



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ*

### *НА ТЕМУ:*

*«Классификация методов поиска географических  
данных в реляционных базах данных»*

Студент ИУ7-74Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Ланкин Д. Л.  
(И. О. Фамилия)

Руководитель НИР

\_\_\_\_\_  
(Подпись, дата)

Кузнецова О. В.  
(И. О. Фамилия)

*2024 г.*

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие сокращения и обозначения.

ГИС — географическая информационная система

СУБД — система управления базами данных

JSON — JavaScript Object Notation

# ВВЕДЕНИЕ

Последнее десятилетие в нашей стране отмечается повышенным вниманием к геоинформатике. За это время опубликовано большое количество статей и книг, посвященных вопросам и проблемам связанным с ней. Вместе с тем сравнительно недавно возникла потребность работы с географическими данными через среду Internet. Это связано с рядом факторов: широким распространением Internet-технологии в Российской Федерации, появлением высокопропускных каналов, развитием микроэлектронной базы ЭВМ [1].

Пространственные данные необходимы для предупреждения природных явлений различного рода (например, анализ сейсмической активности при выборе этажности планируемой застройки), в процедурах городского планирования (выбор места постройки больницы, детского сада и аналогичных объектов социальной инфраструктуры). Геоинформационные системы и технологии являются мощным инструментом информационной поддержки процессов планирования и управления в транспортных системах [2].

Целью работы является классификация методов поиска пространственных данных в реляционных базах данных.

Чтобы достигнуть поставленной цели, необходимо решить следующие задачи:

- провести анализ предметной области географических (пространственных) данных;
- провести обзор существующих методов хранения географических (пространственных) данных в реляционных СУБД;
- сформулировать критерии классификации методов поиска географических (пространственных) данных;
- классифицировать методы поиска географических (пространственных) данных.

# **1 Аналитический раздел**

Необходимо рассмотреть способы организации хранения пространственных данных в реляционных СУБД.

## **1.1 Способы хранения географических данных в реляционных СУБД**

### **1.1.1 Использование XML**

### **1.1.2 Использование JSON**

Для хранения географических (пространственных) данных возможно использовать JSON в случае, если СУБД поддерживает этот тип данных.

GeoJSON это основанный на JSON открытый стандарт и формат, созданный для представления простых географических объектов наряду с их непространственными атрибутами [3].

### **1.1.3 Использование расширений СУБД**

В настоящее время многие известные системы управления реляционными базами данных предлагают возможности хранения географических (пространственных) данных с применением сторонних расширений [4].

Для СУБД Postgres [5] существует расширение PostGIS [6], а для Oracle [7] — Oracle Spatial [8].

Расширения СУБД для работы с географическими (пространственными) данными вводят новые типы данных для хранения геометрий (точка, линия, полигон), а также операции работы с ними.

Преимущества:

1)

Недостатки:

1) Необходимо детально разрабатывать схему хранения данных [4].

2) Трудоемкий процесс загрузки данных в связи с возможной необходимостью преобразований атрибутов и геометрий [4].

## 1.2 Методы поиска

### 1.2.1 Последовательный поиск

Последовательный поиск – это метод доступа к данным, при котором СУБД читает все строки таблицы, чтобы найти те, которые соответствуют заданным критериям. Этот метод может быть эффективен в следующих случаях:

- отсутствие индексов;
- большая выборка данных в результате выполнения запроса, вернувшего значительное число строк (более 50% от общего числа). Связано с тем, что в данном случае последовательный поиск более эффективен в связи с минимизацией количества операций ввода–вывода [9].

Последовательный поиск стоит применять в следующих ситуациях:

- небольшие таблицы: для небольших наборов данных последовательный поиск может быть быстрым и эффективным;
- запросы на большую выборку: если запрос возвращает много строк или фильтрация по критериям не очень узкая (например, выбираются все записи за определённый период), то последовательный поиск будет предпочтительным [9].

### 1.2.2 Использование индекса Quadtree

Индекс Quadtree вычисляет аппроксимации фрагментов для геометрий и использует существующие индексы В-дерева для выполнения пространственного поиска и других операций с DML [10].

QuadTree — это иерархическая структура данных, используемая для разбиения двумерного пространства на четыре квадранта или подрегионы. Каждый узел дерева представляет собой квадрат, который может быть разделён на четыре равных квадрата, если в нём содержится больше определённого количества объектов. Это позволяет эффективно выполнять операции поиска, такие как нахождение всех объектов в заданной области [11].



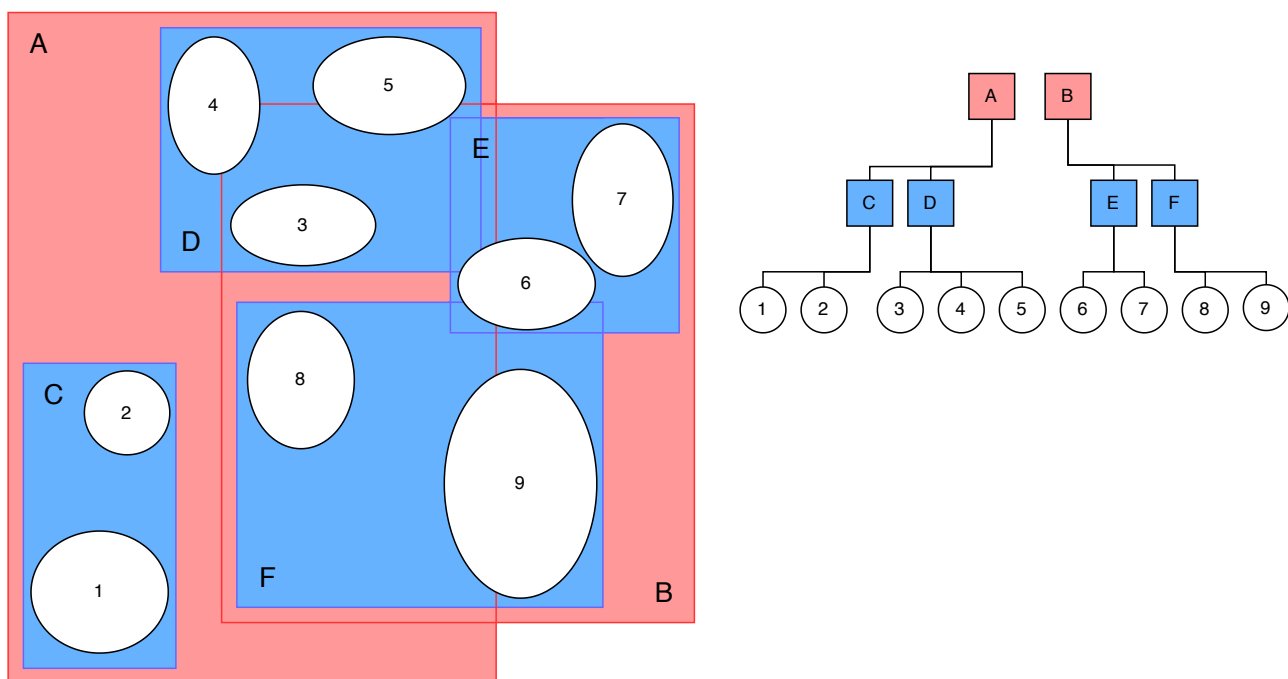


Рисунок 1.2 – Визуализация структуры R-Tree

Процедура вставки нового элемента в R-дерево состоит из:

- 1) Поиск подходящего узла: находится узел, в который можно добавить элемент, минимизируя увеличение объема его ограничивающего прямоугольника.
- 2) Добавление элемента: если узел переполняется (превышает максимальное количество элементов), он разделяется на два узла с использованием алгоритмов кластеризации, таких как квадратичное или линейное разбиение.
- 3) Поддержание баланса: структура R-дерева остается сбалансированной, что обеспечивает эффективный поиск.

Квадратичное разбиение заключается в разбиении на два прямоугольника с минимальной площадью, покрывающие все объекты. Линейный – в разбиении по максимальной удаленности.

Поиск в R-дерево осуществляется путем рекурсивного обхода дерева, начиная с корня. Каждый узел содержит информацию о своих дочерних узлах и ограничивающих прямоугольниках, что позволяет быстро отсеивать ненужные ветви дерева.

### 1.3 Сравнительная таблица индексов, основанных на деревьях

Характеристика	R-Tree	Quadtree
<b>Структура</b>	Иерархическая структура с ограничивающими прямоугольниками	Делит пространство на квадраты
<b>Балансировка</b>	Сбалансированное дерево	Не обязательно сбалансировано
<b>Перекрытие узлов</b>	Узлы могут перекрываться	Узлы не перекрываются
<b>Наличие пустых листьев</b>	Невозможно	Возможно
<b>Обновления</b>	Могут требовать перестройки узлов	Легче обновлять без значительных изменений в структуре

### Вывод



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Гриценко Ю. Б. Еськин Д. М. Ж. О. И., Игушев К. В.* Web-технология доступа к географическим данным. — URL: <https://cyberleninka.ru/article/n/web-tehnologiya-dostupa-k-geograficheskim-dannym> (дата обращения: 2.10.2024).
2. *Зайцева Н. М., Тайлаков А. А.* ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ГЕОИНФОРМАЦИОННОЙ СИСТЕМЫ // Научный журнал. — 2021. — URL: <https://cyberleninka.ru/article/n/teoreticheskie-osnovy-geoinformatsionnoy-sistemy> (дата обращения: 15.10.2024).
3. *Mohamad Hasan E. P., Badenko V.* Comparative evaluation of NoSQL and relational databases performance while analyzing semi-structured geospatial data. — DOI: <http://dx.doi.org/10.18509/GBP.2019.64>. — URL: [https://www.researchgate.net/profile/Vladimir-Badenko-2/publication/335573656\\_COMPARATIVE\\_EVALUATION\\_OF\\_NOSQL\\_AND\\_RELATIONAL\\_DATABASES\\_PERFORMANCE\\_WHILE\\_ANALYZING\\_SEMI-STRUCTURED\\_GEOSPATIAL\\_DATA/links/5e36ec48299bf1cdb9083203/COMPARATIVE-EVALUATION-OF-NOSQL-AND-RELATIONAL-DATABASES-PERFORMANCE-WHILE-ANALYZING-SEMI-STRUCTURED-GEOSPATIAL-DATA.pdf](https://www.researchgate.net/profile/Vladimir-Badenko-2/publication/335573656_COMPARATIVE_EVALUATION_OF_NOSQL_AND_RELATIONAL_DATABASES_PERFORMANCE_WHILE_ANALYZING_SEMI-STRUCTURED_GEOSPATIAL_DATA/links/5e36ec48299bf1cdb9083203/COMPARATIVE-EVALUATION-OF-NOSQL-AND-RELATIONAL-DATABASES-PERFORMANCE-WHILE-ANALYZING-SEMI-STRUCTURED-GEOSPATIAL-DATA.pdf) (дата обращения: 15.10.2024).
4. *Sveen A.* Efficient storage of heterogeneous geospatial data in spatial databases. // Journal Big Data. — 2019. — Т. 6. — DOI: <https://doi.org/10.1186/s40537-019-0262-8>. — URL: <https://rdcu.be/dYsfo> (дата обращения: 5.10.2024).
5. PostgreSQL Documentation. — URL: <https://www.postgresql.org/docs/> (дата обращения: 5.10.2024).
6. PostGIS Manual. — URL: <https://postgis.net/docs/> (дата обращения: 5.10.2024).
7. Oracle Database Documentation. — URL: <https://docs.oracle.com/en/database/oracle/oracle-database/> (дата обращения: 5.10.2024).
8. Oracle Spatial Documentation. — URL: <https://gdal.org/en/latest/drivers/vector/oci.html> (дата обращения: 5.10.2024).

9. *Obe R. O., Hsu L. S.* PostGIS in Action. — 2nd. — Manning Publications, 2015. — ISBN 9781617291395. — URL: <https://www.manning.com/books/postgis-in-action-second-edition>.
10. *Kanth K., Ravada S., Abugov D.* Quadtree and R-tree indexes in Oracle Spatial: A comparison using GIS data //. — 01.2002. — С. 546—557.
11. *Shekhar S., Chawla S.* Spatial Databases: A Tour. — Prentice Hall, 2003. — ISBN 978-0130174805.
12. Differential Private Spatial Decomposition and Location Publishing Based on Unbalanced Quadtree Partition Algorithm / Y. Yan [и др.] // IEEE Access. — 2020. — Т. 8. — С. 104775—104787. — DOI: 10.1109/ACCESS.2020.2999580.