_NAME - это специальная переменная, обозначающая название таблицы, триггер которой сработал.

Таким образом, в базе данных появляется новая таблица logs.

Похожим образом, мы можем осуществлять проверку топологии: при внесении изменений в таблицы, проверять новые объекты на наличие пересечений с другими слоями и с другими объектами того же слоя. В случае наличия недопустимых пересечений, отклонять транзакцию. Соответственно, изменения не будут сохранены, пока ошибки пересечений не будут устранены. Конечный вид функции приводится в Приложении 3.

***Создание ролей, доступ к таблицам***

Изначально в базе данных существует одна администраторская роль, от имени которой была создана база данных. Она является суперюзером и наделена всеми возможными правами. Далее выделялось еще две группы ролей: специалистов, занимающихся определением фактических границ землепользования (Users) и специалистов, ведущих проект и занимающихся проверкой работы (Staff). Соответственно, в первую очередь мы создаем две этих роли, которые позже будут выступать в роли групп:

**CREATE** **ROLE** "Users" **WITH**  
 NOLOGIN  
 NOSUPERUSER  
 **NOCREATEDB**  
 NOCREATEROLE  
 INHERIT  
 NOREPLICATION  
**CONNECTION** **LIMIT** -**1**;

Далее каждого нового создаваемого пользователя, с логином и паролем, определяем к соответствующей группе:

**CREATE** **USER** user\_name **WITH**  
 LOGIN  
 NOSUPERUSER  
 **NOCREATEDB**  
 NOCREATEROLE  
 INHERIT  
 NOREPLICATION  
 **CONNECTION** **LIMIT** -**1**;  
**GRANT** "Users" **TO** user\_name;

Теперь необходимо для каждой таблицы, представления и последовательности установить границы доступа к ней различных групп пользователей.

Во-первых, для таблиц, в которых осуществляется векторизация (zu\_cad, zu, zu\_agro\_1, zu\_agro\_2, ru), разрешаем доступ обеих групп на выбор, добавление, удаление и обновление. Для соответствующих им последовательностей также открываем доступ на выбор и использование.

Во-вторых, обеим группам пользователей необходимо доступ на внесение изменений в таблицы логов, т.к. действия, совершаемые пользователями в основных таблицах, инициируют создание новых записей в таблицах с логами. Следовательно, достаточно прав SELECT и INSERT.

В-третьих, для представлений с исходными данными, группе Users достаточно лишь права на просмотр SELECT. А группе Staff также право на добавление данных INSERT, но уже не в представлениях, а в исходных таблицах, т.к. может возникнуть ситуация, когда исходные данные поступают частями и добавляются в уже созданную таблицу по мере поступления.

Также, ранее оговаривалось, что для корректного отображения систем координат и типов геометрии, пользователям необходимо иметь доступы к представлению geometry\_columns и таблице spatial\_ref\_sys.

***Присвоение системы координат***

При установке расширения PostGIS, в основной схеме (по умолчанию - public) создается таблица spatial\_ref\_sys, в которой содержатся параметры более чем 3000 систем координат. Однако, часто кадастровые данные хранятся в местных системах координат регионов. Удобно, что данная таблица редактируется как любая другая. И, если у нас есть описание СК в формате библиотеки proj4, мы можем добавить новую запись в таблицу и использовать ее для наших данных.

В PostGIS предусмотрено несколько способов установки или изменения системы координат данных:

1. Указать SRID при создании колонки геометрии:

AddGeometryColumn(  
 <**schema\_name**>,  
 <**table\_name**>,  
 <**column\_name**>,  
 <srid>,  
 <**type**>,  
 <dimension>  
);

1. С помощью функции ST\_SetSRID:

**ALTER** **TABLE** **table\_name**  
 **ALTER** **COLUMN** geom **TYPE** geometry(geometry\_type,SRID)   
 **USING** ST\_SetSRID(geom,SRID);

1. С помощью функции UpdateGeometrySRID:

**SELECT** UpdateGeometrySRID(<**schema\_name**>, <**table\_name**>, <geom\_column\_name>, SRID).

Если необходимо не просто присвоить систему координат, но трансформировать координаты, существует функция ST\_Transform.

# Глава 4. Подготовка и анализ данных.

## 4.1. Конвертация данных в формат ESRI shapefile. Способы импорта в базу данных.

Как говорилось выше, кадастровые паспорта территорий поставляются в формате .xml. Несомненно, PostgreSQL имеет функции для работы с такими данными, но особенность КПТ в том, что в них содержится информации о геометрии объектов. Если точнее, это просто координаты узлов объекта под соответствующими тегами. Например:

- <spatials\_elements>

- <spatial\_element>

- <ordinates>

- <ordinate>

<x>479430.69</x>

<y>2260670.74</y>

<ord\_nmb>1</ord\_nmb>

<num\_geopoint>н 350</num\_geopoint>

<geopoint\_zacrep>Закрепление отсутствует</geopoint\_zacrep>

<delta\_geopoint>0.5</delta\_geopoint>

</ordinate>

Для успешной работы необходима конвертация этих координат в более привычный формат .shp. В рамках проекта, это осуществлялось в ПО «Спутник Кадастр» компании Геоскан, в котором поддерживается разбор xml-схем КПТ различной актуальности с приведением их к единому виду для удобства объединения кадастровых данных.

Далее появляется задача импорта полученных слоев в базу данных. Это можно сделать тремя способами:

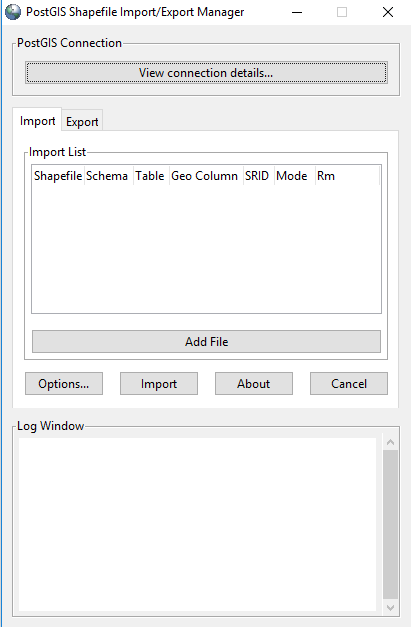
1. *С помощью утилиты shp2pgsql через командную строку.* Это стандартная утилита, имеющая много полезных ключей, позволяющих корректно определить параметры будущей таблицы в базе данных. Основные:
   1. -s - позволяет указать идентификатор системы координат;
   2. -W - установить кодировку таблицы;
   3. -a - добавить данные в уже существующую таблицу;
   4. -I - создание пространственного индекса для колонки с геометрией;
   5. -S - создать простые геометрии вместо мультигеометрий.

Пример команды:

shp2pgsql -I -S -s **4326** <PATH\_TO\_SHP> <**schema\_name**>.<**table\_name**> | psql -h <**host**> -d <database\_name> -U <User\_name>

Ключи -h, -d, -U - это общие ключи, обозначающие соответственно сервер, к которому производится подключение, название базы данных и имя пользователя, от имени которого добавляется таблица. Пользователь должен обладать правами на добавление новых таблиц в БД.

1. *PostGIS 2.0 Shapefile and DBF Loader/Exporter.* Приложение, представляющее собой ту же утилиту shp2pgsql, но с графическим интерфейсом (Рисунок 7).

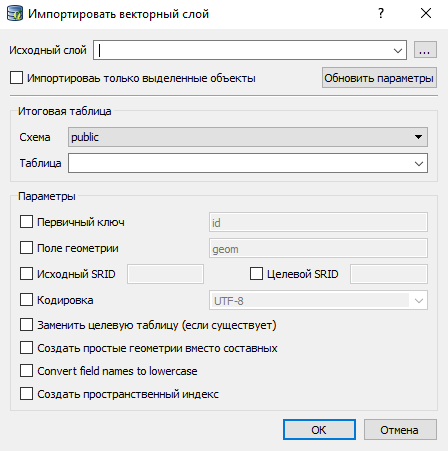


*Рисунок 7. Интерфейс ПО PostGIS Shapefile Import/Export Manager.*

По кнопке View connection details вводим параметры подключения к базе.

По кнопке Options можем указать дополнительные параметры импорта: кодировку, регистр наименований полей, формирование простых геометрий вместо составных.

1. *Модуль DBManager для QGIS.* Специальный модуль для настольной ГИС QGIS, позволяющий загрузить shp-файлы, открытые в программе при наличии подключения к БД. Выбираем нужное подключение, по кнопке Импорт открывается новое окно (Рисунок 8).



*Рисунок 8. Импорт векторного слоя в БД с помощью модуля Менеджер БД.*

Выбираем слой, схему, наименование будущей таблицы, поля первичного ключа, поля геометрии, системы координат.

## 4.2. Интерфейсы подключения к базе данных: pgAdmin IV и QGIS.

Подключение к базе данных осуществляется за счет выполнения SQL-запросов. Большая часть таких запросов легко унифицируется и выглядит одинаково независимо от того, для каких данных в конечном счете мы используем СУБД. Логичным решением является создание клиентского ПО, упрощающего процесс выполнения простых рутинных операций, а также имеющего удобный графический интерфейс для наглядного представления структуры базы данных и таблиц.

***pgAdmin IV***

Удобное клиентское ПО с открытым кодом, которое используется для администрирования баз данных (рис. 11). Установочный пакет поставляется сразу вместе с установочным пакетом PostgreSQL.

На стартовой странице есть несколько вкладок, на которых отражается общая информация о базах данных и таблицах:

* Dashboard. Здесь администратор БД может отслеживать активные сессии, графики транзакций, проводимых в БД, возникающие блокировки. А также параметры, установленные в файле конфигураций postgresql.conf, о котором рассказывалось выше;
* Properties. Свойства сервера, базы данных, таблицы и др. в зависимости от активного объекта. Например: список всех баз данных, хранимых на сервере, их владельцев, а также отметка о базе данных, установленной по умолчанию;
* SQL. Запрос, с помощью которого был создан активный объект;
* Statistics. Информация об объекте. Например для таблицы: физический размер таблицы, количество кортежей, количество удаленных кортежей;
* Dependencies. Зависимости: схема, в которой находится таблица, роль, от имени которой был создан объект;
* Dependents. Объекты, зависящие от активного объекта: триггеры, внешние и первичные ключи, индексы, последовательности.

Помимо отслеживания общей информации, в pgAdmin реализован графический интерфейс для создания различных объектов с возможностью посмотреть сам SQL-запрос, в котором отображаются все указанные свойства.

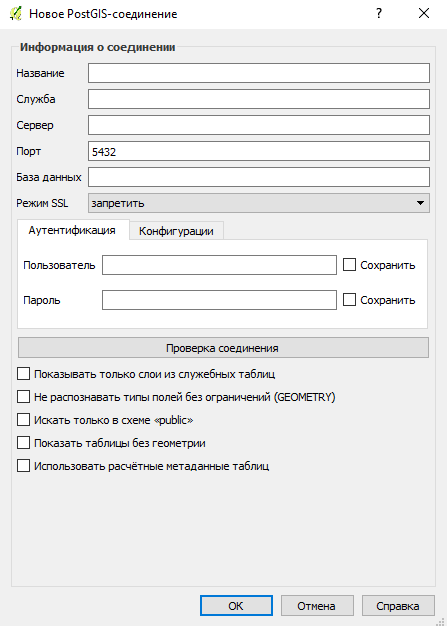
А также реализованы следующие возможности:

* Быстрый просмотр данных, хранимых в таблицах и представлениях;
* Быстрое создание новых объектов;
* Изменение свойств уже существующих объектов;
* Удаление и каскадное удаление объектов;
* Создание бэкапов и восстановление данных из резервных копий;
* Экспорт и импорт данных в форматы binary, text, csv и др.

Стандартно есть окно Query tool, в котором можно написать свой запрос или загрузить его из файла с расширением .sql. Там же можно посмотреть историю последних запросов или выгрузить результат запроса как csv-таблицу.

***QGIS***

Настольная открытая геоинформационная система, в которой осуществлялась вся основная работа над проектом. Поддерживает добавление и редактирование PostGIS-слоев (Рисунок 9).



*Рисунок 9. Создание PostGIS-соединения в QGIS.*

С PostGIS-слоями работают все основные функции создания и редактирования векторных объектов. Исключения могут накладываться лишь при несовместимости какого-либо инструмента с типом геометрии таблицы. Например, для таблицы с типом геометрии Polygon, не сработает инструмент создания мультигеометрий.

Стили, настраиваемые в QGIS, можно сохранять по умолчанию в базу данных. Это делается через свойства слоя, в БД создается специальная таблица layer\_styles, содержащая описания стилей в формает qml и sld.

## 4.3. Применение методики поиска ошибок исходных кадастровых данных и варианты использования конечных данных на примере территории одного субъекта РФ.

***Выявление дубликатов земельных участков***

Для поиска геометрических дублей у PostGIS существует функция ST\_Equals(geometry, geometry), где аргументы geometry и geometry это колонки, содержащие геометрию первой и второй таблиц соответственно. Так как в нашем случае мы ищем совпадения геометрий в пределах одной таблицы, можно сравнить ее с самой собой, задав для нее псевдонимы. Выглядеть это будет следующим образом:

**SELECT** t1.\* **FROM** polygons\_source **AS** t1, polygons\_source **AS** t2

**WHERE** ST\_Equals(t1.geom, t2.geom);

Таким образом, эта функция возвращает булево выражение: TRUE, если совпадение есть, в противном случае FALSE. Теперь необходимо добавить условия поиска. Так как мы сравниваем две абсолютно идентичных таблицы, логично, что каждый для каждого объекта найдется совпадающий. Поэтому, во-первых, добавим ограничение table1.gid<>table2.gid. А также нас интересуют не просто дублирующиеся объекты (такие мы должны были исключить на этапе подготовки исходных данных), а с различным кадастровым номером. Отсюда вытекает второе условие table1.”Кадастровый номер земельного участка”<>table2.”Кадастровый номер земельного участка”. Теперь сгруппируем объекты по геометрии, а кадастровые номера выведем через точку с запятой. Получим следующий запрос:

**SELECT** string\_agg(t1."Кадастровый номер земельного участка",';') **AS** "Кадастровые номера", t1.geom

**FROM** polygons\_source **AS** t1, polygons\_source **AS** t2  
**WHERE** ST\_Equals(t1.geom, t2.geom) **and** t1.gid<>t2.gid **and** t1."Кадастровый номер земельного участка"<>t2."Кадастровый номер земельного участка"  
**GROUP** **BY** t1.geom;

Функция string\_agg - агрегатная функция, создающая строку из входных элементов с заданным разделителем.

Но с использованием стандартной функции обработки геометрии решение задачи получается чересчур громоздким. Можно сделать его лаконичнее просто сгруппировав объекты по геометрии, если кадастровых номеров для данной геометрии больше одного:

**SELECT** string\_agg("Кадастровый номер земельного участка", ';') **AS** "Кадастровые номера", geom   
**FROM** polygons\_source   
**GROUP** **BY** geom   
**HAVING** **COUNT**("Кадастровый номер земельного участка")>**1**;

Таким образом на 421449 участков по данным КПТ было выявлено 16678 дублирующихся объектов общей площадью 402, 8 км2.

***Поиск пересечений земельных участков***

В PostGIS предусмотрены 2 функции, определяющие пересечения:

* ST\_Intersects(geometry, geometry) - возвращает TRUE, если существует пересечение между объектами. Аргументы соответствуют колонкам геометрии таблиц, между которыми происходит поиск пересечений;
* ST\_Intersection(geometry, geometry) - возвращает геометрию пересечения между объектами;
* ST\_Within(geometry A,geometry B) - возвращает TRUE, если геометрия А полностью находится в геометрии В.

Нас интересует именно геометрия пересечений. Однако из общего числа пересечений, найденных с помощью функции, необходимо исключить те, в которых область пересечения полностью повторяет геометрию одного из объектов. Это означает, что один объект целиком находится в другом. Такие ситуации скорее можно отнести к дублям.

Во всем регионе было найдено 67588 пересечений общей площадью 169,4 км2.

***Самозахваты***

Самозахватом называют самовольное занятие земельного участка лицом, не имеющим предусмотренных законодательством Российской Федерации прав на указанный земельный участок. Согласно статье 7.1 Кодекса Российской Федерации об административных правонарушениях (от 30.12.2001г.) самовольное занятие земельного участка влечет за собой наложение административного штрафа. Однако, если площадь самовольно занятых земель не превышает 10% от общей площади земельного участка, возникает возможность перераспределения земельных участков.

Одним из видов производных данных является определение самозахватов. SQL-запрос, с помощью которого выводятся самозахваты, представлен в Приложении 4.

Во всем регионе было выявлено около 67 тыс. самозахватов общей площадью 34,5 км2.

***ЛПХ***

ЛПХ - вид разрешенного использования земельных участков, означающий личное подсобное хозяйство. Земельный участок для ЛПХ выделяется как в пределах населенных пунктов, так и в пределах земель сельскохозяйственного назначения (полевой земельный участок). На второй вид участков для ЛПХ налагаются различные, утвержденные действующим законодательством, ограничения, поэтому их необходимо выделять отдельно.

В классификаторе «Виды разрешенного использования» личному подсобному хозяйству соответствует код 141003000000, а категория земель сельскохозяйственного назначения обозначается кодом 003001000000. По этим двум условиям мы можем выделить в слоях zu\_cad, zu\_agro\_1 и zu\_agro\_2 интересующие нас участки.

SQL-запрос приводится в Приложении 5.

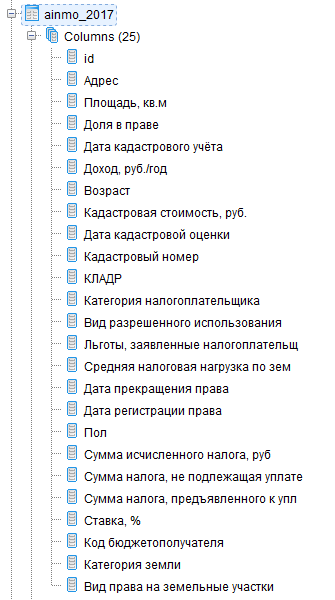
***Налоги***

Вместе с кадастровыми данными в качестве исходных также были получены данные Федеральной налоговой службы. Это таблицы, не содержащие пространственных данных, их структуру можно видеть на Рисунке 14.

Используя наши данные, можно визуализировать налоговую статистику, т.к. у них есть общее поле «Кадастровый номер».

**SELECT** t1."Кадастровый номер земельного участка", t1."Контур", t2."Вид права на земельные участки", t2."Кадастровая стоимость, руб.", t1."Площадь земельного участка", t2."Площадь, кв.м", t1.geom   
**FROM** zu\_cad **AS** t1, ainmo\_2017 **AS** t2  
**WHERE** t1."Кадастровый номер земельного участка"=t2."Кадастровый номер"

На Рисунке 15 наглядно представлено, что налог платится далеко не со всех участков, стоящих на кадастровом учете. Не говоря уже о тех, которые на кадастровом учете не стоят.



*Рисунок 10.Структура данных ФНС.*



*Рисунок 11. Схема наложения данных ФНС на исходные кадастровые данные. Фиолетовым цветом показаны участки, стоящие на учете в ФНС и на кадастровом учете, голубым цветом - участки, стоящие только на кадастровом учете*

# Заключение

В процессе выполнения поставленной компанией Геоскан задачи была разработана структура базы данных на основе СУБД PostgreSQL и полностью осуществлен переход на новую схему организации работ с использованием системы управления базами пространственных данных. В итоге:

1. Значительно сократилось время выполнения различных операций над исходными и конечными данными, возникающих в процессе выполнения проекта;
2. Автоматизированы все этапы работы и, как следствие, исключены некоторые из них. А именно: сборка материалов от разных сотрудников, проверка на наличие топологических ошибок и полноты атрибутивных данных;
3. Оптимизирован процесс поиска ошибок исходных данных за счет разработанных универсальных SQL-запросов;
4. Снижена вероятность возможных ошибок, причиной которых является человеческий фактор за счет установки ограничений доступа, топологических правил.
5. Была разработана схема заполнения таблиц учета изменений, которые выполняли роль резервных копий, а также позволяли осуществлять дополнительный контроль за ходом работ.

Также совокупными средствами ГИС и СУБД производился расчет оплаты труда, что играет огромную роль в администрировании работ.

Таким образом, все поставленные задачи были выполнены, проект в рамках Национальной технологической инициативы, выполняемый компанией Геоскан, был успешно завершен. Разработанная структура базы данных активно использовалась в работе над проектом, а также планируется ее использование для дальнейших подобных задач, модифицируя и усовершенствуя.

# Список литературы

1. Баканов В.М. Введение в язык SQL запросов к базам данных. Учебное пособие. М.: «Изд-во МГАПИ», 2002, 61 c.;
2. Моргунов Е.П. Язык SQL. Базовый курс: учеб.-практ. пособие. М: «Postgres Professional», 2017, 257 с.;
3. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных. Учебник для высших учебных заведений. 6-е изд. М.: «Корона-Принт», 2004, 736 с.;
4. Date C.J. An Introduction to Database Systems. 8th Edition. MA: «Addison-Wesley Longman Publishing Co», 2004, 1034 с.;
5. <http://www.consultant.ru/document/cons_doc_LAW_60683/> - Водный кодекс Российской Федерации (от 03.06.2006 № 74-ФЗ), 10 октября 2017;
6. <http://www.consultant.ru/document/cons_doc_LAW_51040/> - Градостроительный кодекс Российской Федерации (от 29.12.2004 № 190-ФЗ), 10 октября 2017;
7. <http://www.consultant.ru/document/cons_doc_LAW_5142/> - Гражданский Кодекс Российской Федерации (от 30.11.1994 № 51-ФЗ), 9 октября 2017;
8. <http://www.consultant.ru/document/cons_doc_LAW_51057/> - Жилищный кодекс Российской Федерации (от 29.12.2004 № 188-ФЗ), 10 октября 2017;
9. <http://www.consultant.ru/document/cons_doc_LAW_33773/> - Земельный кодекс Российской Федерации (от 25.10.2001 № 136-ФЗ), 9 октября 2017;
10. <http://www.constitution.ru/> - Конституция Российской Федерации (принята всенародным голосованием 12.12.1993) (с учетом поправок, внесенных Законами РФ о поправках к Конституции РФ от 30.12.2008 N 6-ФКЗ, от 30.12.2008 N 7-ФКЗ, от 05.02.2014 N 2-ФКЗ), 8 октября 2017;
11. <http://www.consultant.ru/document/cons_doc_LAW_64299/> - Лесной кодекс Российской Федерации (от 04.12.2006 № 200-ФЗ), 10 октября 2017;
12. <http://www.consultant.ru/document/cons_doc_LAW_85128/> - Приказ Минэкономразвития РФ от 24.12.2008 N 467 "Об утверждении требований к составу, структуре, порядку ведения и использования единой электронной картографической основы федерального, регионального и муниципального назначения" (Зарегистрировано в Минюсте РФ 18.02.2009 N 13396), 23 декабря 2017;
13. <http://hanslodge.com/extensible_markup_language.html> - Спецификация Extensible Markup Language (XML) 1.0 (Second Edition) (W3C Recommendation), 2000, 41 с.;
14. <http://www.consultant.ru/document/cons_doc_LAW_182661/> - Федеральный закон №218 «О государственной регистрации недвижимости» (от 13.07.2015), 15 октября 2017;
15. <http://www.consultant.ru/document/cons_doc_LAW_70088/> - Федеральный закон №221 «О кадастровой деятельности» (от 24.07.2007), 15 октября 2017;
16. <https://docs.microsoft.com/ru-ru/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017> - онлайн-документация MS SQL Server 2017;
17. <https://downloads.mysql.com/docs/refman-5.7-en.pdf> - MySQL 5.7 Reference Manual, 2018, 5560 с.;
18. <https://docs.oracle.com/database/121/SPATL/toc.htm> - Oracle Spatial and Graph Developer’s Guide 12*c* Release 1 (12.1), 2017, 475 с.;
19. <https://postgis.net/stuff/postgis-2.4.pdf> - PostGIS 2.4.5dev Manual, 2018, 830 с.;
20. <https://www.postgresql.org/files/documentation/pdf/9.6/postgresql-9.6-A4.pdf> - PostgreSQL 9.6.9 Documentation, 3425 с.;

## Приложение 1а. Схема разработанной базы данных.



## Приложение 1б. Схема разработанной базы данных.



## Приложение 1в. Схема разработанной базы данных.



## Приложение 2. Код SQL-запроса, создающего триггерную функцию add\_to\_log, заполняющую таблицу logs.

**CREATE** **OR** **REPLACE** **FUNCTION** add\_to\_log() **RETURNS** **TRIGGER** **AS** $$  
  
**DECLARE**  
retstr varchar(**254**);  
**BEGIN**  
IF TG\_OP = 'INSERT' **THEN**  
retstr := 'INSERT';  
**INSERT** **INTO** logs(action\_,user\_name,object\_gid,time,geom, table\_name) **values** (retstr,**session\_user**,**NEW**.gid,NOW(),**NEW**.geom, TG\_TABLE\_NAME);  
**RETURN** **NEW**;  
ELSIF TG\_OP = 'UPDATE' **THEN**  
retstr := 'UPDATE';  
**INSERT** **INTO** logs(action\_,user\_name,object\_gid,time,geom, table\_name) **values** (retstr,**session\_user**,**NEW**.gid,NOW(),**OLD**.geom, TG\_TABLE\_NAME);  
**RETURN** **NEW**;  
ELSIF TG\_OP = 'DELETE' **THEN**  
retstr := 'DELETE';  
**INSERT** **INTO** logs(action\_,user\_name,object\_gid,time,geom, table\_name) **values** (retstr,**session\_user**,**OLD**.gid,NOW(),**OLD**.geom, TG\_TABLE\_NAME);  
**RETURN** **OLD**;  
**END** IF;  
**END**;  
$$ **LANGUAGE** plpgsql;

Триггер:

**CREATE** **TRIGGER** logs\_trigger  
**AFTER** **INSERT** **OR** **UPDATE** **OR** **DELETE** **ON** <**table\_name>** **FOR** **EACH** **ROW** **EXECUTE** **PROCEDURE** add\_to\_log ();

## Приложение 3. Код SQL-запроса, создающего триггерную функцию для проверки топологических правил.

**DECLARE**  
tab varchar(**254**);  
temprow RECORD;  
tables\_ varchar[]:=ARRAY['ru','zu','zu\_cad'];  
**BEGIN**  
IF TG\_OP = 'INSERT' **or** TG\_OP = 'UPDATE'

**THEN** FOREACH tab **IN** ARRAY tables\_  
 LOOP  
 IF tab = TG\_TABLE\_NAME **THEN**

**FOR** temprow **in** **execute** format('SELECT ST\_Overlaps($1.geom,t1.geom) result\_ FROM<schema\_name>.%I t1 WHERE $1.gid<>t1.gid UNION ALL SELECT ST\_Equals($1.geom,t1.geom) result\_ FROM <schema\_name>.%I t1 WHERE $1.gid<>t1.gid',TG\_TABLE\_NAME,TG\_TABLE\_NAME) **USING NEW**  
LOOP  
IF temprow.result\_ = **TRUE** **THEN**  
RAISE exception 'TOPOLOGY ERROR: %', ST\_extent(**NEW**.geom);  
**END** IF;  
**END** LOOP;  
**ELSE** **FOR** temprow **in** **execute** format('SELECT ST\_Overlaps($1.geom,t1.geom) result\_ FROM <schema\_name>.%I t1 UNION ALL SELECT ST\_Equals($1.geom,t1.geom) result\_ FROM <schema\_name>.%I t1',tab, tab) **USING NEW**  
LOOP  
IF temprow.result\_ = **TRUE** **THEN**  
RAISE exception 'TOPOLOGY ERROR: %', ST\_extent(**NEW**.geom);  
**END** IF;  
**END** LOOP;  
**END** IF;  
**END** LOOP;  
**RETURN** **NEW**;  
ELSIF TG\_OP = 'DELETE' **THEN**  
**RETURN** **OLD**;  
**END** IF;  
**END**;

## Приложение 4. Код SQL-запроса для выявления самозахватов.

**SELECT** t3."Кадастровый номер земельного участка", t3."Контур", t3.gid,  
 t3.current\_area - t3.source\_area **AS** sz\_area,  
(t3.current\_area / t3.source\_area - **1**) \* **100** **AS** percentage,  
t3.geom **AS** mygeom  
**FROM** ( **SELECT** t1.gid, t1.geom,  
t1."Кадастровый номер земельного участка",  
t1."Контур",  
st\_area(t1.geom) **AS** current\_area,  
st\_area(t2.geom) **AS** source\_area  
**FROM** zu\_cad **AS** t1  
**LEFT** **JOIN** polegon\_source **AS** t2 **ON** t1.gid = t2.gid  
**WHERE** ST\_Area(t1.geom) > (**1**.**1** \* ST\_Area(t2.geom))  
**UNION** **ALL**  
**SELECT** ru.gid, ru.geom,  
ru."Кадастровый номер земельного участка", ru."Контур",  
ST\_Area(ru.geom) **AS** ST\_Area, ru."Площадь, кв.м"

**FROM** ru  
**WHERE** ST\_Area(ru.geom) > (**1**.**1** \* ru."Площадь, кв.м")) t3

## Приложение 5. Код SQL-запроса для выделения участков с видом разрешенного использования «Личное подсобное хозяйство».

(**SELECT** zu\_cad."Кадастровый номер земельного участка",  
zu\_cad."Контур", zu\_cad."Вид использования участка по документу",  
zu\_cad."Категория земель", zu\_cad.geom,  
ST\_Area(zu\_cad.geom) **AS** "Area"  
**FROM** zu\_cad  
**WHERE** zu\_cad."Вид разрешенного использования"= '141003000000' **AND** zu\_cad."Категория земель" = '003001000000'  
**UNION**  
**SELECT** zu\_agro\_1."Кадастровый номер земельного участка",  
zu\_agro\_1."Контур",  
zu\_agro\_1."Вид использования участка по документу",  
zu\_agro\_1."Категория земель", zu\_agro\_1.geom,  
ST\_Area(zu\_agro\_1.geom) **AS** "Area"  
**FROM** zu\_agro\_1  
**WHERE** zu\_agro\_1."Вид разрешенного использования" = '141003000000' **AND** zu\_agro\_1."Категория земель" = '003001000000'  
**UNION**  
**SELECT** zu\_agro\_2."Кадастровый номер земельного участка",  
zu\_agro\_2."Контур",  
zu\_agro\_2."Вид использования участка по документу",  
zu\_agro\_2."Категория земель", zu\_agro\_2.geom,  
ST\_Area(zu\_agro\_2.geom) **AS** "Area"  
**FROM** zu\_agro\_2  
**WHERE** zu\_agro\_2."Вид разрешенного использования" = '141003000000' **AND** zu\_agro\_2."Категория земель" = '003001000000');

1. proj4 - открытая библиотека, содержащая параметры географических систем координат и картографических проекций (https://proj4.org) [↑](#footnote-ref-1)