

PostGIS + k-NN

как найти ближайший бар,
и правда ли, что Земля не плоская

Алексей Фадеев, старший разработчик .NET

О себе

Алесей Фадеев, старший разработчик .NET

Компания sibedge, г.Томск

Сфера деятельности

Backend-разработка

Работа с СУБД, оптимизация

Используем: PostgreSQL,

Microsoft SQL Server

Контакты

Email: fadeevas@sibedge.com

<https://vk.com/fadeev>



Мало докладов по PostGIS...

О чём на самом деле доклад

k-NN

**Непрямо-
угольные
координаты**

ORM

**Индексный
поиск**

**SRID
EPSG**

от произвольной точки

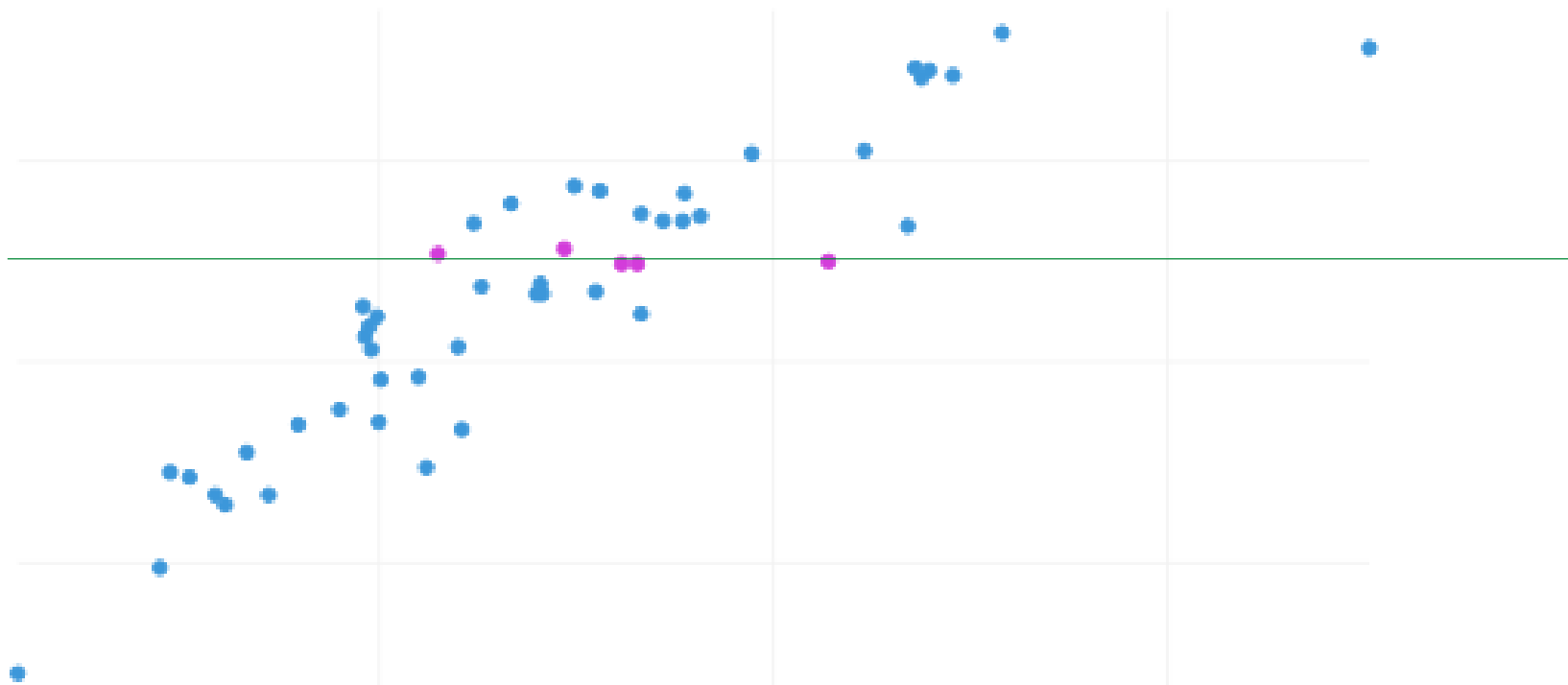
как не запутаться

Поиск в одномерном пространстве

Всё просто

Пример: история показаний датчика за большой период времени

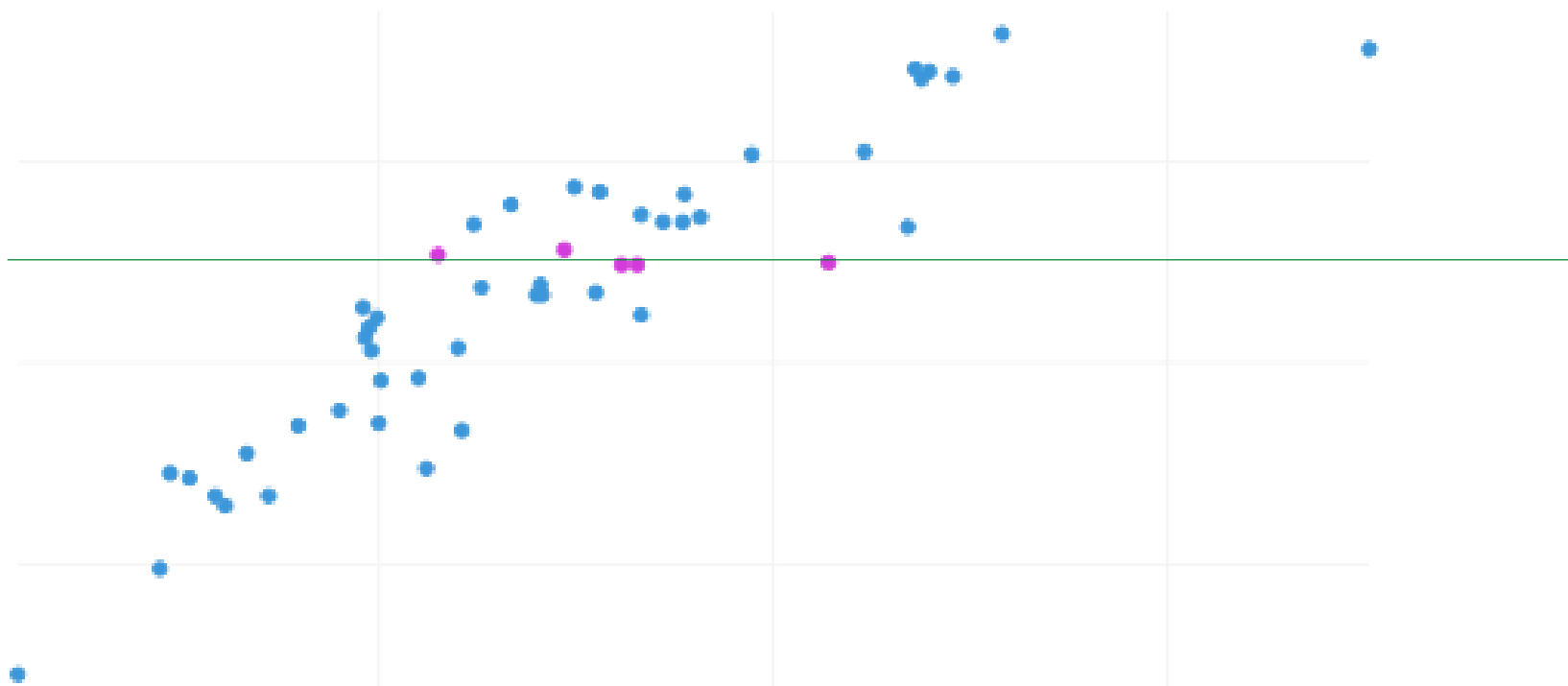
Найти k ближайших значений (в обе стороны)



Поиск в одномерном пространстве

id	datetime	value
1	2020-05-01 13:26:33	19.36
2	2020-05-01 13:26:34	19.35

Найти k ближайших значений (в обе стороны)



Поиск в одномерном пространстве

id	datetime	value
1	2020-05-01 13:26:33	19.36
2	2020-05-01 13:26:34	19.35

Найти k ближайших значений (в обе стороны)

```
SELECT * FROM data
ORDER BY ABS(:myvalue - value)
LIMIT 50
```

Не будет использован индекс

Поиск в одномерном пространстве

id	datetime	value
1	2020-05-01 13:26:33	19.36
2	2020-05-01 13:26:34	19.35

Найти k ближайших значений (в обе стороны)

```
SELECT * FROM  
(SELECT * FROM data WHERE value >= :myvalue  
  ORDER BY value LIMIT 50) s1
```

UNION ALL

```
SELECT * FROM  
(SELECT * FROM data WHERE value < :myvalue  
  ORDER BY value DESC LIMIT 50) s2
```

Поиск в 2D-пространстве

Задача: найти ближайший бар

Может быть так же?

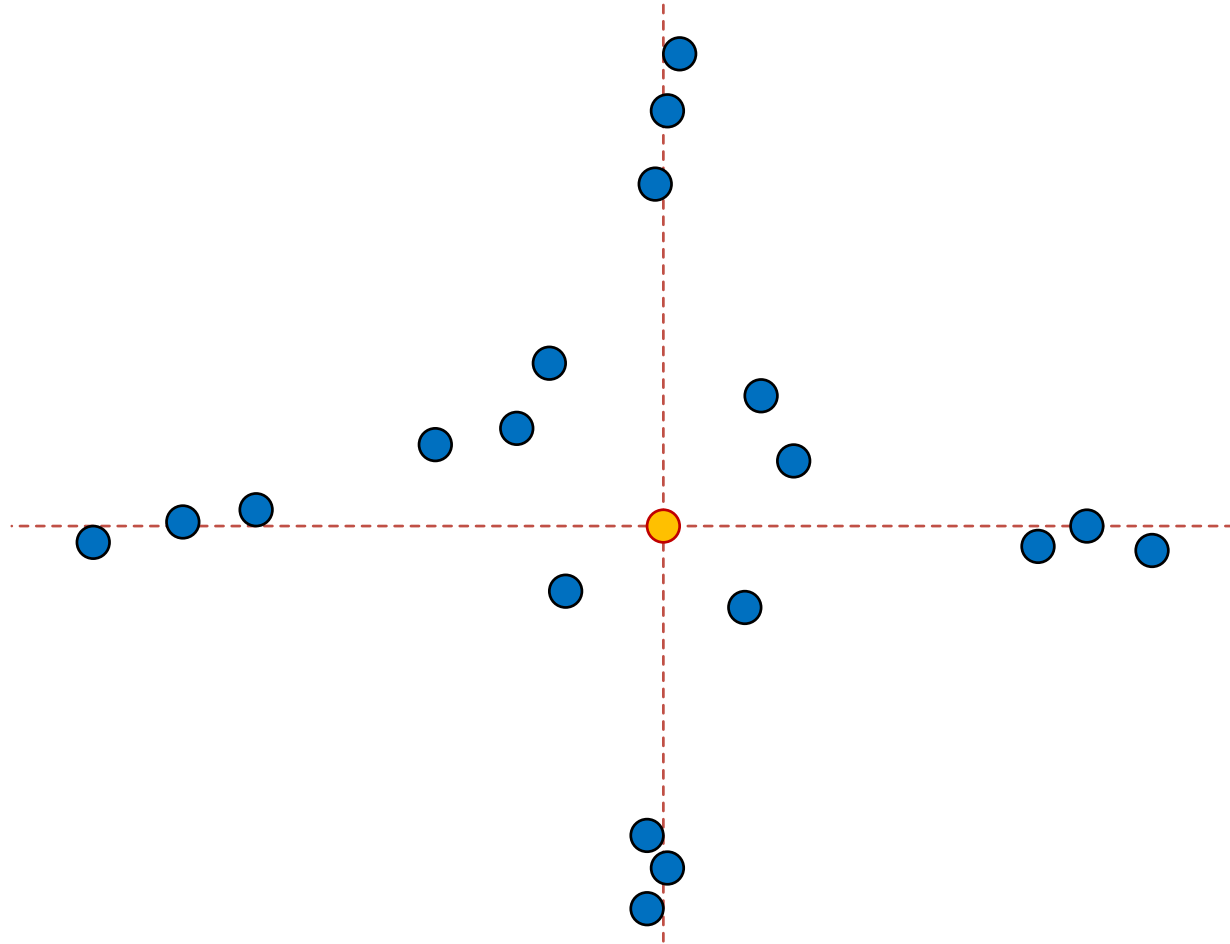
B-tree индекс по каждой координате

Берём k ближайших точек по каждой координате в обе стороны

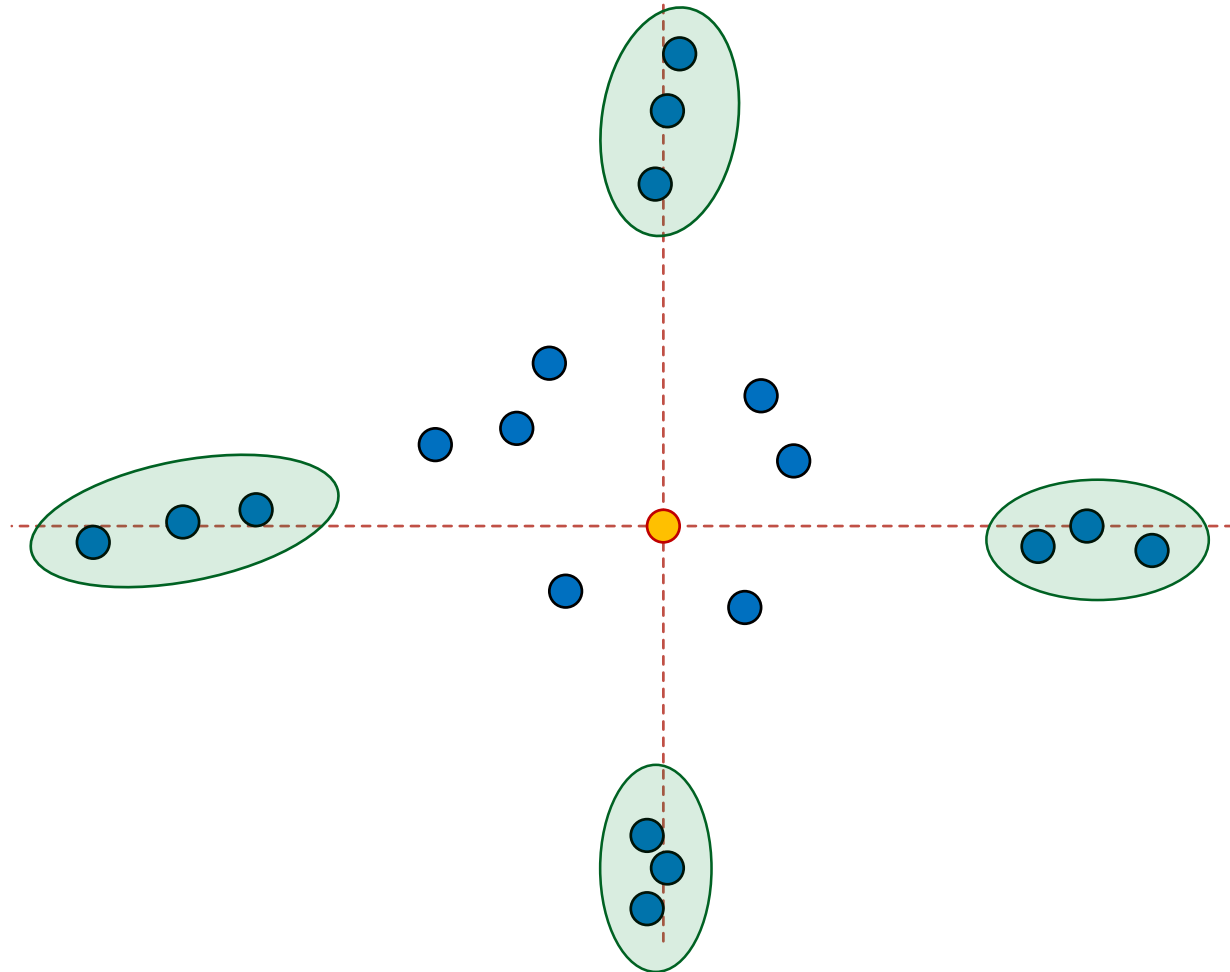
...

Profit?

Поиск в 2D-пространстве

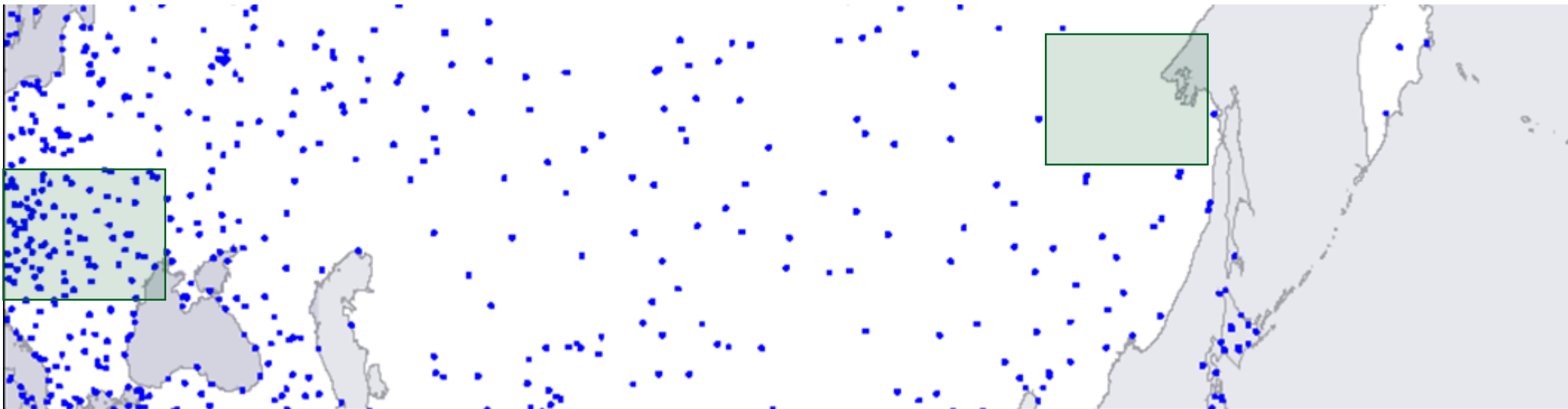


Поиск в 2D-пространстве



Поиск в 2D-пространстве

Ограничить прямоугольную область?



k-NN

Непрямо-
угольные
координаты

ORM

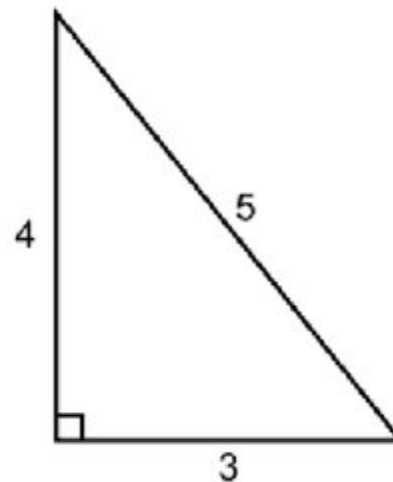
Индексный
поиск

**SRID
EPSG**

Поиск в пространстве (k-NN)

```
SELECT point(3, 0) <-> point (0, 4);
```

5



Поиск в пространстве (k-NN)

	id	amenity	coordinates
	1	shop	(64.2, 88.3)
	2	bar	(50.5, 87.4)

```
SELECT * FROM building
ORDER BY coordinates <-> point (53.7, 87.7)
LIMIT 10;
```

Поиск в пространстве (k-NN)

	id	amenity	coordinates
	1	shop	(64.2, 88.3)
	2	bar	(50.5, 87.4)

```
SELECT * FROM building
ORDER BY coordinates <-> point (53.7, 87.7)
LIMIT 10;
```

Индекс GiST: индексный поиск от произвольной точки

k-NN

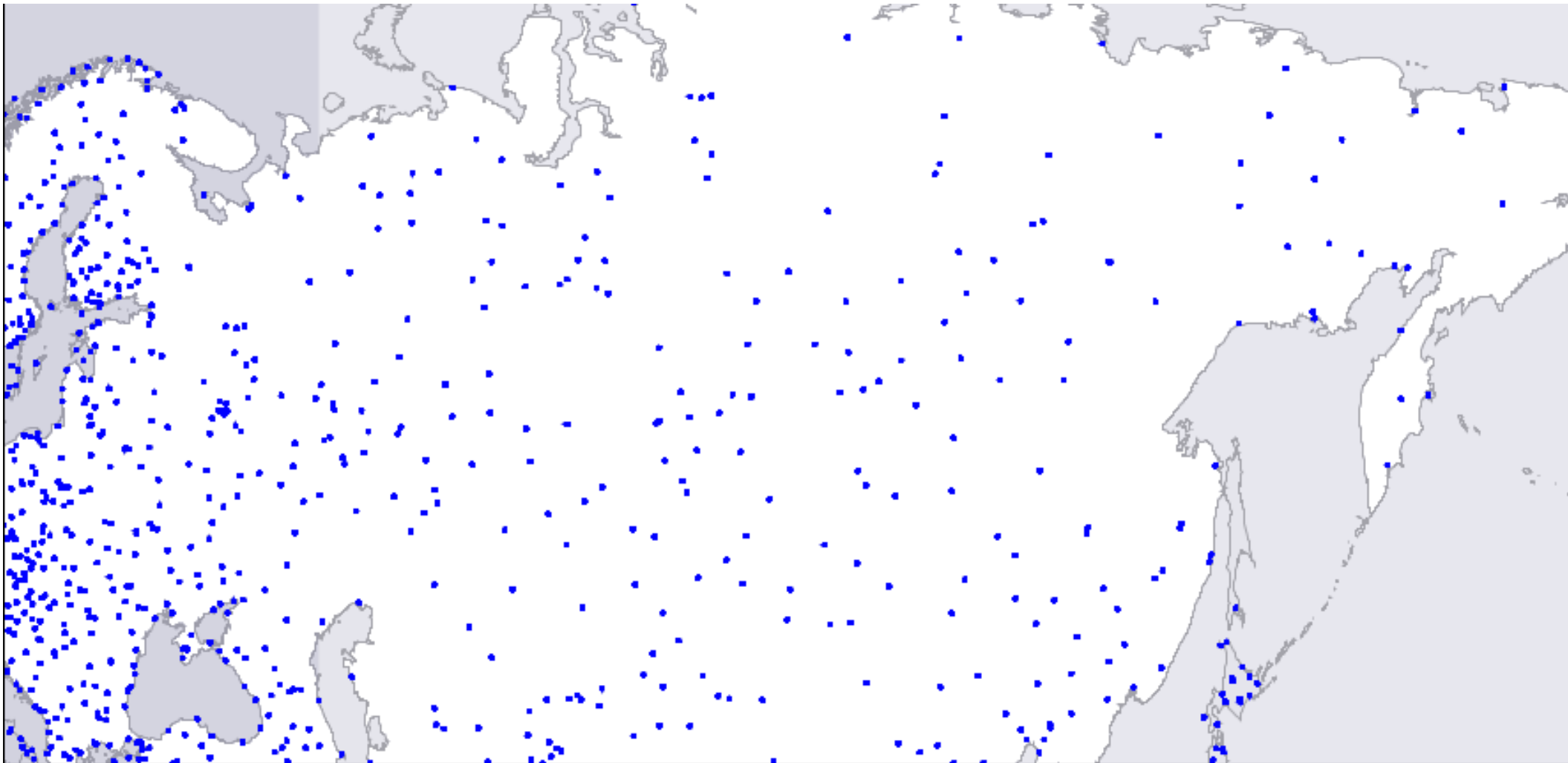
**Непрямо-
угольные
координаты**

ORM

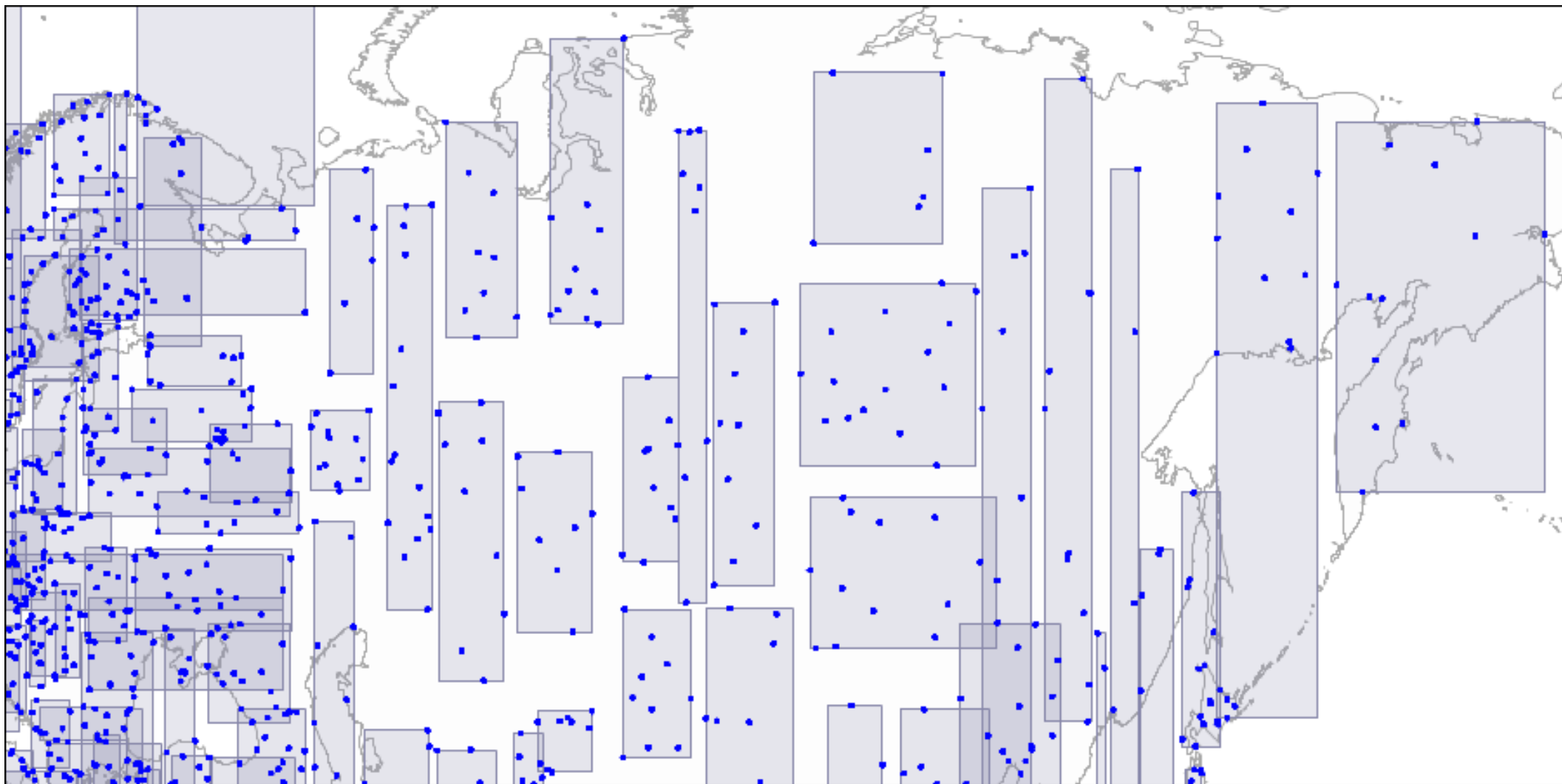
**Индексный
поиск**

**SRID
EPSG**

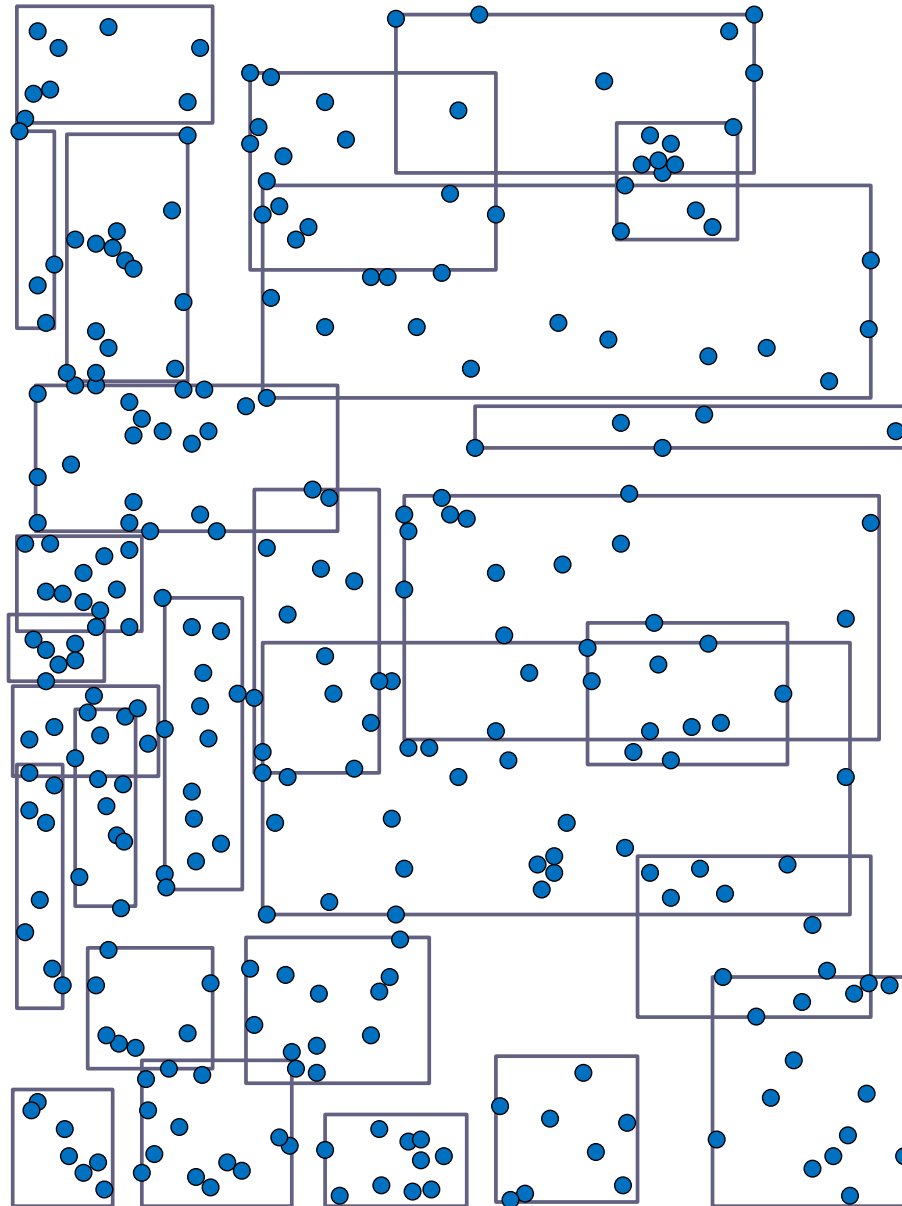
Индекс GiST



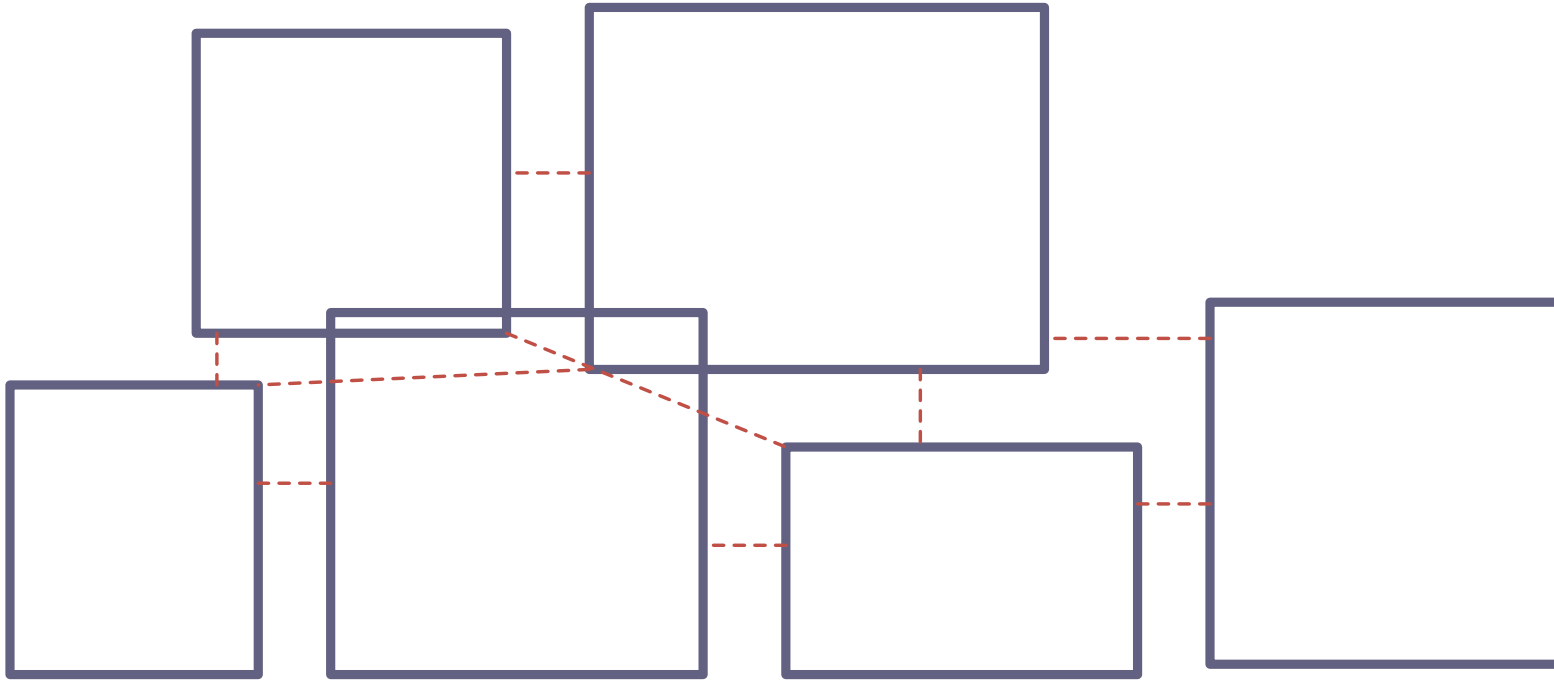
Индекс GiST



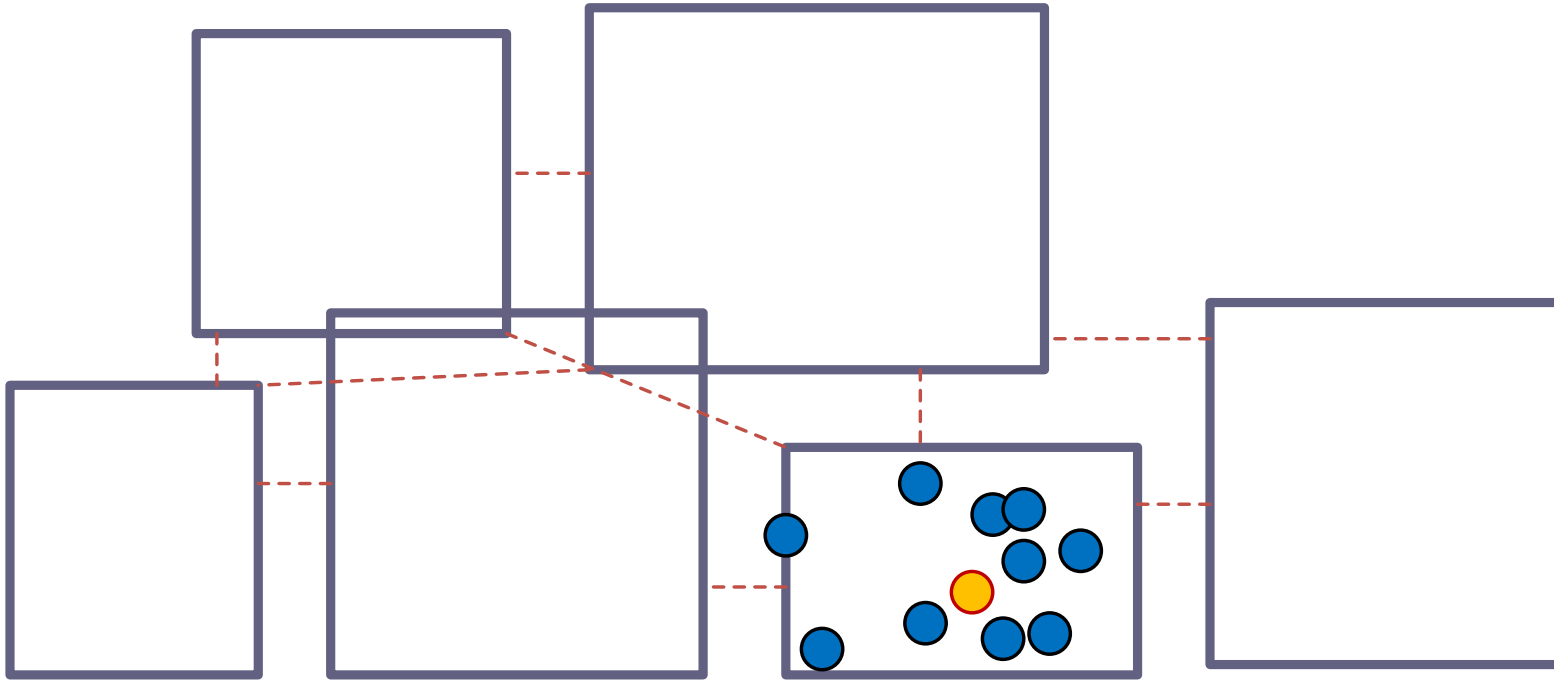
Индекс GiST



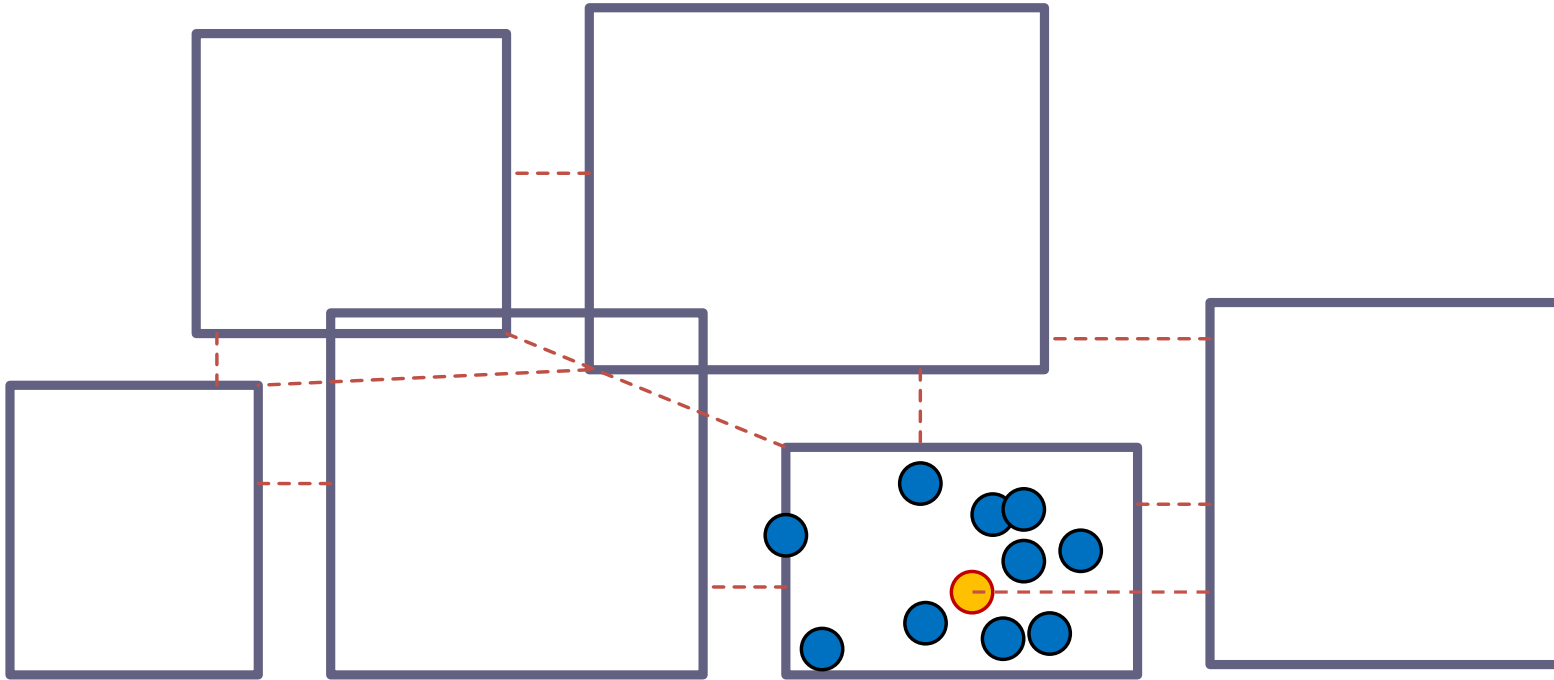
Индекс GiST



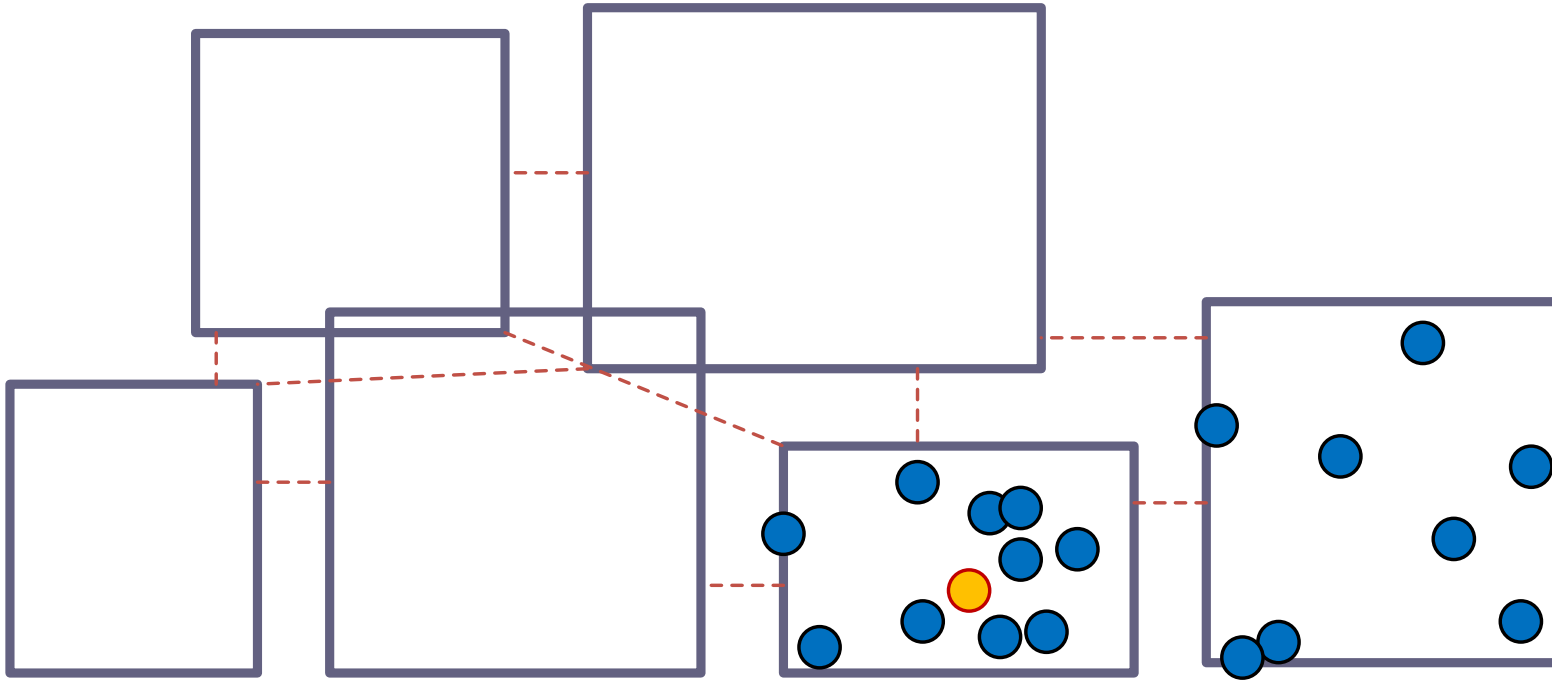
Индекс GiST



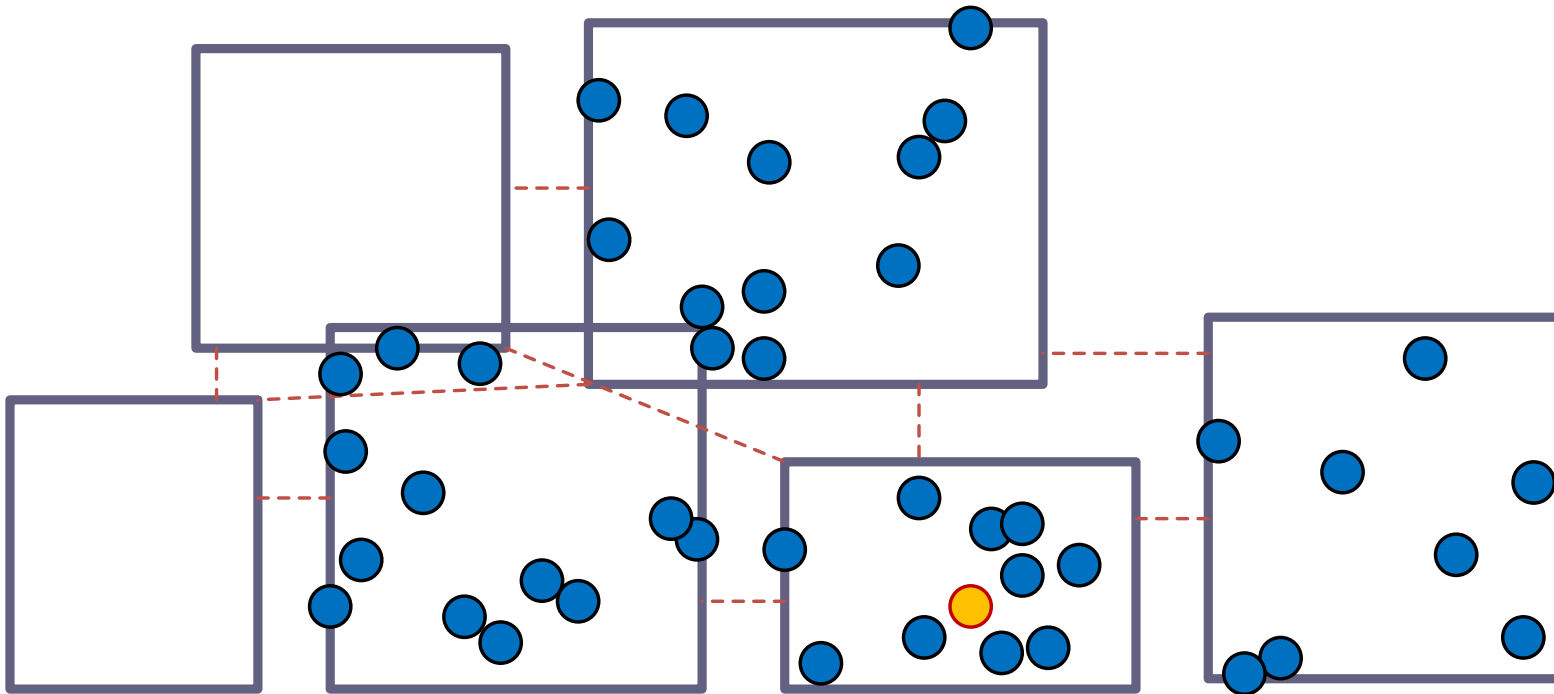
Индекс GiST



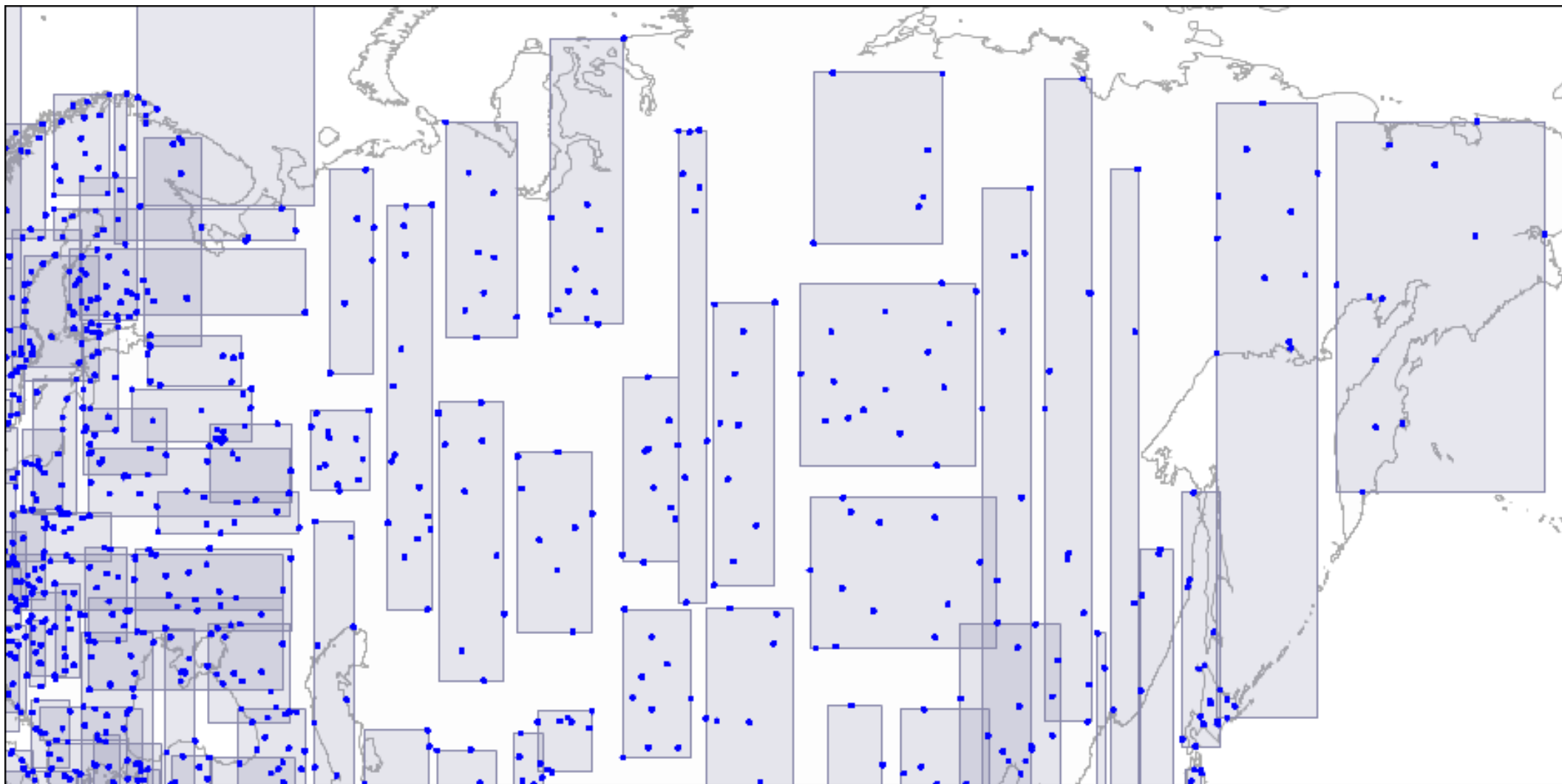
Индекс GiST



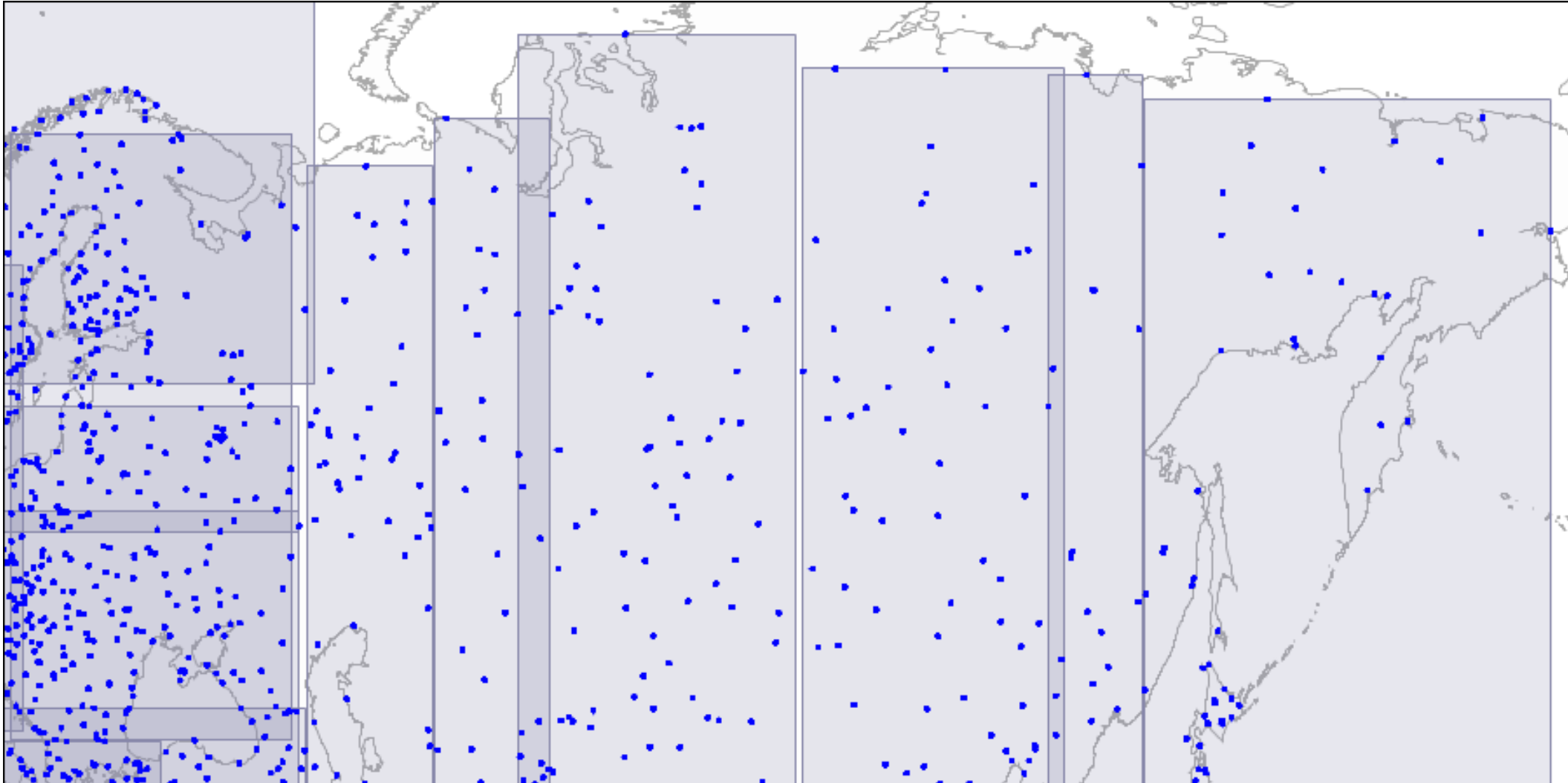
Индекс GiST



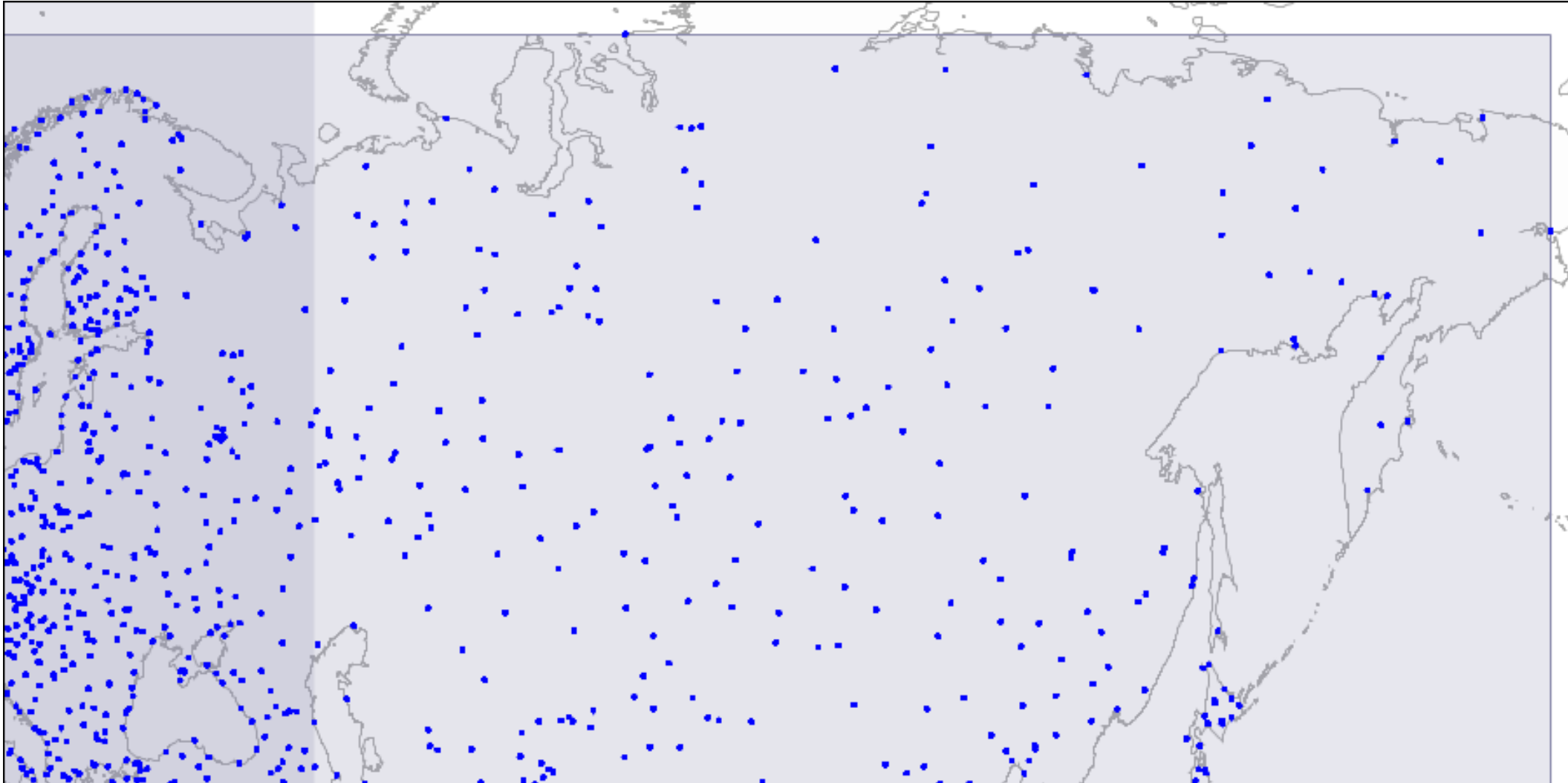
Индекс GiST



Индекс GiST



Индекс GiST



Поиск в пространстве (k-NN)

	id	amenity	coordinates
	1	shop	(64.2, 88.3)
	2	bar	(50.5, 87.4)

```
SELECT * FROM building
ORDER BY coordinates <-> point (53.7, 87.7)
LIMIT 10;
```

Индекс GiST: индексный поиск от произвольной точки

Поиск в пространстве (k-NN)

	id	amenity	coordinates
	1	shop	(64.2, 88.3)
	2	bar	(50.5, 87.4)

```
SELECT * FROM building
ORDER BY coordinates <-> point (53.7, 87.7)
LIMIT 10;
```

Индекс GiST: индексный поиск от произвольной точки

Оператор <-> не поддерживается многими ORM

k-NN

**Непрямо-
угольные
координаты**

ORM

**Индексный
поиск**

**SRID
EPSG**

LINQ to DB – расширяемая ORM

```
[Sql.Expression("{0} <-> {1}", ServerSideOnly = true)]  
public static double Distance(NpgsqlPoint p1, NpgsqlPoint p2) =>  
    throw new NotImplementedException();
```

linq2db.EntityFrameworkCore

Позволяет выполнять запросы LINQ to DB

Классы DbContext и Entities – для EF Core

Поиск в пространстве (k-NN)

```
[Sql.Expression("{0} <-> {1}", ServerSideOnly = true)]  
public static double Distance(NpgsqlPoint p1, NpgsqlPoint p2) =>  
    throw new NotImplementedException();
```

```
var myPoint = new NpgsqlPoint { X = 48, Y = 46 };
```

```
var result = await context.Buildings  
    .Select(x => new  
    {  
        Data = x,  
        Distance = PgExt.Distance(myPoint, x.Coordinates)  
    })  
    .OrderBy(x => x.Distance)  
    .Take(10)  
    .ToListAsync();
```

k-NN + ORM

id	amenity	coordinates
1	shop	(64.2, 88.3)
2	bar	(50.5, 87.4)

```
SELECT x.id, x.amenity, x.coordinates
FROM building x
ORDER BY :myPoint <-> x.coordinates
LIMIT 10;
```

Поиск в пространстве (k-NN)

Поиск от произвольной точки с поддержкой индексов

LINQ to DB: поиск в пространстве через ORM

Всё, что нужно! ... или нет?

PGCONF.RUSSIA 2021



Владимир Сурдин

МГУ

доцент

17:45

26 октября

Мифы и заблуждения о Вселенной

Мы живем в окружении мифов: постоянно заблуждаемся в отношении политики, экономики, истории своей страны и ... космоса. Попробуем вместе разобраться в наиболее острых вопросах, связанных с астрономией и космическими исследованиями. Возможно, мы лучше начнем понимать окружающий нас мир, избегать дутых сенсаций и увидим действительно интересные и привлекательные стороны научных исследований.

<https://pgconf.ru/talk/1589284>

k-NN

**Непрямо-
угольные
координаты**

ORM

**Индексный
поиск**

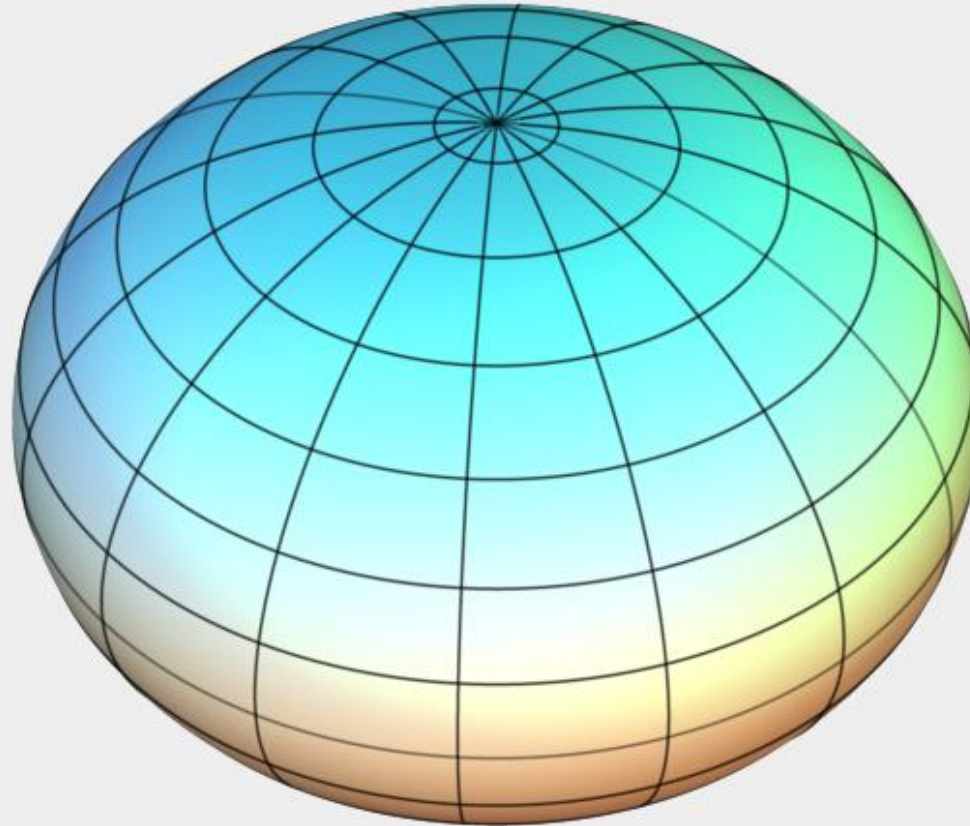
**SRID
EPSG**

Непрямоугольные координаты

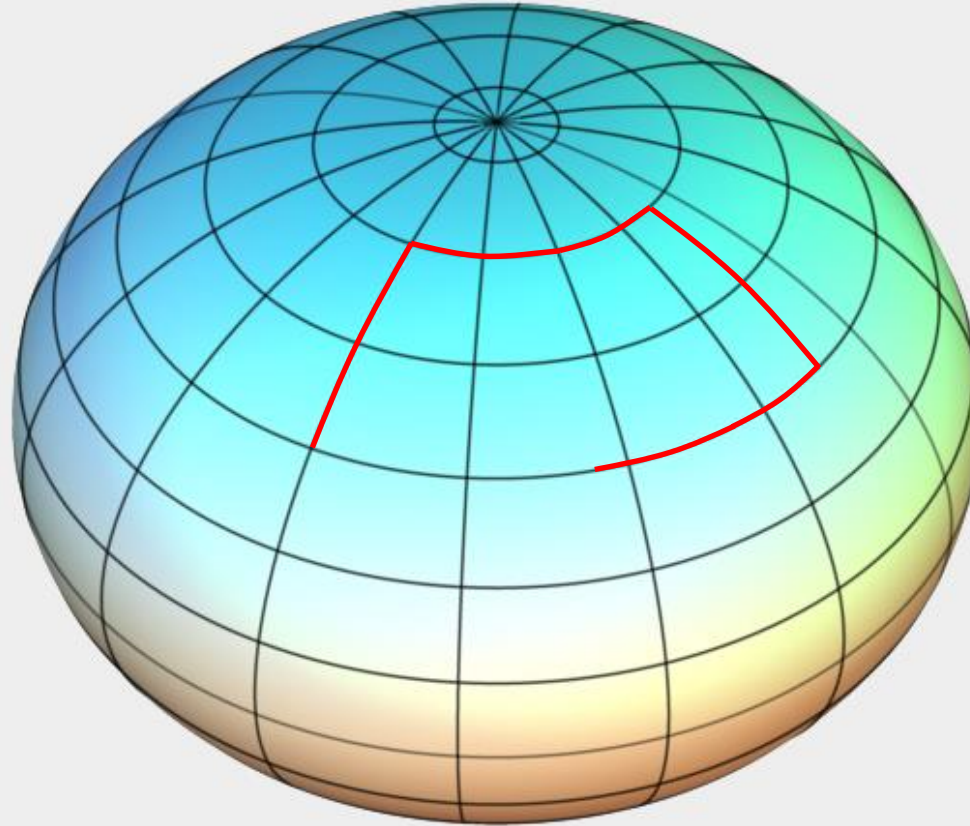
Земля - ~~шар~~

~~эллипсоид~~

геоид



Непрямоугольные координаты



PostGIS – геолокация, картография

Расширение для PostgreSQL

Открытый исходный код

Типы пространственных данных

Пространственные функции

Функция ST_Distance

Расстояние между двумя точками

```
SELECT ST_Distance(  
    ST_Transform('SRID=4326;POINT(-72.1235 42.3521)::geometry, 3857),  
    ST_Transform('SRID=4326;POINT(-72.1260 42.45)::geometry, 3857) );
```

Функция ST_Distance

Расстояние между двумя точками

```
SELECT ST_Distance(  
    ST_Transform('SRID=4326;POINT(-72.1235 42.3521)::geometry, 3857),  
    ST_Transform('SRID=4326;POINT(-72.1260 42.45)::geometry, 3857) );
```

Текстовый формат WKT (Well-known text) / OGC (Open Geospatial Consortium)

Функция ST_Distance

Расстояние между двумя точками

```
SELECT ST_Distance(  
    ST_Transform('SRID=4326;POINT(-72.1235 42.3521)'::geometry, 3857),  
    ST_Transform('SRID=4326;POINT(-72.1260 42.45)'::geometry, 3857) );
```

Текстовый формат WKT (Well-known text) / OGC (Open Geospatial Consortium)

Magic numbers

Функция ST_Distance

Расстояние между двумя точками

```
SELECT ST_Distance(  
    ST_Transform('SRID=4326;POINT(-72.1235 42.3521)::geometry, 3857),  
    ST_Transform('SRID=4326;POINT(-72.1260 42.45)::geometry, 3857) );
```

Текстовый формат WKT (Well-known text) / OGC (Open Geospatial Consortium)

Magic numbers

Где здесь k-NN?

Функция ST_Distance

Расстояние между двумя точками

```
SELECT ST_Distance(  
    ST_Transform(ST_SetSRID(ST_MakePoint(-72.1235, 42.3521), 4326), 3857),  
    ST_Transform(ST_SetSRID(ST_MakePoint(-72.1260, 42.45), 4326), 3857) );
```

~~Текстовый формат~~ функции

Magic numbers

Где здесь k-NN?

Функция ST_Distance

Расстояние между двумя точками

SELECT

```
ST_Transform(ST_SetSRID(ST_MakePoint(-72.1235, 42.3521), 4326), 3857)  
<-> ST_Transform(ST_SetSRID(ST_MakePoint(-72.1260, 42.45), 4326), 3857) );
```

Функции

Magic numbers

Оператор k-NN

k-NN

**Непрямо-
угольные
координаты**

ORM

**Индексный
поиск**

**SRID
EPSG**

Пространственные привязки

SRID (Spatial Reference System Identifiers)

SRID 4326

EPSG registry (European Petroleum Survey Group)

EPSG:3857

<https://epsg.io/?q=4326>

Функция ST_Distance

Расстояние между двумя точками

SELECT

```
ST_Transform(ST_SetSRID(ST_MakePoint(-72.1235, 42.3521), 4326), 3857)  
<-> ST_Transform(ST_SetSRID(ST_MakePoint(-72.1260, 42.45), 4326), 3857) );
```

Геодезические эллипсоидальные с/к

EPSG:4326 - WGS 84, World Geodetic System 1984, used in GPS

EPSG:4923 – ПЗ-90, Параметры Земли 1990

EPSG:7678 – ПЗ-90.02, Параметры Земли 1990.02, 2007

EPSG:7680 – ПЗ-90.11, Параметры Земли 1990.11, 2014

PostGIS: 4326 (WGS 84) и 7678 (ПЗ-90.02) эквивалентны

Функция ST_Distance

Расстояние между двумя точками

SELECT

```
ST_Transform(ST_SetSRID(ST_MakePoint(-72.1235, 42.3521), 4326), 3857)  
<-> ST_Transform(ST_SetSRID(ST_MakePoint(-72.1260, 42.45), 4326), 3857) );
```

Функция ST_Distance

Расстояние между двумя точками

SELECT

```
ST_Transform(ST_SetSRID(ST_MakePoint(-72.1235, 42.3521), 4326), 3857)  
<-> ST_Transform(ST_SetSRID(ST_MakePoint(-72.1260, 42.45), 4326), 3857) );
```

Метрические с/к

EPSG:3857 – WGS 84 / Pseudo-Mercator (Spherical Mercator)

EPSG:3395 – WGS 84 / World Mercator

EPSG:26986 – NAD83 / Massachusetts Mainland

EPSG:2163 Deprecated – US National Atlas Equal Area

EPSG:2100 – GGRS87 / Greek Grid (Transverse Mercator)

EPSG:2463 – Pulkovo 1995 / Gauss-Kruger

Расстояние Шереметьево-Пулково

По данным из Интернета: ~ 600км

EPSG:3857 – WGS 84 / Pseudo-Mercator (Spherical Mercator)

EPSG:3395 – WGS 84 / World Mercator

EPSG:26986 – NAD83 / Massachusetts Mainland

EPSG:2163 Deprecated – US National Atlas Equal Area

EPSG:2100 – GGRS87 / Greek Grid (Transverse Mercator)

EPSG:2463 – Pulkovo 1995 / Gauss-Kruger

Расстояние Шереметьево-Пулково

По данным из Интернета: ~ 600км

EPSG:3857 1130.4

EPSG:3395 1129.3

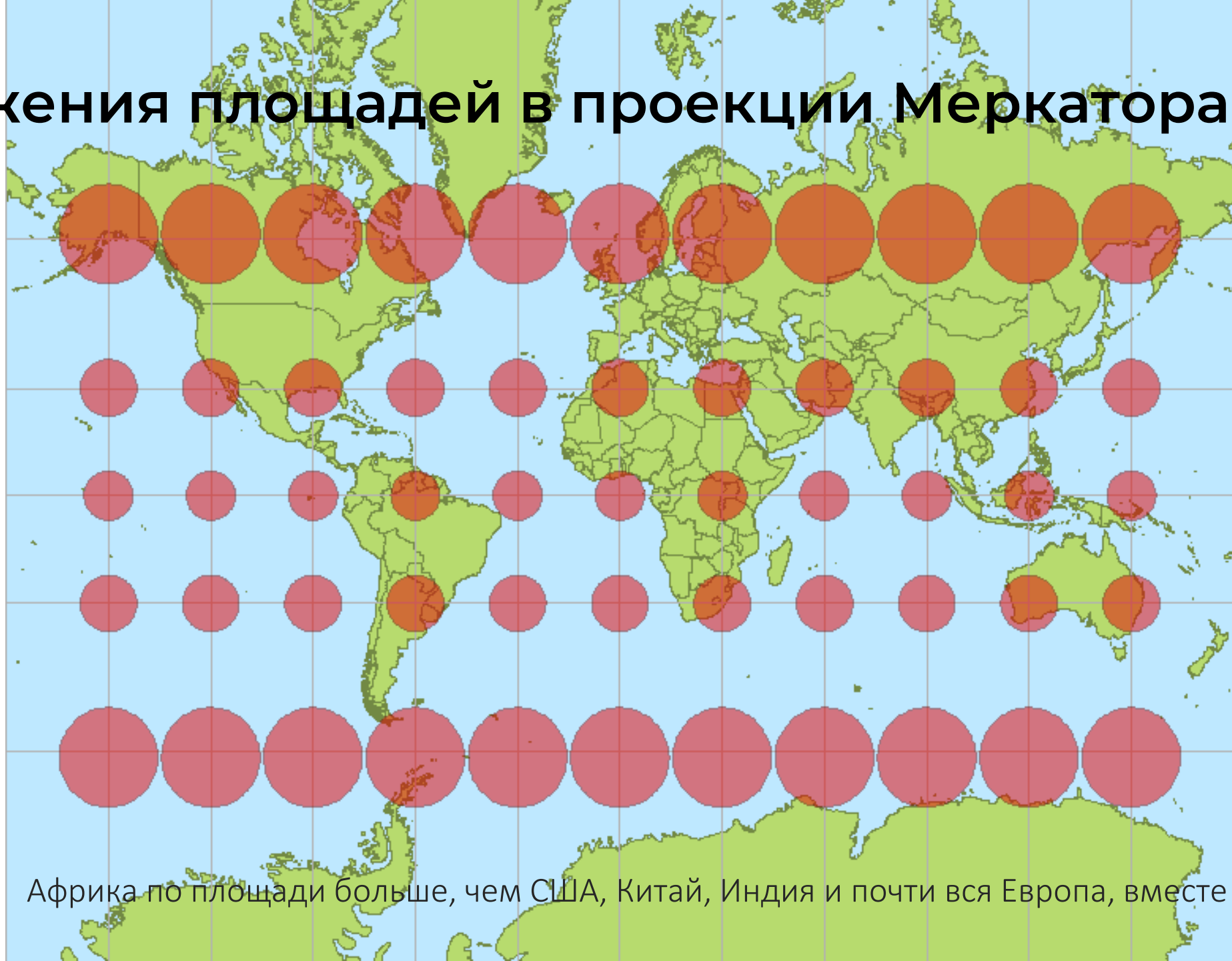
EPSG:26986 626.8

EPSG:2163 503.1

EPSG:2100 603.3

EPSG:2463 605.3

Искажения площадей в проекции Меркатора



Африка по площади больше, чем США, Китай, Индия и почти вся Европа, вместе взятые

Расстояние Шереметьево-Пулково

По данным из Интернета: ~ 600км

EPSG:26986	626.8
EPSG:2163	503.1
EPSG:2100	603.3
EPSG:2463	605.3

Расстояние Шереметьево-Пулково

По данным из Интернета: ~ 600км

EPSG:2100	603.3	Универсальная поперечная проекция Меркатора
EPSG:2463	605.3	Проекция Гаусса — Крюгера

Тип geography

Расстояние между двумя точками

```
SELECT ST_Distance(  
    ST_SetSRID(ST_MakePoint(37.4221, 55.9744), 4326)::geography,  
    ST_SetSRID(ST_MakePoint(30.2649, 59.7994), 4326)::geography, true);
```

Тип geography

Расстояние между двумя точками

```
SELECT ST_Distance(  
    ST_SetSRID(ST_MakePoint(37.4221, 55.9744), 4326)::geography,  
    ST_SetSRID(ST_MakePoint(30.2649, 59.7994), 4326)::geography, :spheroid );
```

Эллипсоид	600.8
Сфера	599.3

Тип geography

Расстояние между двумя точками

```
SELECT (ST_SetSRID(ST_MakePoint(37.4221,55.9744), 4326)::geography)  
    <-> (ST_SetSRID(ST_MakePoint(30.2649,59.7994), 4326)::geography);
```

Эллипсоид 600.8

Сфера 599.3 Поддержка k-NN

Расстояние Шереметьево-Пулково

Тип geometry:

EPSG:2100	603.3	Универсальная поперечная проекция Меркатора
EPSG:2463	605.3	Проекция Гаусса — Крюгера

Тип geography:

Эллипсоид	600.8
Сфера	599.3

Расстояние Шереметьево-Пулково

Тип geometry:

EPSG:2100	603.3	Универсальная поперечная проекция Меркатора
EPSG:2463	605.3	Проекция Гаусса — Крюгера

Тип geography:

Эллипсоид	600.8
Сфера	599.3

Все системы неточные?

Насколько точно определены большие расстояния?

Не учтён рельеф

...

Какую задачу решаем?

Где находится ближайший бар

Пример 1:

- 600.3 км на юг
- 600.8 км на запад

Пример 2:

- 32 м 47 см на юг
- 32 м 51 см на восток

Расстояние на Земле, k-NN

Координаты lat/lon:

EPSG:4923 – ПЗ-90

EPSG:4326 - WGS 84

EPSG:7678 – ПЗ-90.02

EPSG:7680 – ПЗ-90.11



Тип geometry:

EPSG:2100, Универсальная поперечная проекция Меркатора

EPSG:2463, Проекция Гаусса — Крюгера

Тип geography (сфера)

Берём любую

Берём любую

Типы geometry и geography

```
SELECT ST_SetSRID(ST_MakePoint(37.4221,55.9744), 7680);
```

```
0101000020001E000020D26F5F07B64240789CA223B9FC4B40
```

Типы geometry и geography

```
WITH s1 AS  
  (SELECT ST_SetSRID(ST_MakePoint(37.4221,55.9744), 7680) AS p)  
SELECT ST_x(p), ST_y(p), ST_SRID(p) FROM s1;
```

```
37.4221  55.9744  7680
```

Типы geometry и geography

```
WITH s1 AS  
  (SELECT ST_Transform(ST_SetSRID(ST_MakePoint(37.4221,55.9744), 7680),  
    2100) AS p)  
SELECT ST_x(p), ST_y(p), ST_SRID(p) FROM s1;
```

```
1334739.9136771765  6284871.696198237  2100
```

Типы geometry и geography

```
WITH s1 AS  
  (SELECT ST_SetSRID(ST_MakePoint(37.4221,55.9744), 7680)::geography AS p)  
SELECT p, ST_x(p::geometry), ST_y(p::geometry), ST_SRID(p) FROM s1;
```

```
37.4221  55.9744  7680
```

k-NN

**Непрямо-
угольные
координаты**

ORM

**Индексный
поиск**

**SRID
EPSG**

Индекс для k-NN

Индекс по выражению

Вычисляемый столбец

Индекс для k-NN

Исходные координаты: point (Postgres)

Вычисляемый столбец: point -> geometry/geography

Индекс по вычисляемому столбцу

Индекс для k-NN

```
CREATE TABLE building
(
  id serial PRIMARY KEY NOT NULL,
  coordinates point NOT NULL,
  comment text,
  epsg geometry GENERATED ALWAYS AS ST_Transform(ST_SetSRID(ST_MakePoint(
    coordinates[0], coordinates[1]), 4326), 2100) STORED,
  geo geography GENERATED ALWAYS AS (ST_SetSRID(ST_MakePoint(
    coordinates[0], coordinates[1]), 4326))::geography STORED
);
```

```
CREATE INDEX ON building USING GIST (epsg);
CREATE INDEX ON building USING GIST (geo);
```

Исходные координаты: point

```
SELECT id, comment, coordinates FROM building;
```

	id	comment	coordinates
	1	Аэропорт	(37.4221,55.9744)
	2	Бар	(50.5142,87.4356)

k-NN

**Непрямо-
угольные
координаты**

ORM

**Индексный
поиск**

**SRID
EPSG**

ORM

NetTopologySuite (NTS) – библиотека пространственных типов для .NET

Npgsql.EntityFrameworkCore.PostgreSQL.NetTopologySuite

LinqToDBPostGisNTS

ORM

```
[Table("building", Schema="public")]
public class Building
{
    [Column("id")]
    [PrimaryKey]
    [Identity]
    public int Id { get; set; }

    [Column("coordinates")]
    public NpgsqlPoint Coordinates { get; set; }

    [Column("epsg")]
    public Point EpsgGeometry { get; set; }

    [Column(Name = "geo", DbType = "geography")]
    public Point Geography { get; set; }
}
```

LINQ to DB – пишем расширение

```
[Sql.Expression("{0} <-> {1}", ServerSideOnly = true)]  
public static double STDistanceKnn([ExprParameter] this Geometry point1,  
    [ExprParameter] Geometry point2) =>  
    throw new NotImplementedException();
```

```
[Sql.Expression("({0}::geography)", ServerSideOnly = true)]  
public static Point ToGeography([ExprParameter] this Geometry point) =>  
    throw new NotImplementedException();
```

ORM, k-NN, geometry

```
var coordinate = new Coordinate(30.2649, 59.7994);
var point1 = new Point(coordinate);

var buildings = await context.Buildings
    .Select(b => new
    {
        Distance = b.EpsgGeometry.STDistanceKnn(point1
            .STSetSrId(Srid).STTransform(Epsg)),
        Building = b
    })
    .OrderBy(x => x.Distance)
    .Take(10)
    .ToListAsync();
```

ORM, k-NN, geometgry

SELECT

```
b.epsg <-> ST_Transform(ST_SetSRID(:point1, 4326), 2100),  
b.id,  
b.coordinates,  
b.epsg,  
b.geo
```

FROM

```
"public".building b
```

ORDER BY

```
b.epsg <-> ST_Transform(ST_SetSRID(:point1, 4326), 2100)
```

LIMIT :take

ORM, k-NN, geography

```
var coordinate = new Coordinate(30.2649, 59.7994);
var point1 = new Point(coordinate);

var buildings = await context.Buildings
    .Select(b => new
    {
        Distance = b.Geography.STDistanceKnn(
            point1.STSetSrid(Srid).ToGeography()),
        Building = b
    })
    .OrderBy(x => x.Distance)
    .Take(10)
    .ToListAsync();
```

ORM, k-NN, geography

SELECT

```
b.geo <-> (ST_SetSRID(:point1, 4326)::geography),  
b.id,  
b.coordinates,  
b.epsg,  
b.geo
```

FROM

```
"public".building b
```

ORDER BY

```
b.geo <-> (ST_SetSRID(:point1, 4326)::geography)
```

LIMIT :take

Расстояние на Земле, PostGIS

Поддержка k-NN

Работает через ORM (LINQ to DB)

Итоги

Как найти ближайший бар с помощью Postgres?

Алгоритм k-NN (поддержка индексного поиска)

Правда ли, что Земля не плоская?

Правда. Почти эллипсоид (шар – тоже пойдёт)

Полезное

Алгоритм k-NN

Прямоугольные координаты и не только

Индекс GiST, поддержка оператора <->

PostGIS - Расстояния на Земном шаре/эллипсоиде

ORM: пространственные типы + k-NN

На примере LINQ to DB

Пространственные привязки (SRID/EPSG)

Слайд 67

Docker-образы PLV8

plv8.com

260K скачиваний

A group of four people are walking away from the camera on a wide, snow-covered path. The person in the foreground is wearing a dark coat and has their arms outstretched. The path leads towards a dense forest of bare trees under a clear sky. A tall, thin structure is visible in the distance on the right.

КОНЕЦ

Ссылки

PostGIS

<https://postgis.net>

Реестр EPSG

<https://epsg.io>

Библиотеки для .NET

<https://github.com/linq2db/linq2db>

<https://github.com/apdevelop/linq2db-postgis-extensions>

Документация PostgreSQL на русском

<https://postgrespro.ru/docs>

Мои контакты

fadeevas@sibedge.com

<https://vk.com/fadeev>