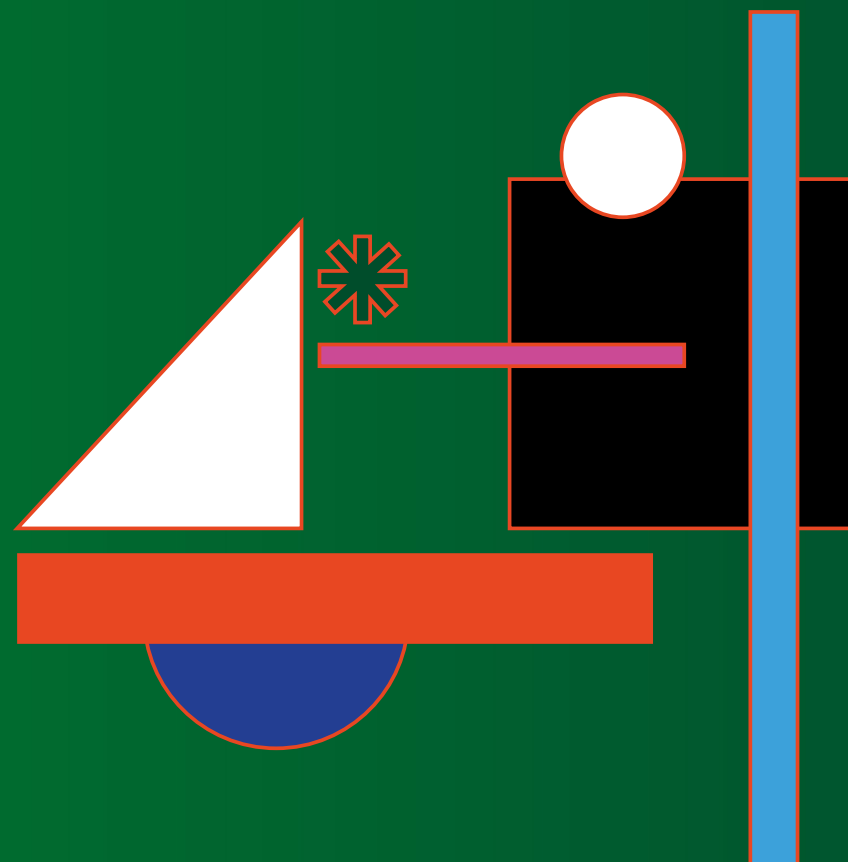


PGCONF.RUSSIA 2024

Сферические
данные в вакууме
сегодня

Иван Панченко

Виталий Давыдов



О чем мы поговорим?

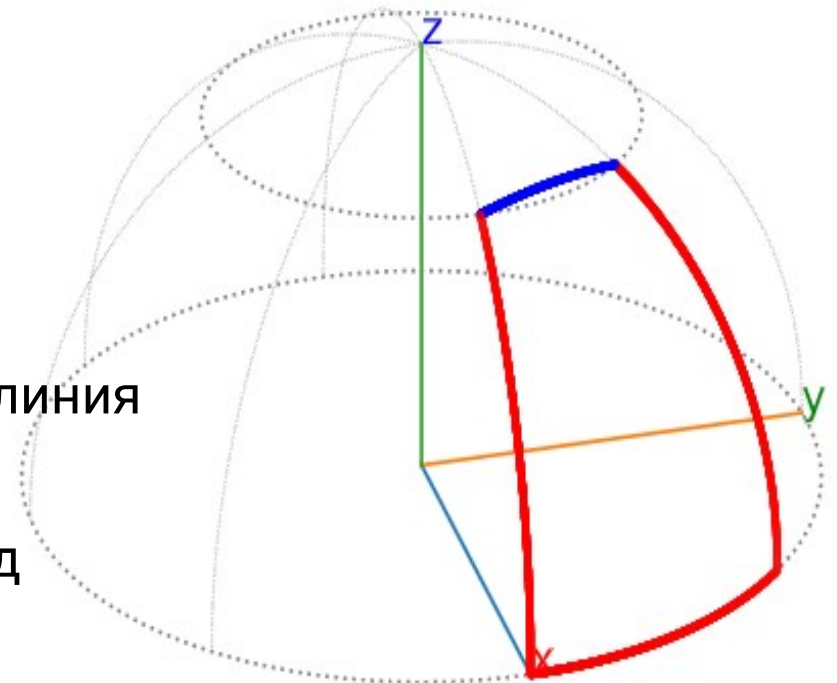
- Освежим знания по плоской и сферической геометрии
- Поговорим о стандартах
- Поймем, какие геометрические задачи стоят перед СУБД
- Обсудим существующие решения, особенности и области применения
- Вспомним, что Земля – не идеальная сфера :)

Кратко о геометрических фигурах

- Точка, линия, треугольник, кривая, многоугольник (полигон)...
- Размерность пространства, в котором определена фигура (1d, 2d, 3d)
- Размерность фигуры (0 – точка, 1 – линия, 2 – поверхность, 3 – тело)
 - С точки зрения аналитической геометрии размерность фигуры равна числу координат, нужных для определения положения лежащей на этой фигуре точки¹.
- У фигуры есть внутренность (interior), граница (boundary), внешняя часть (exterior)
 - У точки нет границы, граница сегмента линии – две крайние точки
- Отношения между фигурами: intersects, within, crosses, touches и др
- Операции над фигурами: пересечение, объединение, исключение

Сфера

- Сфе́ра (от греч. σφαῖρα – шар, мяч), замкнутая поверхность, все точки которой одинаково удалены от одной точки, называемой центром сферы ¹
- Двумерный объект в 3-х мерном пространстве
- Сфера – искривленная двумерная поверхность
- Большой круг – сечение сферы плоскостью, проходящей через центр сферы
- Геодезическая линия или ортодромия – кратчайшая линия между двум точками, лежит на дуге большого круга.
- Локсодромия – линия, пересекающая меридианы под одинаковыми углами



¹ Большая российская энциклопедия

Углы и расстояния на сфере

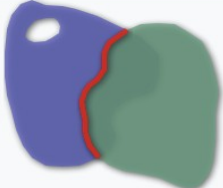
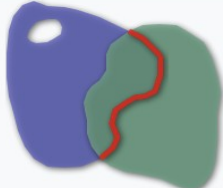
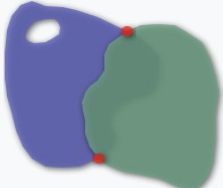
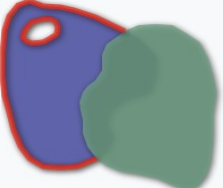
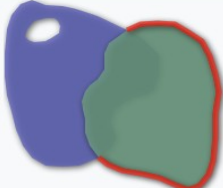
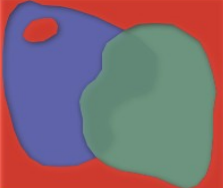
- Абсолютные координаты
 - Декартовы координаты в трехмерном пространстве (x, y, z)
 - Полярные координаты (R, Θ, φ)
 - Географические координаты (широта и долгота – λ, φ)
 - Небесные координаты (прямое восхождение и склонение – α, δ)
- Угловые расстояния
 - Длина дуги большого круга между точками
- Линейные расстояния на поверхности сферы
 - Километры, мили, метры – пропорциональны угловым

Стандарты и модели

- Open Geospatial Consortium (OGC)
 - Стандарт: Simple Feature Access (SFA) Part 1: Common Architecture
 - Стандарт: SFA Part 2: SQL option
- Топологическая модель DE-9IM (в пространстве R^2)
- ISO/IEC 13249-3 SQL Multimedia - Spatial

Модель DE-9IM

- Определяет отношения между фигурами в R^2
- Отношение между двумя фигурами задается в виде матрицы 3x3
- OSG стандартизовал пространственные предикаты (contains, crosses, intersects, touches...), а также функцию (ST_Relate), возвращающую матрицу DE-9IM в виде кода
- Код DE9-IM: 0F1F00102 – intersects & crosses & overlaps
- Применяется в PostGIS

 $\dim[I(a) \cap I(b)] = 2$	 $\dim[I(a) \cap B(b)] = 1$	 $\dim[I(a) \cap E(b)] = 2$
 $\dim[B(a) \cap I(b)] = 1$	 $\dim[B(a) \cap B(b)] = 0$	 $\dim[B(a) \cap E(b)] = 1$
 $\dim[E(a) \cap I(b)] = 2$	 $\dim[E(a) \cap B(b)] = 1$	 $\dim[E(a) \cap E(b)] = 2$

I – interior
B – boundary
E – exterior

<https://en.wikipedia.org/wiki/DE-9IM>

Существующие решения (open source)

- Встроенные типы PostgreSQL (point, line, lseg, box, path, polygon...)
- EarthDistance (расчет расстояний на сфере)
- Q3C (индексация на сфере в астрономических координатах ra, dec)
- PgSphere (геометрические типы, операции, индексация)
- PostGIS (расстояния на поверхности Земли)

Встроенные типы PostgreSQL

- **Объекты на плоскости:**

point, line, lseg, box, path, polygon, circle

- **Операции:**

- Определение свойств фигур (длина, площадь, центр...)
- Преобразования фигур (например, box → polygon)
- Операторы:
 - Объединение фигур, пересечение, расстояние, ближайшая точка,
- Поиск в боксе (или в другой фигуре)

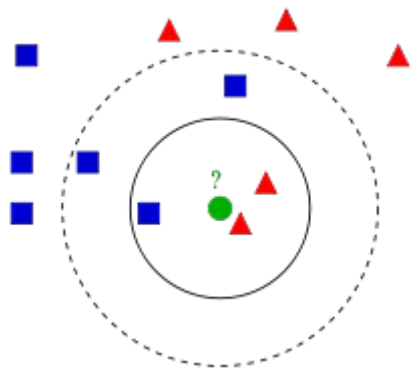
```
SELECT * FROM airports WHERE coordinates <@ box(' (36,55), (38,57) ');
```

- KNN-поиск по индексу

```
SELECT * FROM airports ORDER BY coordinates <-> point(37,56) LIMIT 10;
```

Поиск kNN

Заранее неизвестно, насколько близок искомый «ближайший» объект к данному.
Поэтому просто так индексы не помогут.



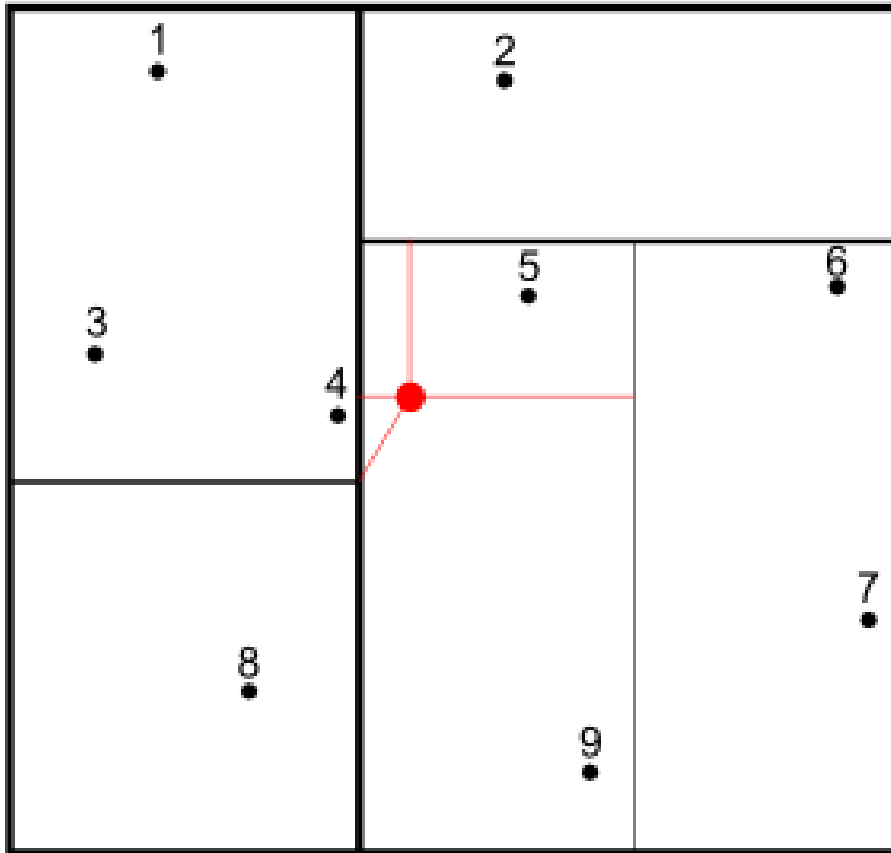
В индексе есть информация о близости, надо её использовать.



Поиск kNN : Алгоритм

- При спуске по дереву сохраняем пройденные узлы в форме очереди приоритетов по расстоянию от искомой точки
- Расстояние от точки до бокса определим как минимальное расстояние от неё до него, и ноль, если точка внутри
- Дойдя до точки, разворачиваем очередь в обратном порядке. Натыкаясь на прямоугольник, применяем алгоритм к нему.
- Получаем упорядоченный список соседей

Поиск kNN : Пример

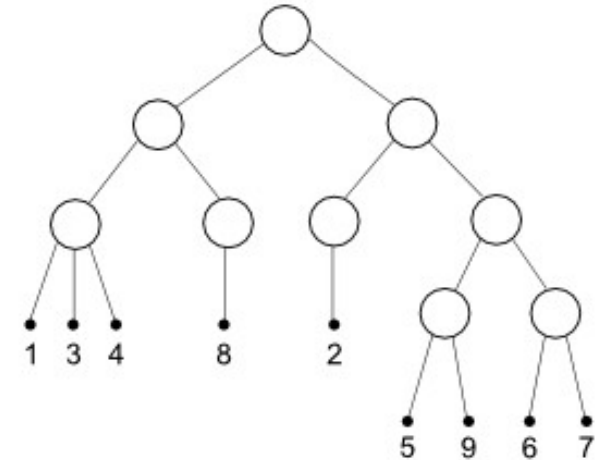


Очередь приоритетов

- 1: {1,2,3,4,5,6,7,8,9}
- 2: {2,5,6,7,9}, {1,3,4,8}
- 3: {5,6,7,9}, {1,3,4,8}, {2}
- 4: {5,9}, {1,3,4,8}, {2}, {6,7}
- 5: {1,3,4,8}, 5, {2}, {6,7}, 9
- 6: {1,3,4}, {8}, 5, {2}, {6,7}, 9
- 7: **4**, {8}, 5, {2}, {6,7}, 3, 1, 9

4 – ближайшая точка!

- 8: **5**, {2}, {6,7}, 3, 8, 1, 9
- 9: {6,7}, 3, 2, 8, 1, 9
- 10: 3, 2, 8, 1, 9, 6, 7



EarthDistance

- Штатное расширение PostgreSQL
- Расчет расстояний на идеальной сфере
- Линейное расстояние на поверхности сферы (в милях)
- Point-based расчет расстояний (тип point – lng, lat)

`point <@> point → float8 (мили)`

- Cube-based

`earth_distance ( earth, earth ) → float8 (метры)`

- Требуется расширение cube (тоже штатное)

<https://postgrespro.ru/docs/postgresql/16/earthdistance>

Особенности EarthDistance

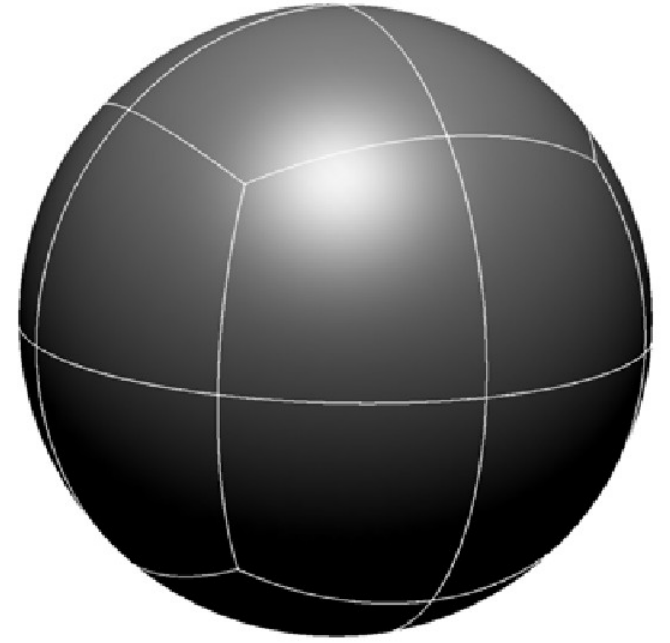
```
SELECT point(0,90) <@>  
point(90,90);  
3.428101559778626e-13
```

```
SELECT earth_distance(  
    ll_to_earth(90,90),  
    ll_to_earth(90,0));  
5.523213207287424e-10
```

```
SELECT earth();  
6378168
```

Quad Tree Cube (Q3C)

- Работает с астрономическими координатами: ra, dec и встроенными типами PostgreSQL
- Партицирование: куб, вписанный в сферу, далее дерево квадрантов (quad tree) на каждой лицевой части (проекция точек сферы на грань квадрата)
- Поиск и индексация на сфере (ra, dec)
- Стороннее расширение (Сергей Копосов, Олег Бартунов)
<https://github.com/segasai/q3c>



pgSphere

- Углы (небесные координаты) и угловые расстояния
- Относительная точность: 1E-9 (можно изменить в исходниках)
- Своя система типов (используются углы Эйлера)
 - Особый способ хранения объектов
- Ограниченный набор геометрических операций над типами
- Геометрические операции НЕ соответствуют DE-9IM (!)
- Индексы (сферический, MOC/HEALPIX, hash)
- Используется в астрономии
- Стороннее расширение (К. Берг, В. Давыдов и др.)

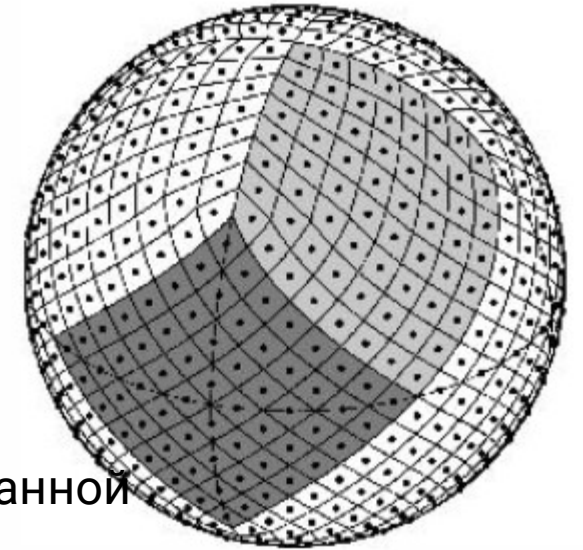
<https://github.com/postgrespro/pgsphere>

pgSphere: Типы

Тип	Представление	Описание
spoint	float8, float8	Точка на сфере
seuler	φ , θ , Ψ	Преобразование Эйлера
sline	φ , θ , Ψ , length	Сегмент линии
scircle	spoint, radius	Круг
sellipse	spoint, minrad, maxrad, incl	Эллипс
sbox	spoint, spoint	Прямоугольник
spath	spoint[]	Путь из линейных сегментов
spoly	spoint[]	Полигон
smoc		Используется в индексации SMOС/HEALPIX

HEALPix Multi-Order Coverage Map (MOC)

- HEALPix - Hierarchical Equal Area isoLatitude Pixelization of a sphere
- Проекция, отображающая сферу на плоскость
- Используется в астрономии (например, в создании карт плотности космического микроволнового фона)
- Каждую область можно пронумеровать.
- Тип SMOC позволяет закодировать произвольную область на сфере с заданной точностью.
- Вся сфера: `SELECT smoc '0/0-11'` – указаны области от 0 до 11-й на нулевом уровне иерархии.
- Диск около точки (0, 0) with radius 1 на уровне 2:
 - `SELECT smoc '0/4 1/0 2 12-13 34-35 45 47 2/4 6 12 14 56-57 60-61 88 90-91 116-117 119 130-131 134-135 177 179 185 187';`



Сайт: <https://healpix.sourceforge.io/>

PostGIS

- Линейные расстояния на поверхности эллипсоида
- Расширение для работы с географическими данными
 - Тип `geometric` – проекция на плоскость, алгоритмы на 2d плоскости
 - Тип `geographic` – сферические координаты
- Следует стандартам OGC SFA
- Геометрические типы (`point`, `linestring`, `polygon`, `collections`)
- Пространственные предикаты и операторы (DE-9IM)
- Индексация (R-tree over GiST)
- Функции обработки геометрических данных (упрощение, преобразование, ...)
- Сертифицирован организацией OGC
- <http://postgis.net>

Эксперимент!

- Таблица с 50М равномерно распределенными по lng, lat точками
- На сфере точки на полюсах расположены плотнее, чем на экваторе
- Процессор: AMD Ryzen 9 7950x3d, ОЗУ: 64GB, SSD
- Запросы:
 - Выборка в заданном радиусе (radius/cone search)
 - Поиск ближайших соседей (KNN)

pgSphere: Запросы

SELECT COUNT(*) FROM t WHERE '(0,0)::spoint <-> pos < 0.0174533		3.2 сек
SELECT COUNT(*) FROM t WHERE '<(0,0),1d>::scircle @> pos	(экватор)	1 мс (2k) ¹
SELECT COUNT(*) FROM t WHERE '<(0,90d),1d>::scircle @> pos	(полюс)	56 мс (280k)
SELECT COUNT(*) FROM t WHERE '<(0,45d),1d>::scircle @> pos		1.6 мс (4k)
SELECT * FROM t ORDER BY pos <-> '(0,0)::spoint LIMIT 10	(KNN)	1 мс
SELECT * FROM t ORDER BY pos <-> '(0,90d)::spoint LIMIT 10	(KNN, полюс)	1 мс

¹ В скобках указано к-во строк в результате поиска

Q3C: Запросы

SELECT COUNT(*) FROM t WHERE q3c_radial_query(lng, lat, 0, 0, 1)	3.8 мс (2k)
SELECT COUNT(*) FROM t WHERE q3c_radial_query(lng, lat, 0, 90, 1)	40 мс (280k)
SELECT * FROM t ORDER BY q3c_dist(lng, lat, 0, 0) LIMIT 10	5 сек

PostGIS: Запросы

SELECT COUNT(*) FROM t WHERE ST_DWithin(pos, 'POINT(0 0)', 1)	2 mc (2k)
SELECT COUNT(*) FROM t WHERE ST_DWithin(pos, 'POINT(0 90)', 1)	2 mc (2k)
SELECT t.pos <-> 'POINT(0 0)::geometry as dist FROM t ORDER BY dist LIMIT 10	0.5 mc
SELECT t.pos <-> 'POINT(0 90)::geometry as dist FROM t ORDER BY dist LIMIT 10	0.5 mc

Сравнительная таблица

Тест	pgSphere	Q3C	PostGIS
Поиск в радиусе (экватор)	1 мс (2k)	4 мс (2k)	2 мс (2k)
Поиск в радиусе (полюс)	56 мс (280k)	40 мс (280k)	2 мс (2k)
Поиск ближайших соседей (KNN)	1 мс	4.8 сек	0.5 мс

Достоинства и недостатки

pgSphere

Углы и угловые расстояния
Своя система типов
Особый способ хранения линий sline

Ограниченное число операций и предикатов
Ограниченная точность (1E-9, можно изменить)
Не соответствует модели DE-9IM

Q3C

Углы и угловые расстояния
Быстрый поиск в заданном радиусе
Работает с float8 или point

Медленный KNN (индекс не применяется)
Нет операций для работы с фигурами на сфере

PostGIS

Линейные расстояния на эллипсоиде
Своя система типов
Развитая функциональность
Следует стандартам OGC SFA (DE-9IM)

Слабо применим для работы с угловыми расстояниями
Некорректно работает на полюсах (тип geometry)

Спасибо!

