FROM MAP TO REALITY: USING POSTGIS IN WARFARE

Taras Kloba

ABOUT ME

Taras Kloba

Associate Director,
Big Data & Analytics,
SoftServe Inc.

- Founded a volunteering IT group for Ukrainian army systems, winner of the TIDE NATO Hackathon and Ukraine Defence Hackathon.
- More than 14 years of experience in Data Engineering
- Co-leader of PostgreSQL Ukraine and Big Data Community
- Ph.D. in Economics
- Winner of the Ukrainian IT Awards 2019 in Software Architecture
- Certified Cloud Architect & Data Engineer on Google Cloud, Microsoft, and Amazon Web Services. Microsoft Certified Trainer
- Father of three daughters

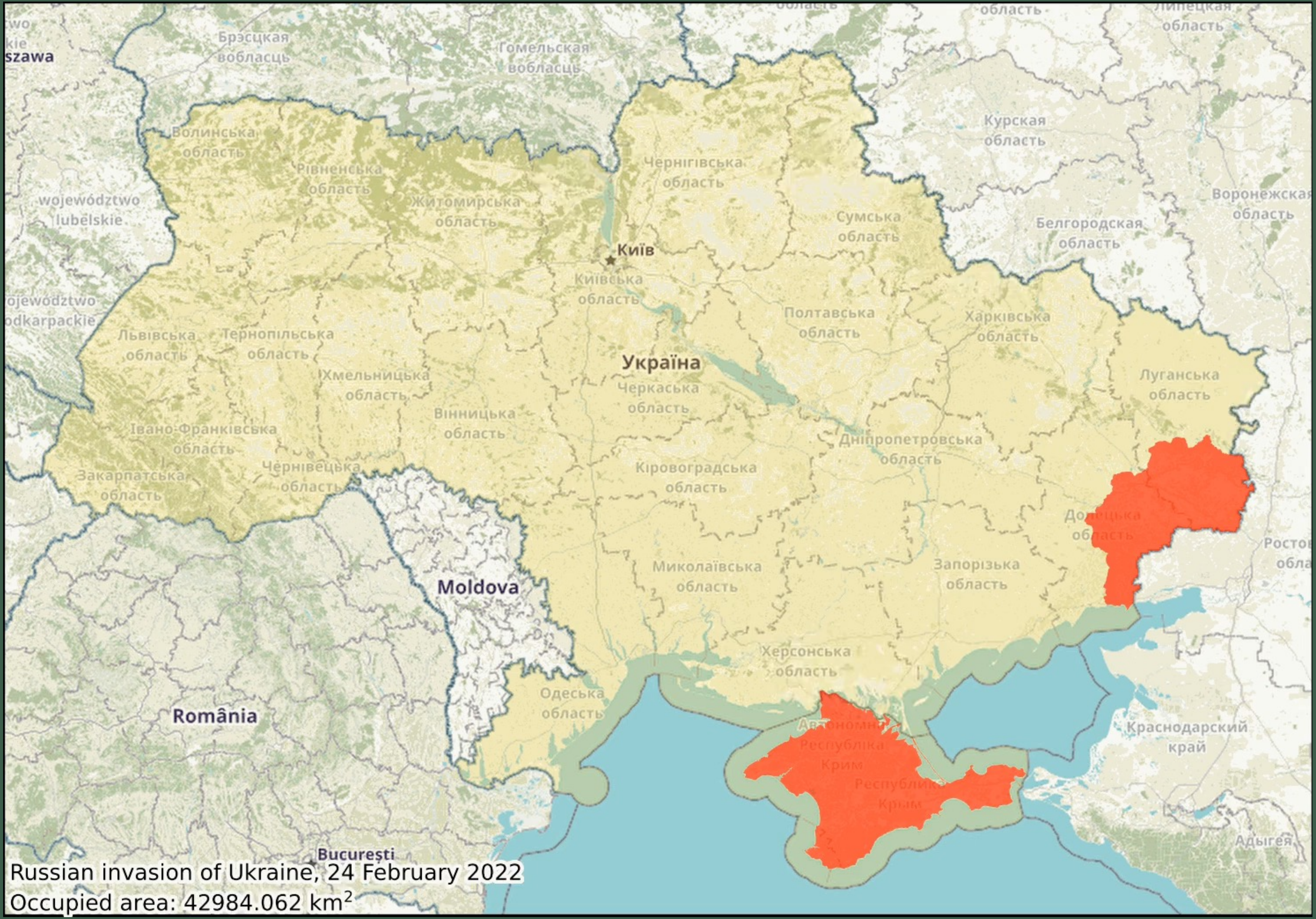
THE DAVID AND GOLIATH STORY

Created by https://openai.com/dall-e-3

UKRAINE'S RESISTANCE IN THE ONGOING WAR

Source: https://en.wikipedia.org/wiki/Russia%E2%80%93Ukraine_relations

VISUALIZATION OF UKRAINE'S RESISTANCE



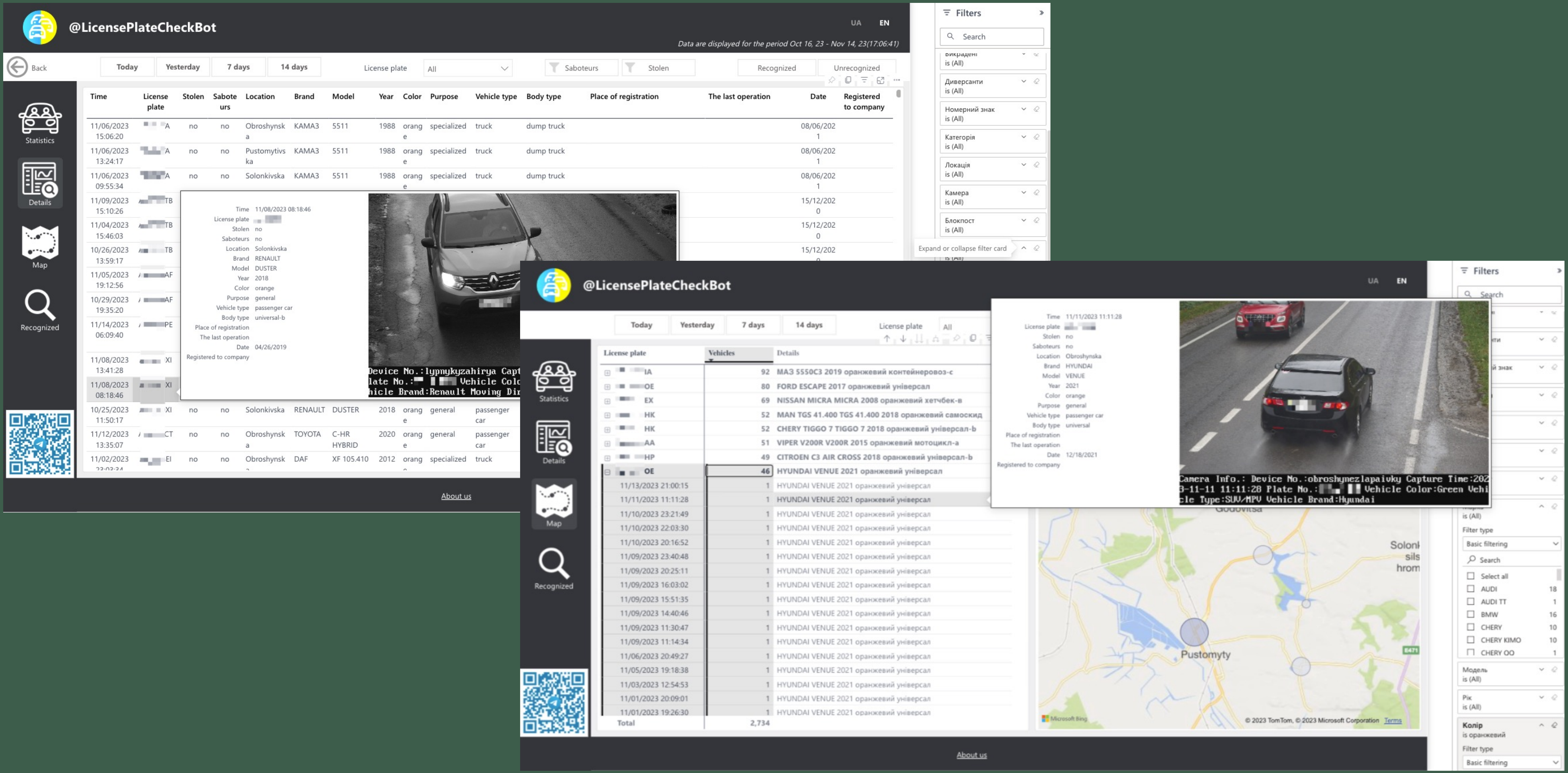
Data source: https://liveuamap.com/ Visualization authors: Kyrylo Kotelevets and Taras Kloba

DATA SOURCES

NOT EVERYBODY CAN BE A SOLDIER.
BUT EVERYBODY CAN BE A PART OF RESISTANCE.

Source: “The First Code” movie https://thefirstcode.com/en/

LICENSEPLATEBOT



Landing page: https://platerecognition.softserveinc.com/?hs_preview=HYIqZvNg-133931468450

AUTOMATED RECOGNITION OF MILITARY VEHICLES

Full video: https://youtube.com/watch?v=rqzIUtJcZBs

AI-ASSISTED UAV SURVEILLANCE

Full video: https://youtube.com/watch?v=rqzIUtJcZBs

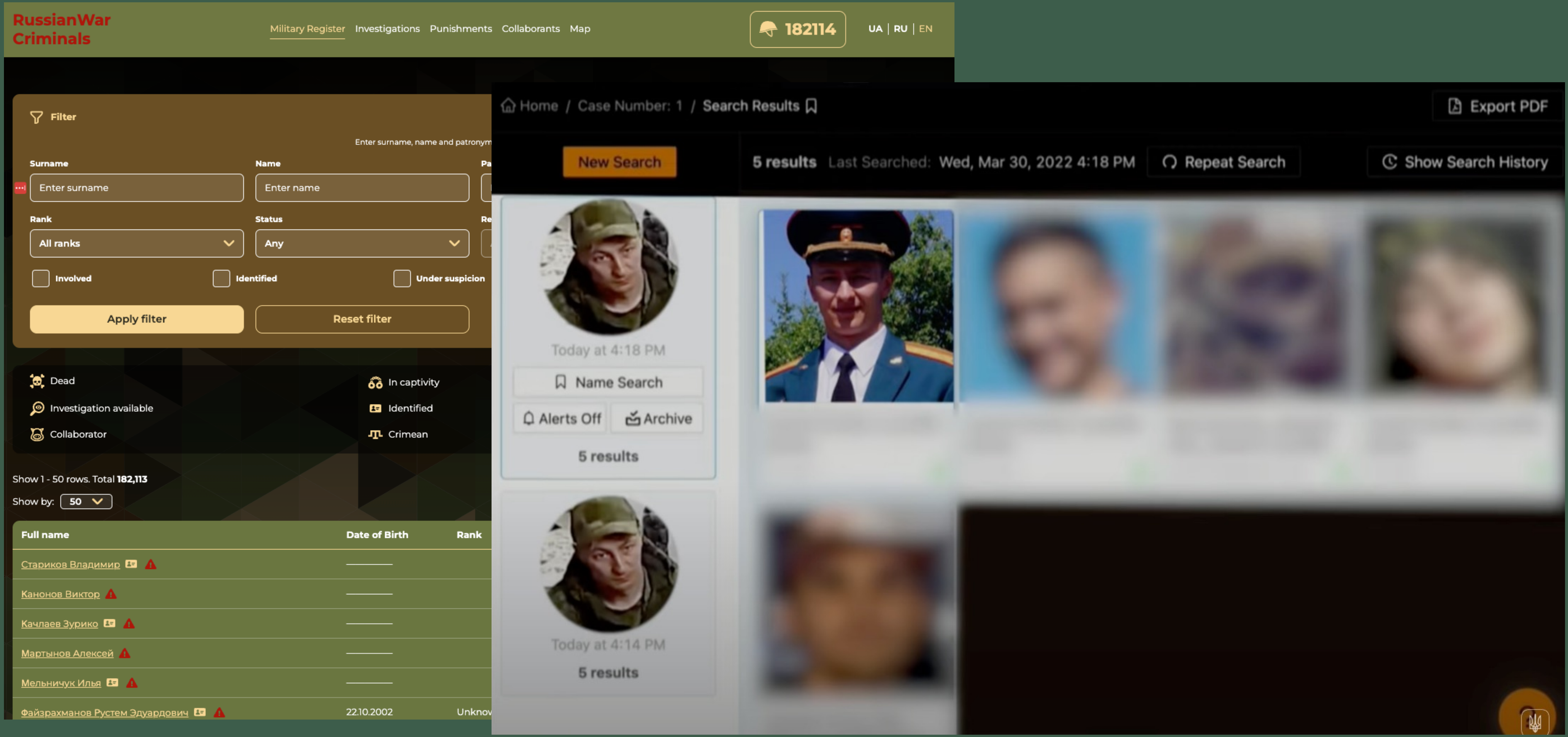
AI-ENABLED DETECTION OF MILITARY AIRCRAFTS

Source: https://texty.org.ua/fragments/107460/aerodrom-bilya-sak-do-i-pislya-udaru-suputnykovi-znimky/

UKRAINIAN AI-PROJECT ZVOOK

Source: https://www.pravda.com.ua/eng/articles/2023/03/1/7391493/

DATA LEAKS



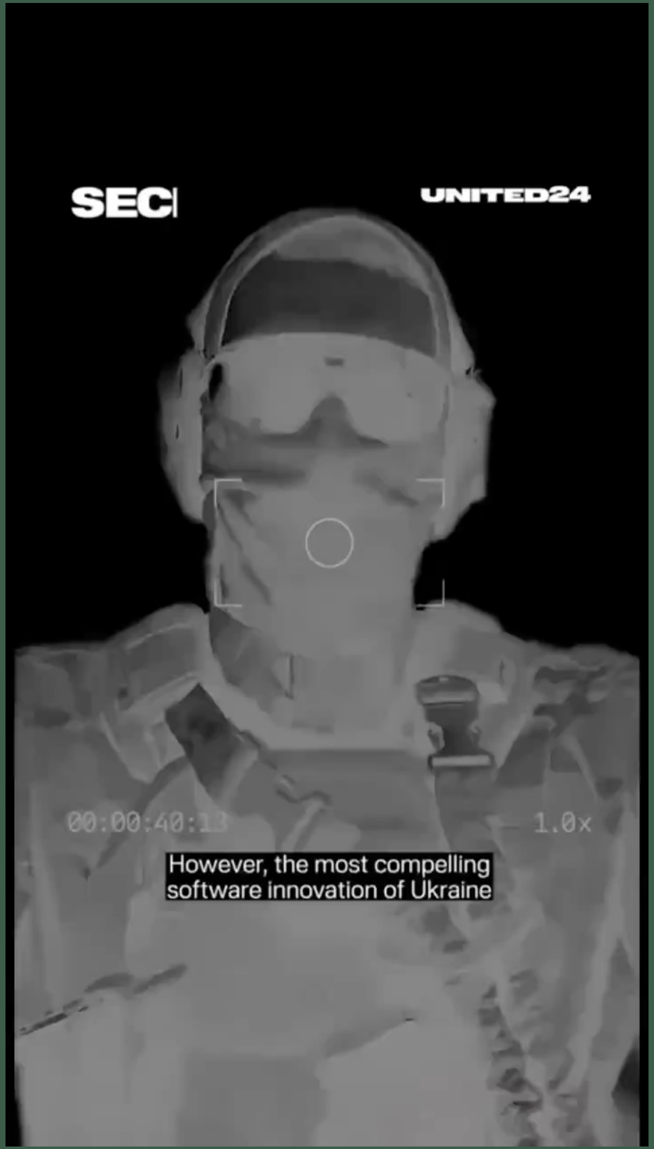
Full video: https://www.youtube.com/watch?app=desktop&v=PBUwJEM6yws

SOLUTIONS

PALANTIR TECHNOLOGIES

Full video: https://www.youtube.com/watch?v=r8LtdKFcAvg

DELTA



Source: https://armyinform.com.ua/2023/07/12/ukrayinska-systema-sytuaczijnoyi-obiznanosti-delta-projshla-vyprobuvannya-nato-i-mozhe-integruvaty-vynyshhuvachi-f-16/,
Full video: https://youtube.com/watch?v=rqzlUtJcZBs

CORVUS INTELLIGENCE

Landing page: https://corvusintell.com/

HOW CAN POSTGIS HELP IN THESE CASES?

DATABASE SCHEMA OVERVIEW

RESTRICTED_ZONES

int	id	PK	Primary Key
varchar	zone_name		Zone Name
geometry	geom		Geometry (Polygon)

DRONE_TELEMETRY

int	id	PK	Primary Key
varchar	drone_id		Drone ID
timestamp	recorded_at		Recorded at
geometry	geom		Geometry (Point)

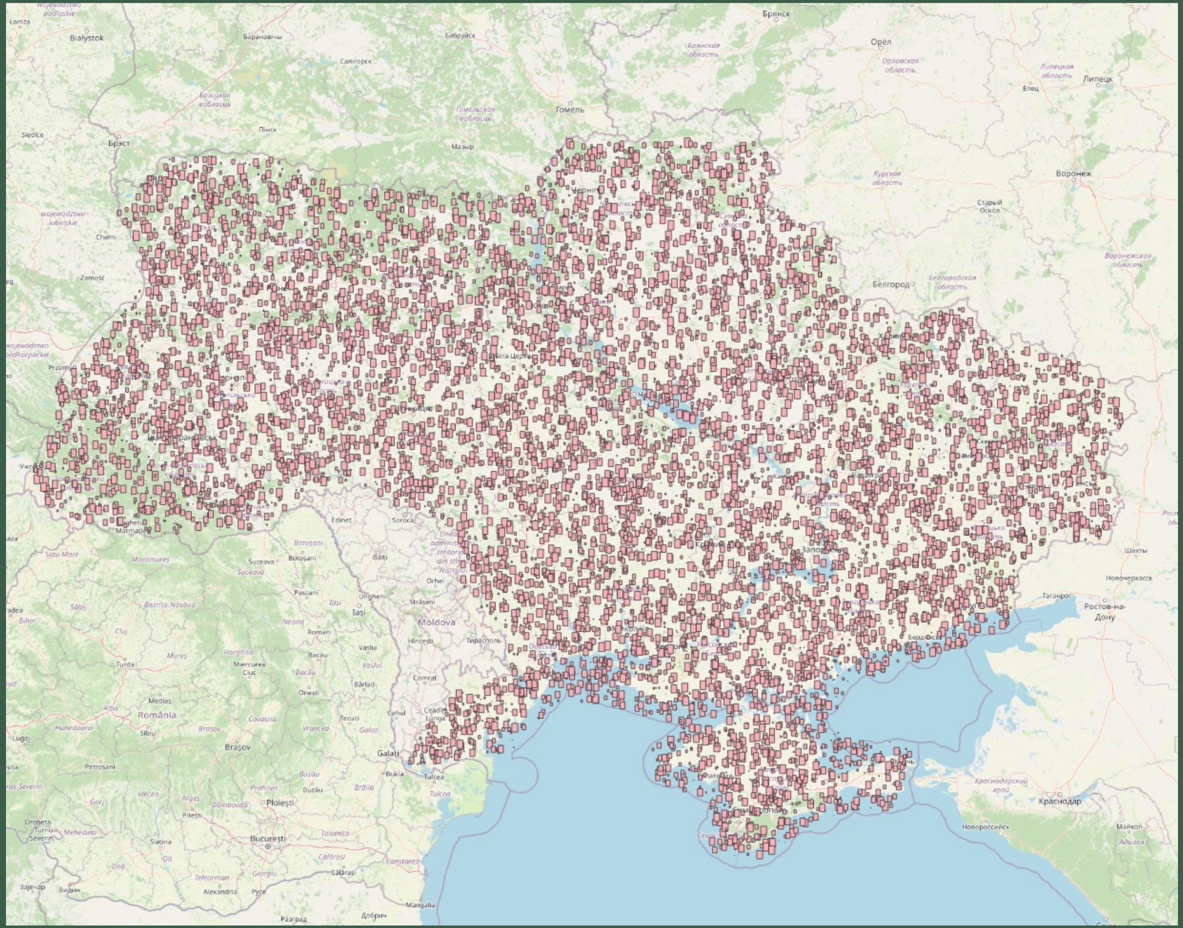
RESTRICTED_ZONES (10,000 rows): This table stores the geographic boundaries of restricted areas, represented as polygons.

```
-- No-fly Zone 1: Area near Kyiv
INSERT INTO spatial.restricted_zones (zone_name,
geom) VALUES
('Kyiv Zone', ST_GeomFromText('POLYGON((30.5238
50.4024, 30.5238 50.4724, 30.6238 50.4724, 30.6238
50.4024, 30.5238 50.4024))', 4326));
```

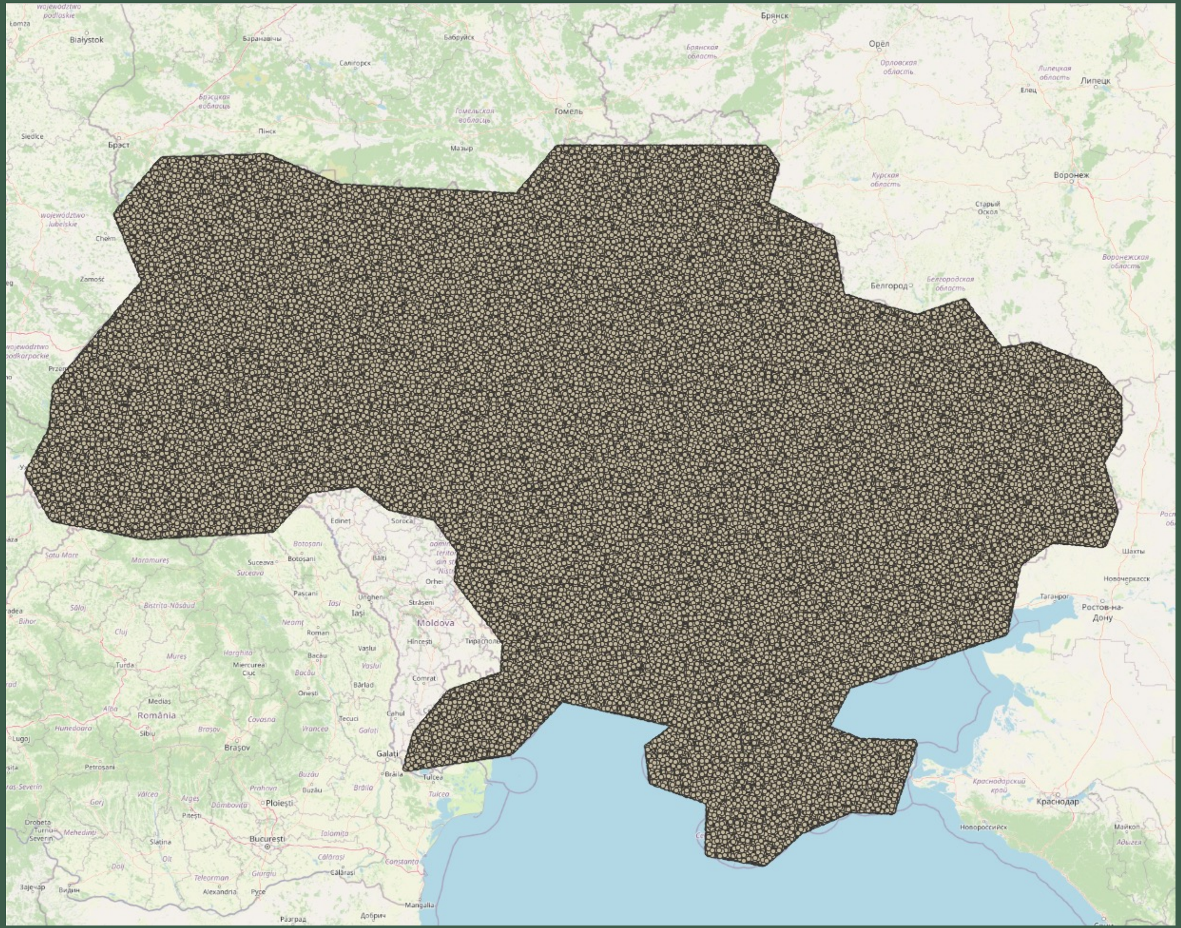
DRONE_TELEMETRY (1,000,000 rows): This table captures telemetry data from drones, including their geographic location at given timestamps.

```
-- Drone 1: Inside the Kyiv Zone
INSERT INTO spatial.drone_telemetry (drone_id,
recorded_at, geom) VALUES
('Drone1', '2023-12-01 10:00:00',
ST_SetSRID(ST_MakePoint(30.5738, 50.4324), 4326));
```

GEOSPATIAL VISUALIZATION OF SAMPLE DATA



RESTRICTED_ZONES (10,000 rows)



DRONE_TELEMETRY (1,000,000 rows)

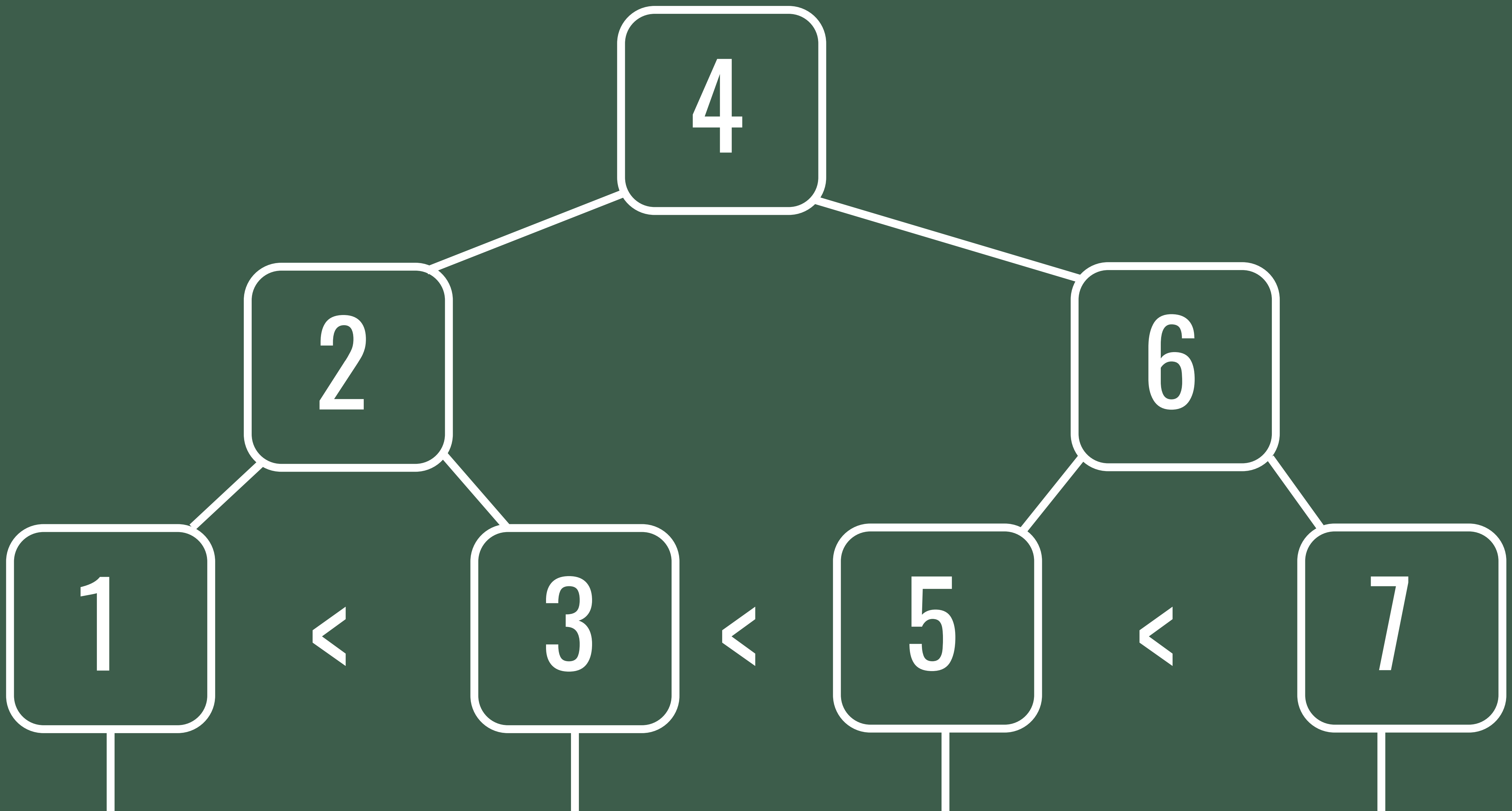
CASE 1: WHICH DRONES HAVE ENTERED RESTRICTED ZONES?

ANALYZING QUERY: EXECUTION TIME

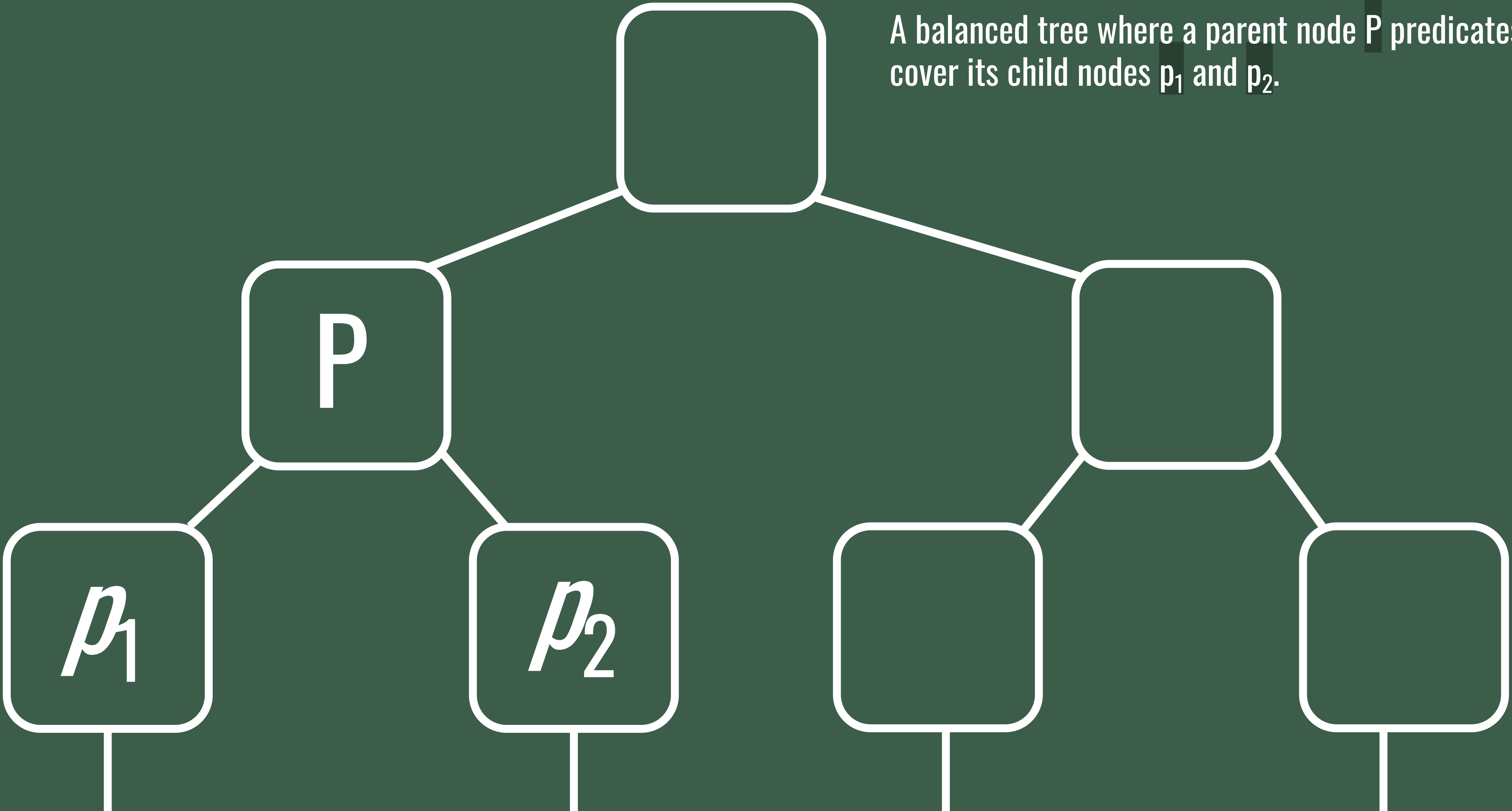
```sql
SELECT t.drone_id,
       t.recorded_at,
       r.zone_name,
       t.geom as drone,
       r.geom as restricted_zone
FROM spatial.drone_telemetry t
         INNER JOIN spatial.resricted_zones r
                    ON ST_Contains(r.geom, t.geom);
```

```
285 rows retrieved starting from 1 in 1 h 2 m 28 s 206 ms
(execution: 1 h 2 m 27 s 506 ms, fetching: 700 ms)
```

B-TREE INDEX STRUCTURE



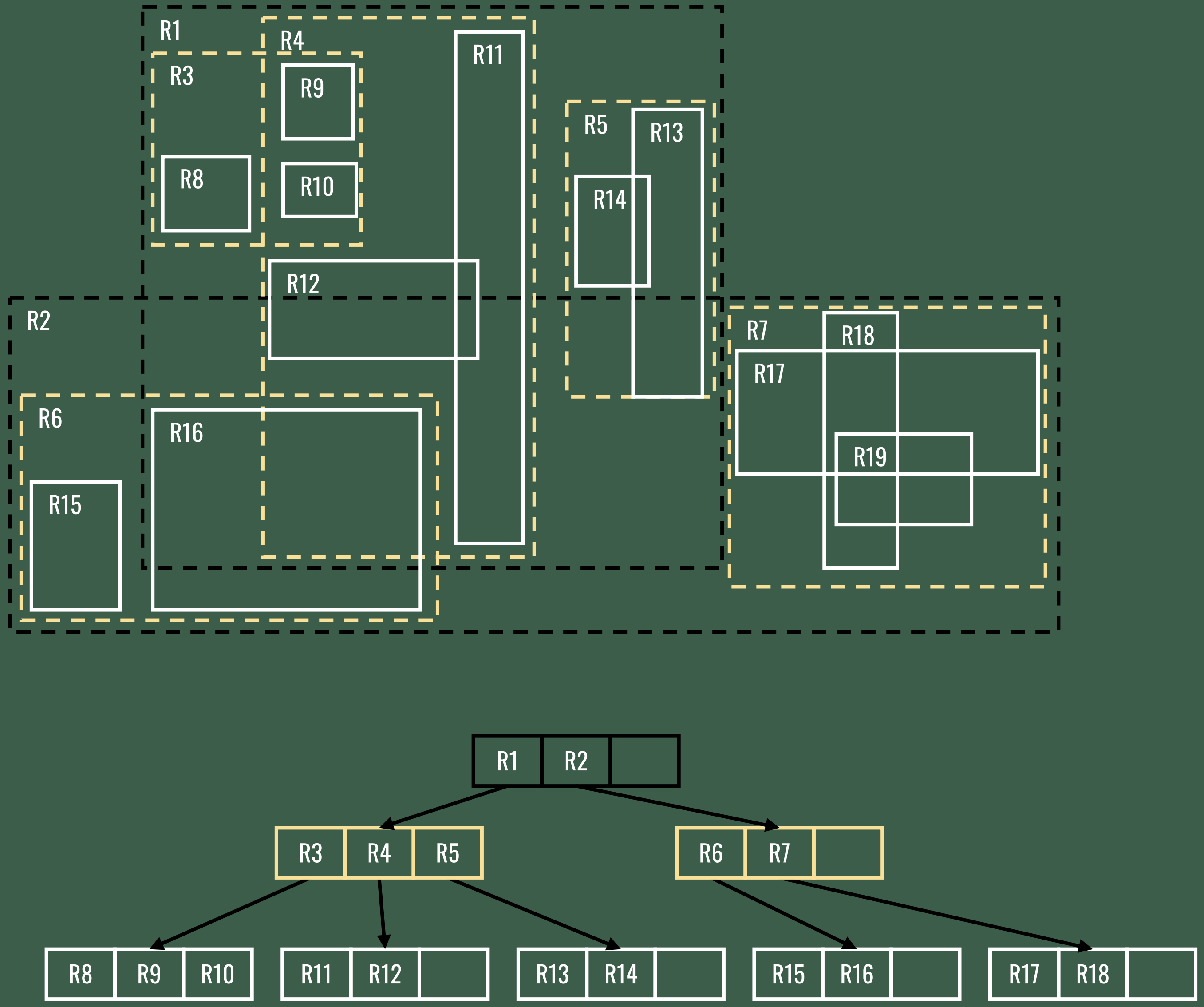
GIST STRUCTURE



A balanced tree where a parent node P predicates cover its child nodes p_1 and p_2.

IMPLEMENTING GIST INDEXING

```
CREATE INDEX
idx_restricted_zones_geom
ON
spatial.restricted_zones
USING GIST (geom);
```



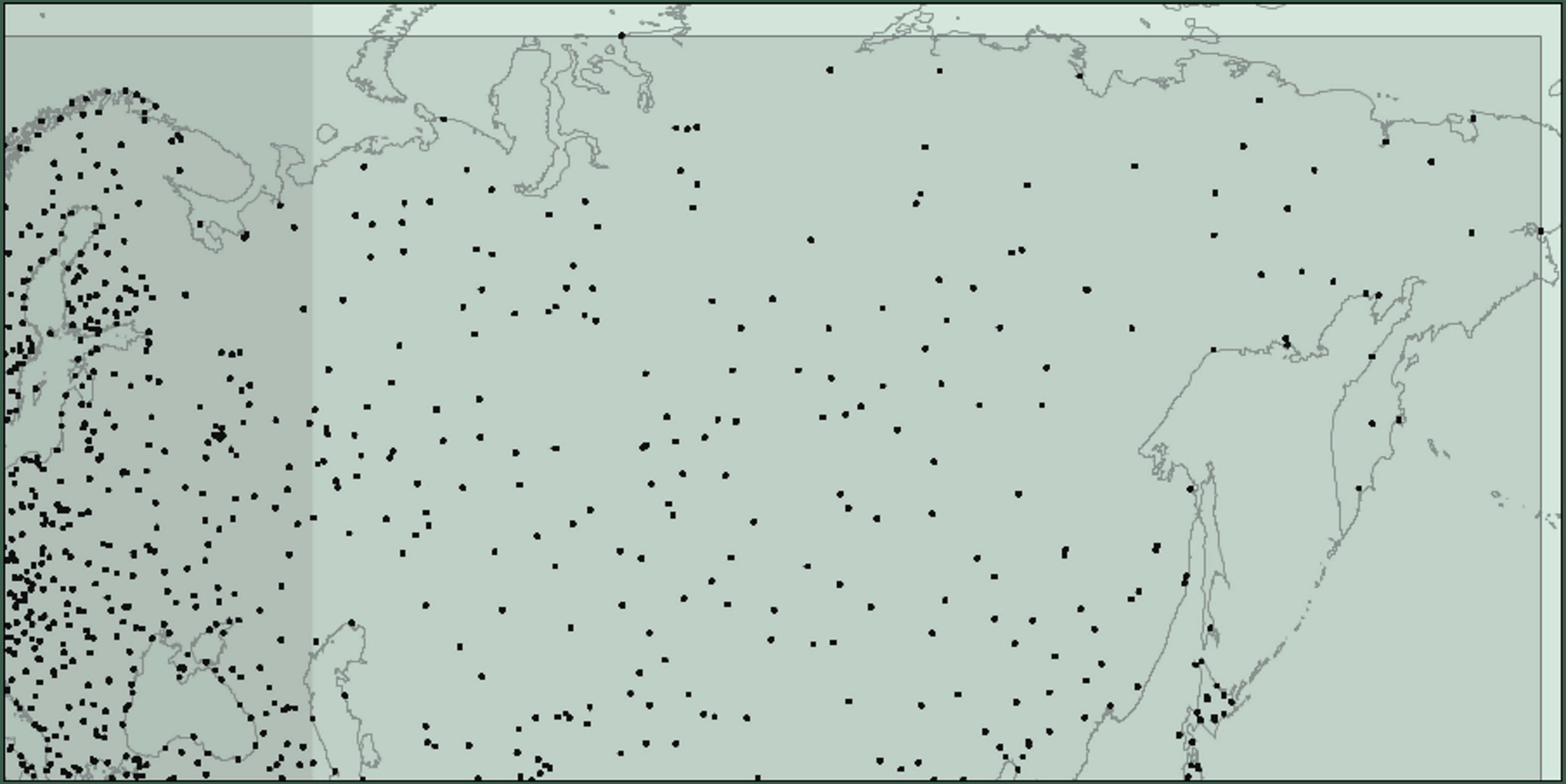
Mastering PostgreSQL By : Hans-Jürgen Schönig: https://subscription.packtpub.com/book/data/9781800567498

ANALYZING GIST: THE POINT_OPS OPERATOR

```sql
SELECT
    amopopr::regoperator, oprcode::regproc, left(obj_description(opr.oid, 'pg_operator'), 19) description
FROM pg_am am
JOIN pg_opclass opc ON opcmethod = am.oid
JOIN pg_amop amop ON amopfamily = opcfamily
JOIN pg_operator opr ON opr.oid = amopopr
WHERE amname = 'gist'
AND opcname = 'point_ops'
ORDER BY amopstrategy;
```

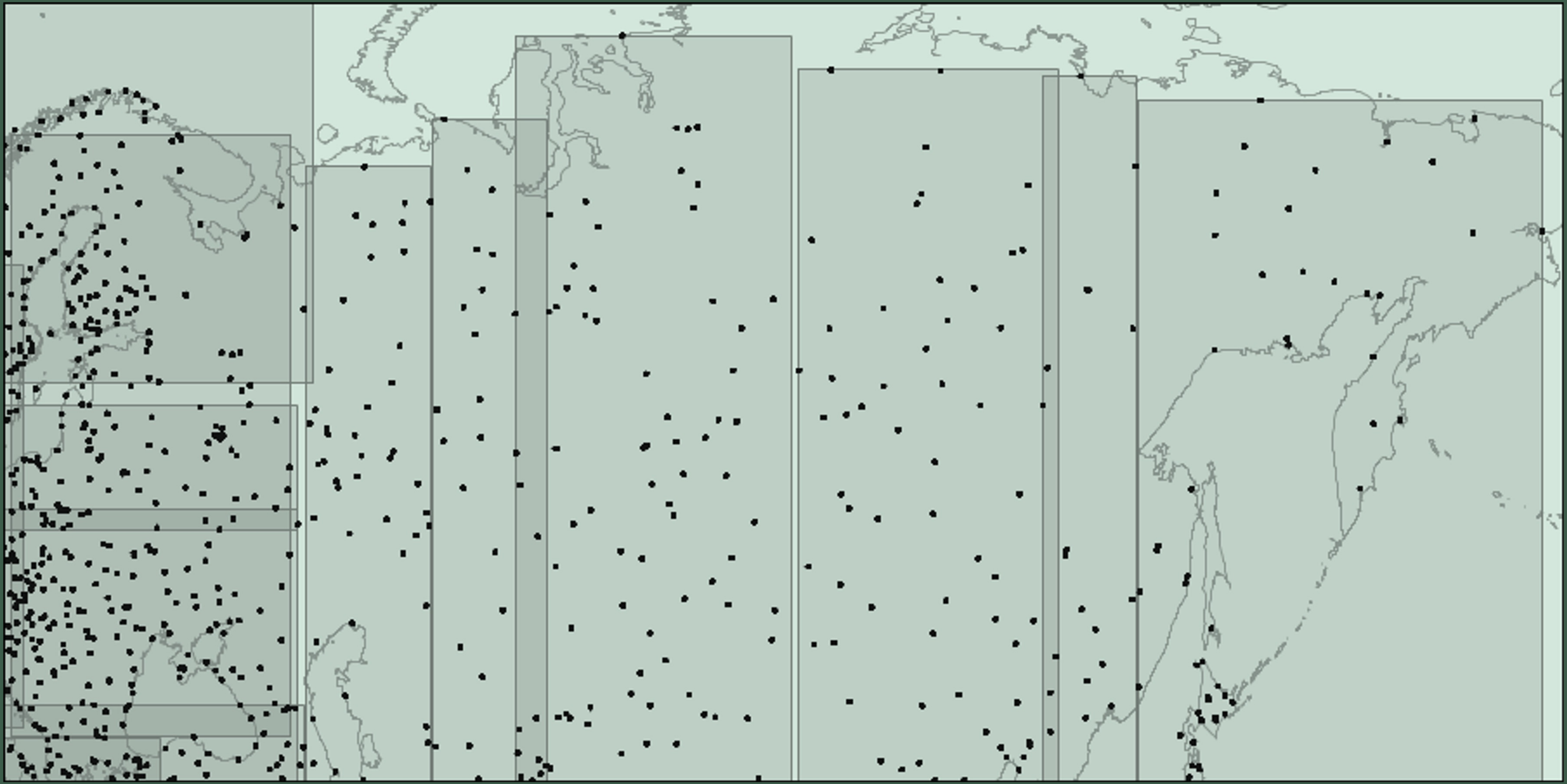
	amopopr	oprcode	description
1	<<(point,point)	point_left	is left of
2	>>(point,point)	point_right	is right of
3	~=(point,point)	point_eq	same as
4	<<|(point,point)	point_below	is below
5	|>>(point,point)	point_above	is above
6	<->(point,point)	point_distance	distance between
7	<@(point,box)	on_pb	point inside box
8	<^(point,point)	point_below	deprecated, use <<|
9	>^(point,point)	point_above	deprecated, use |>>
10	<@(point,polygon)	pt_contained_poly	is contained by
11	<@(point,circle)	pt_contained_circle	is contained by

GIST, LEVEL 1 (POINT_OPS)



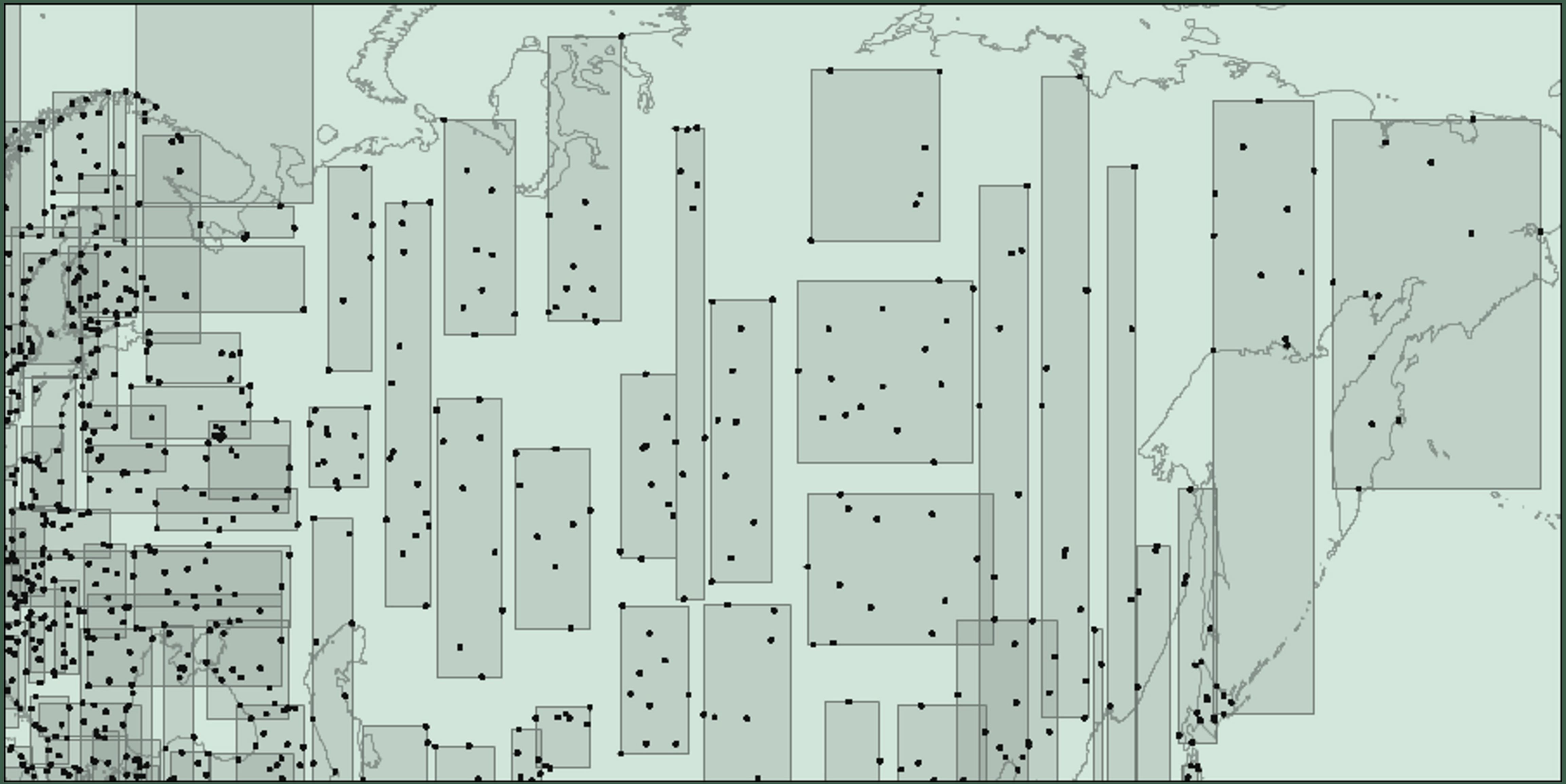
Source: https://blog.csdn.net/Michaelia_hu/article/details/123181525?spm=1001.2014.3001.5501

GIST, LEVEL 2 (POINT_OPS)



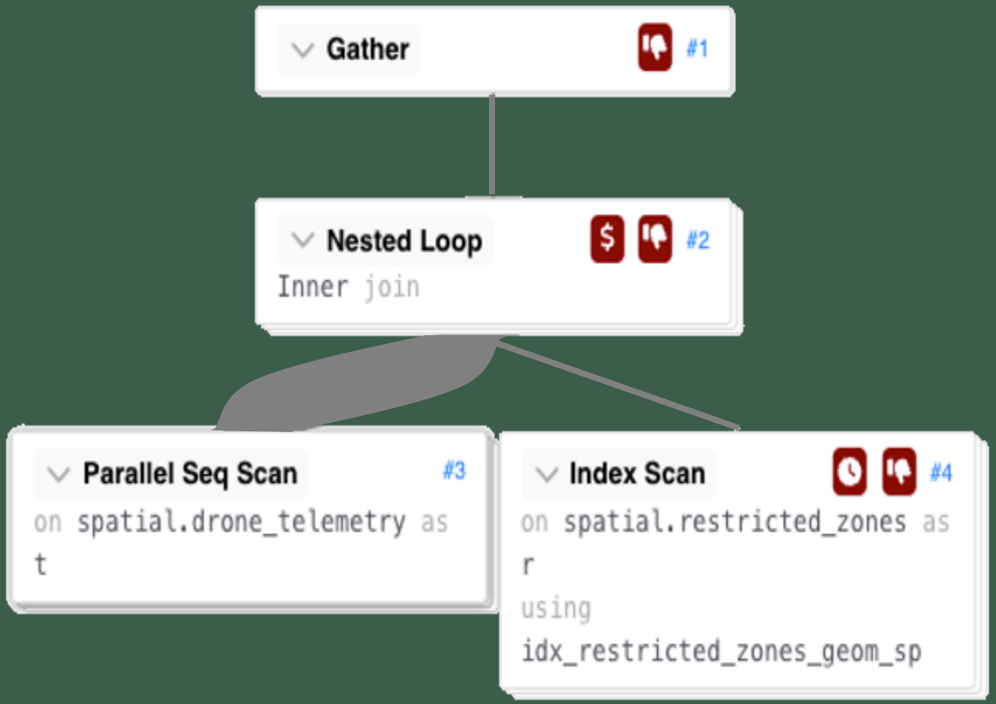
Source: https://blog.csdn.net/Michaelia_hu/article/details/123181525?spm=1001.2014.3001.5501

GIST, LEVEL 3 (POINT_OPS)



Source: https://blog.csdn.net/Michaelia_hu/article/details/123181525?spm=1001.2014.3001.5501

QUERY RESULTS, GIST

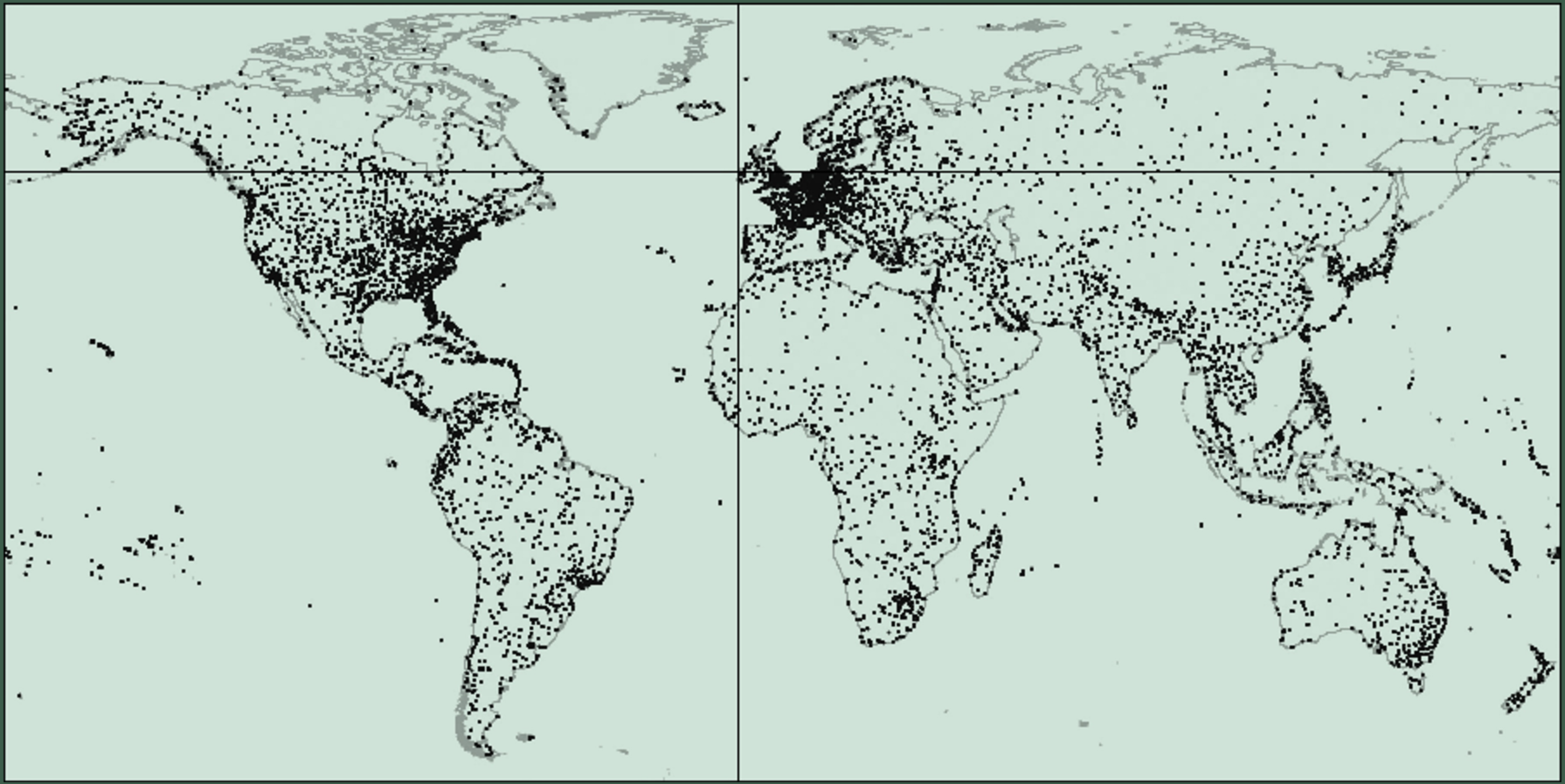



```
Nested Loop  (cost=0.15..250356360.00
rows=9336232 width=174)
  ->  Seq Scan on drone_telemetry t
(cost=0.00..20300.00 rows=1000000
width=45)
  ->  Index Scan using
idx_restricted_zones_geom on
restricted_zones r  (cost=0.15..250.33
rows=1 width=129)
        Index Cond: (geom ~ t.geom)
        Filter: st_contains(geom, t.geom)
```

285 rows retrieved starting from 1 in 16 s 356 ms
(execution: 16 s 265 ms, fetching: 91 ms)

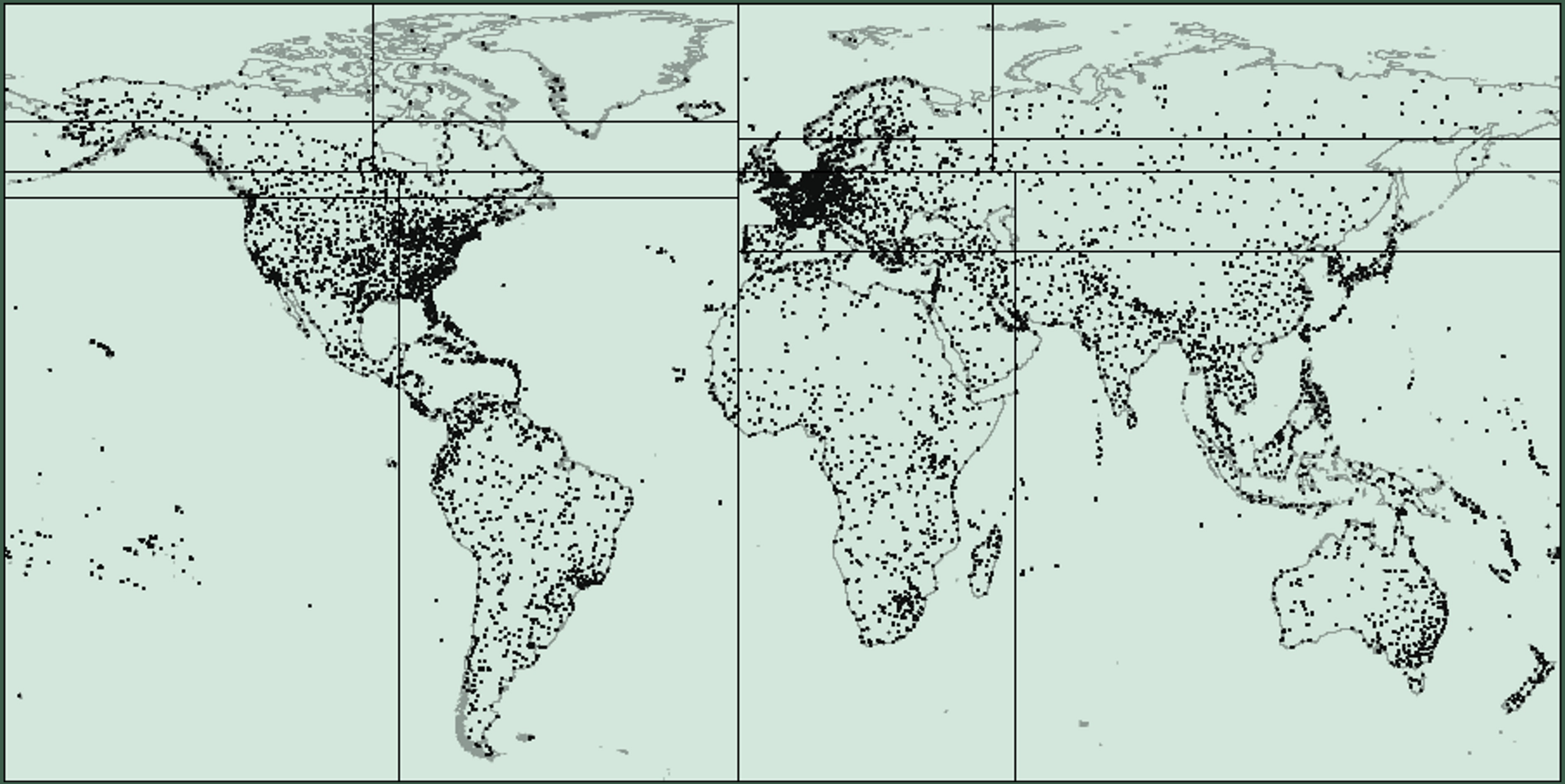
200x Performance Speed-Up

SP-GIST, LEVEL 1 (QUAD_POINT_OPS)



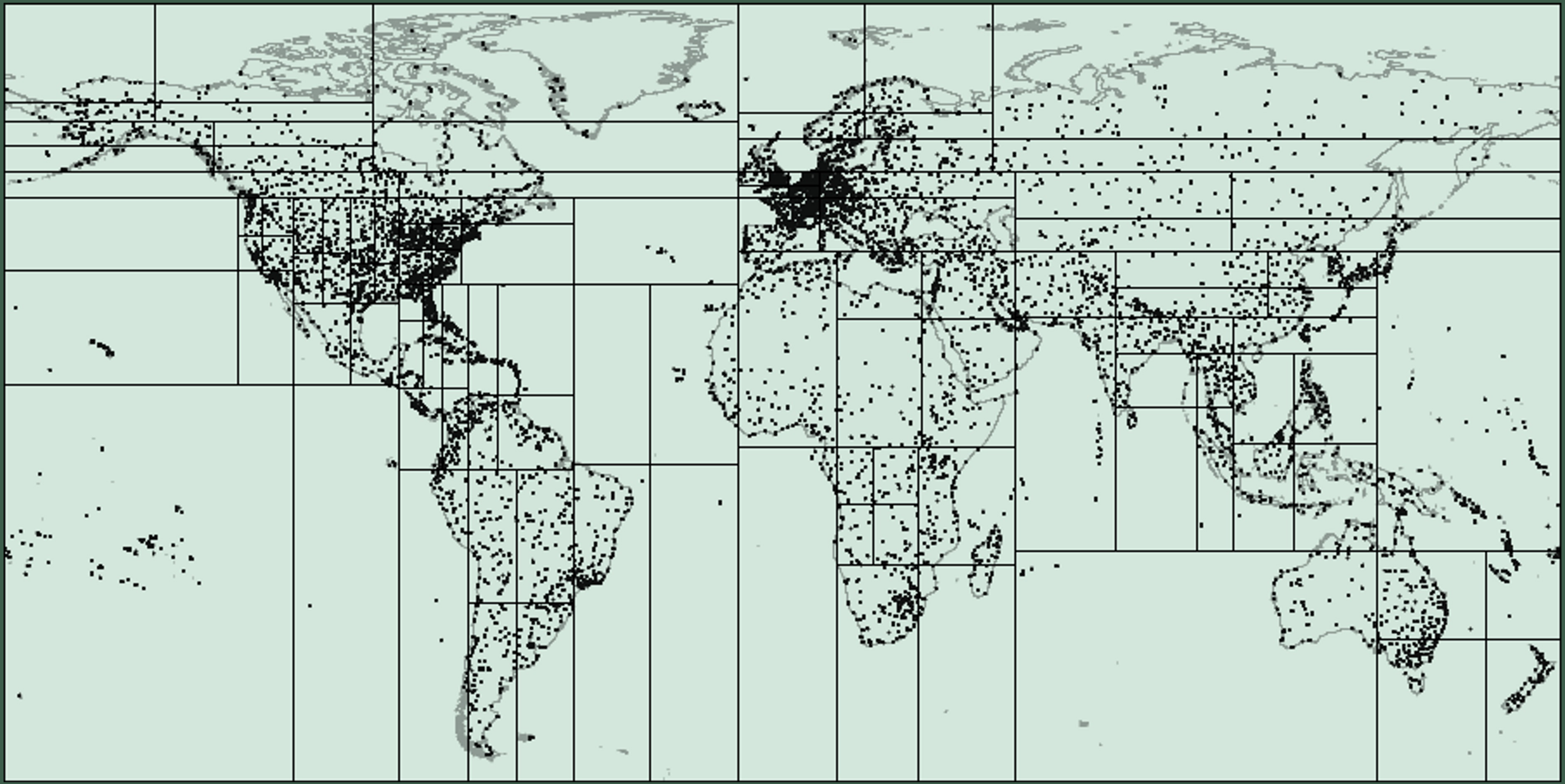
Source: https://programs.wiki/wiki/index-in-postgresql-6-sp-gist.html

SP-GIST, LEVEL 2 (QUAD_POINT_OPS)



Source: https://programs.wiki/wiki/index-in-postgresql-6-sp-gist.html

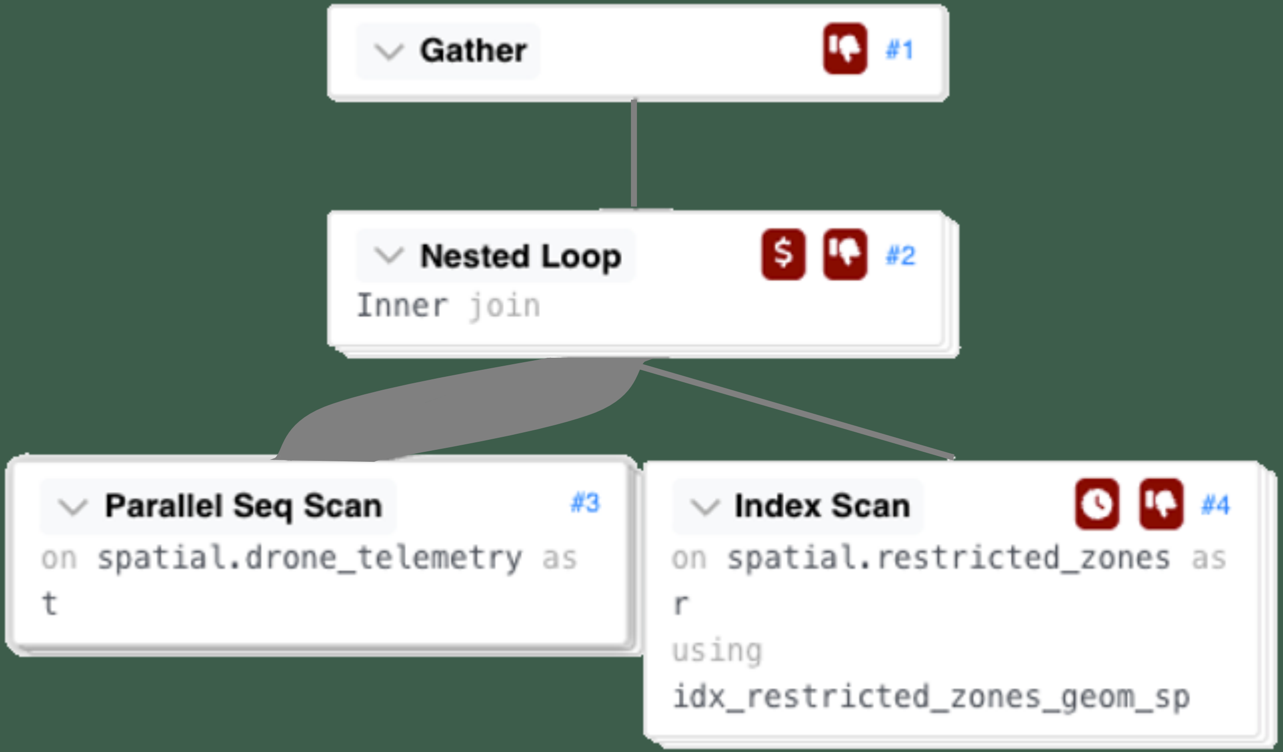
SP-GIST, LEVEL 3 (QUAD_POINT_OPS)



Source: https://programs.wiki/wiki/index-in-postgresql-6-sp-gist.html

QUERY RESULTS: SP-GIST

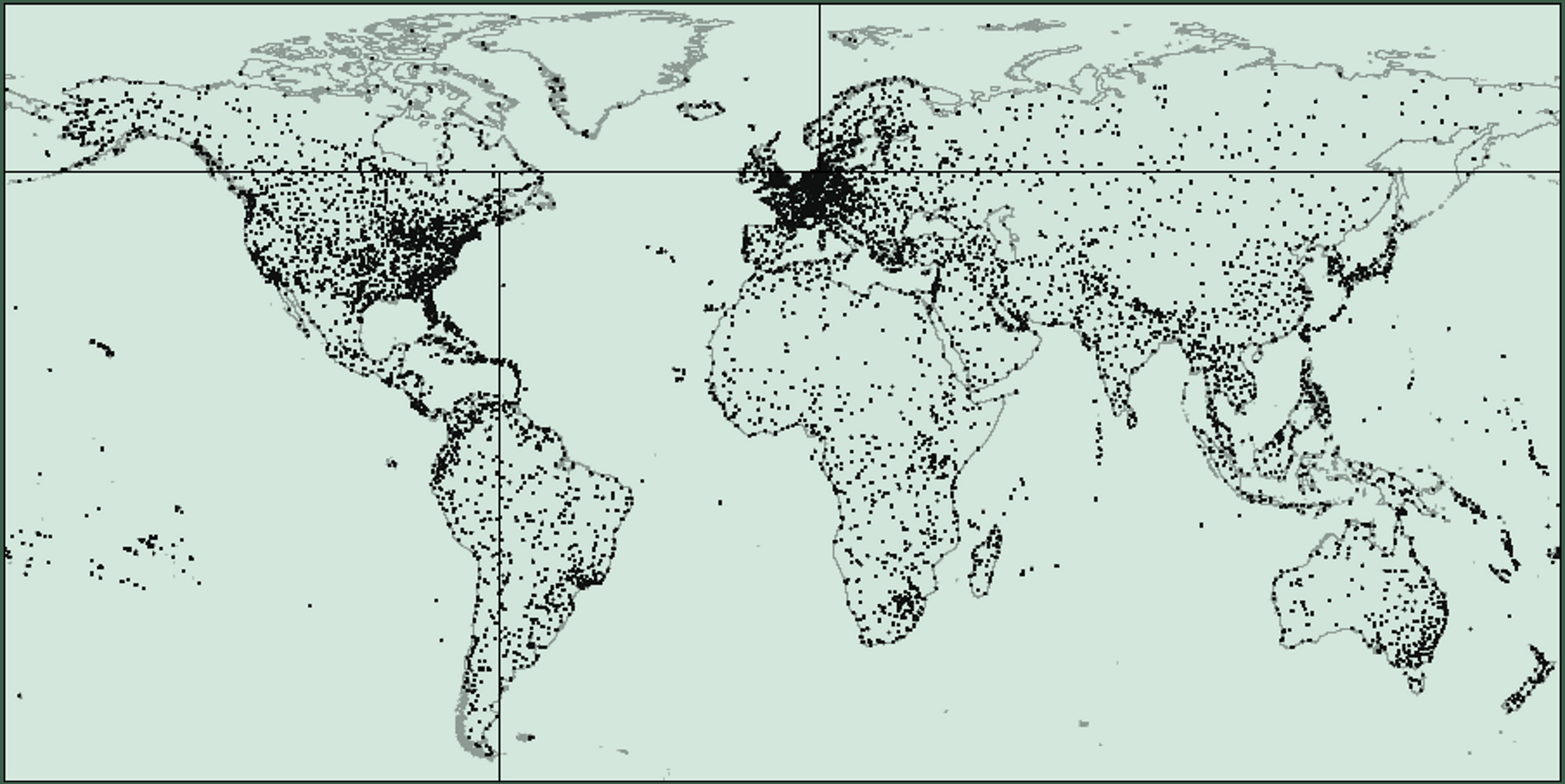
```
DROP INDEX IF EXISTS
spatial.idx_restricted_zones_geom;
CREATE INDEX
idx_restricted_zones_geom_sp ON
spatial.restricted_zones USING
SPGIST (geom);

ANALYZE spatial.restricted_zones;

285 rows retrieved starting from 1
in 14 s 111 ms
(execution: 14 s 60 ms, fetching:
51 ms)
```



SP-GIST, LEVEL 1 (KD_POINT_OPS)

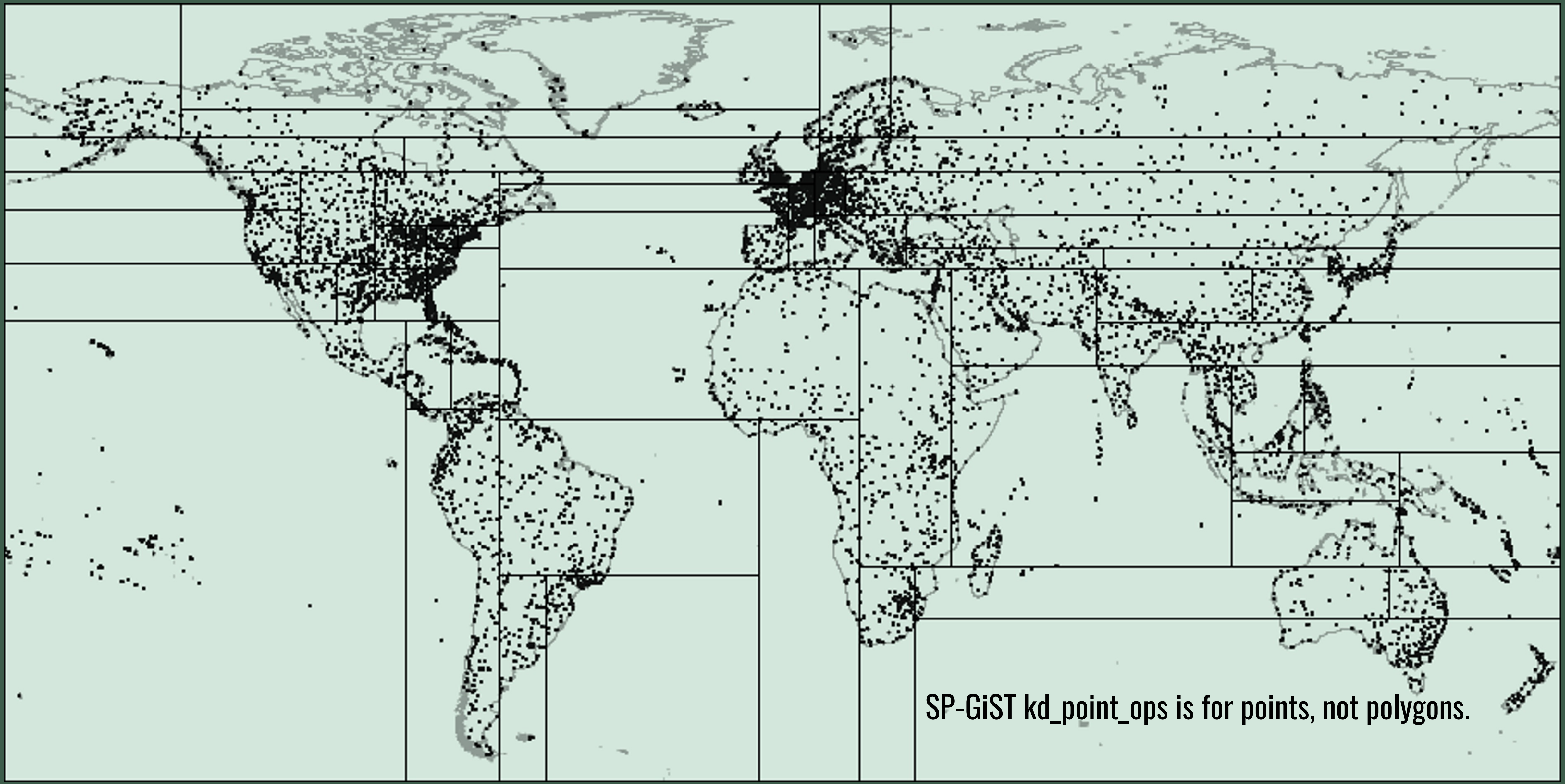
Source: https://programs.wiki/wiki/index-in-postgresql-6-sp-gist.html

SP-GIST, LEVEL 2 (KD_POINT_OPS)



Source: https://programs.wiki/wiki/index-in-postgresql-6-sp-gist.html

SP-GIST, LEVEL N (KD_POINT_OPS)



Source: https://programs.wiki/wiki/index-in-postgresql-6-sp-gist.html

SIZE COMPARISON OF GIST AND SP-GIST

RESTRICTED_ZONES (10,000 rows)

	index_name	index_size
1	idx_restricted_zones_gist	480 kB
2	idx_restricted_zones_gist_sp	504 kB

DRONE_TELEMETRY (1,000,000 rows)

	index_name	index_size
1	idx_drone_telemetry_geom_spgist_kd	51 MB
2	idx_drone_telemetry_geom_gist	44 MB
3	idx_drone_telemetry_geom_spgist	44 MB

CASE 2: WHICH DRONE IS THE CLOSEST TO THE SPECIFIED TARGET AREA?

DATABASE SCHEMA OVERVIEW

LATEST_DRONE_TELEMETRY			
string	drone_id	PK	Primary Key
timestamp	latest_recorded_at		Latest Recorded at
geometry	latest_geom		Latest Geometry (Point)

DRONE_TELEMETRY			
int	id	PK	Primary Key
string	drone_id		Drone ID
timestamp	recorded_at		Recorded at
geometry	geom		Geometry (Point)

LATEST_DRONE_TELEMETRY (100,000 rows):

- Contains the most recent location data for each drone
- drone_id is the primary key
- latest_recorded_at is the timestamp of the latest record
- latest_geom is the most recent geometry point

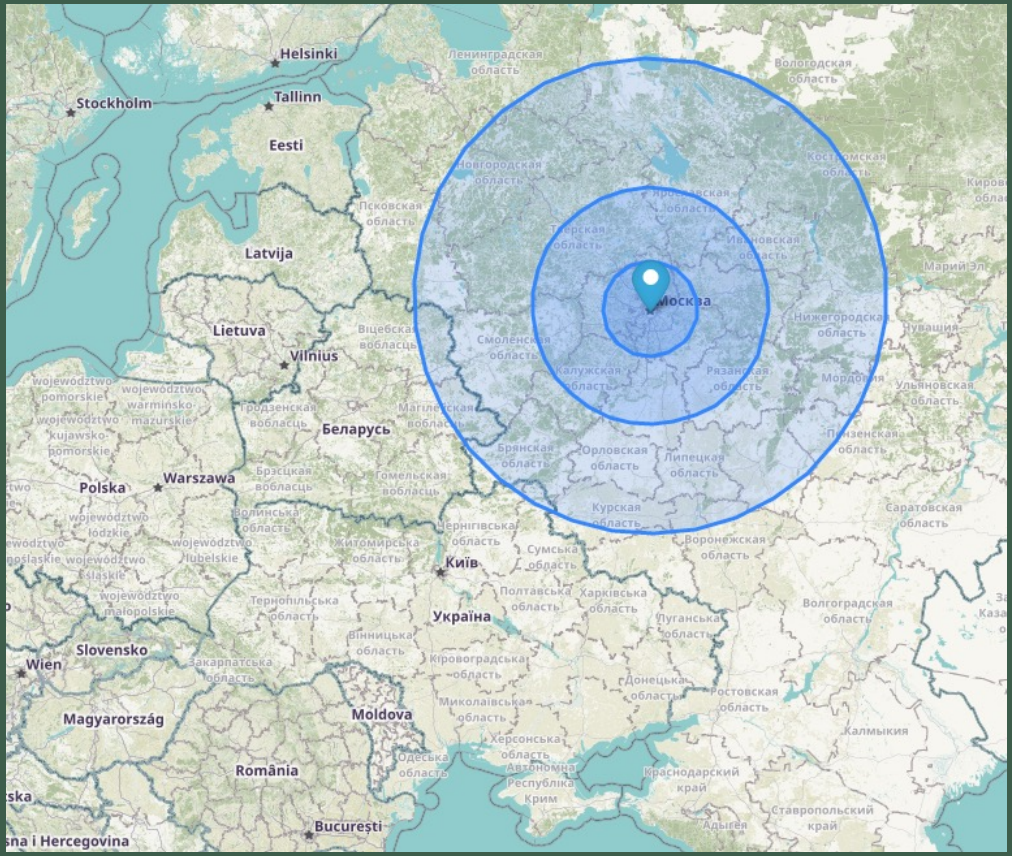
Our primary objective is to determine which drone is currently the closest to a predetermined target of interest—the **Kremlin in Moscow.**

Latitude: 55.7517° N

Longitude: 37.6176° E

DRONES WITHIN 500 KM OF THE KREMLIN

```sql
SELECT
    drone_id,
    latest_recorded_at,
    latest_geom,
    ST_Distance(
        ST_Transform(latest_geom, 4326)::geography,
        ST_MakePoint(37.6176, 55.7517)::geography
    ) AS distance_to_kremlin
FROM
    spatial.latest_drone_telemetry
WHERE
    ST_DWithin(
        ST_Transform(latest_geom, 4326)::geography,
        ST_MakePoint(37.6176, 55.7517)::geography,
        500000
    )
ORDER BY distance_to_kremlin;
```



TOP 10 NEAREST DRONES USING GIST INDEX

```sql
SELECT
    drone_id,
    latest_recorded_at,
    latest_geom,
    ST_Distance(latest_geom::geography, ST_MakePoint(37.6176,
55.7517)::geography) AS distance_to_kremlin
FROM
    spatial.latest_drone_telemetry
ORDER BY
    latest_geom <-> ST_SetSRID(ST_MakePoint(37.6176, 55.7517),
4326)
LIMIT 10;
```

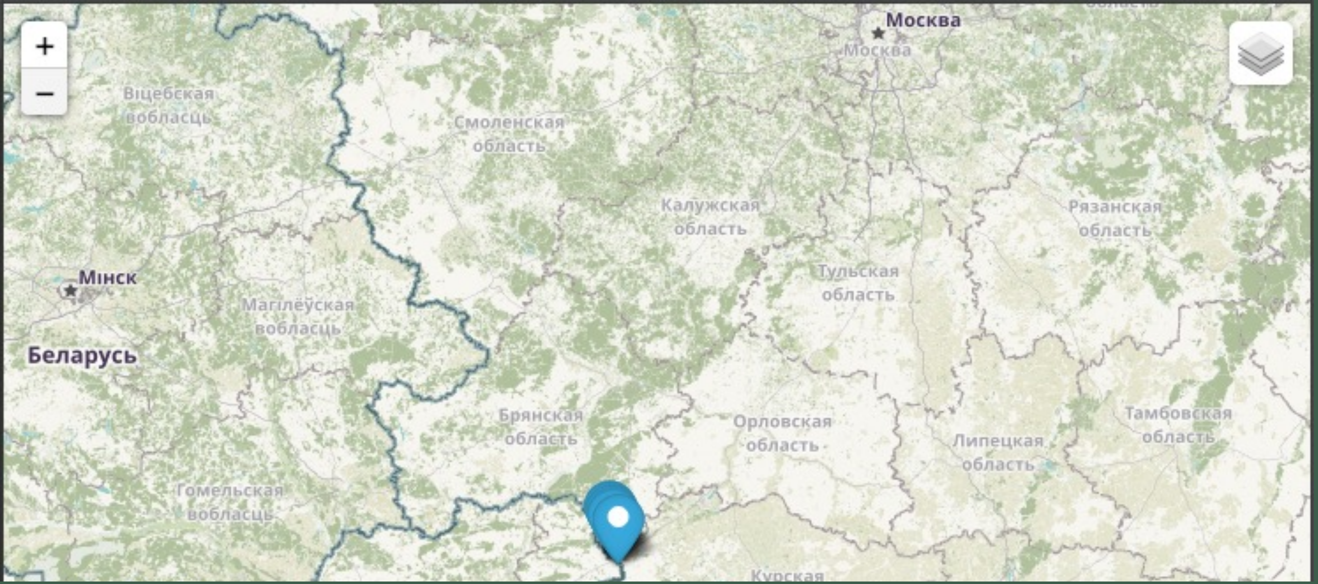
```
Limit  (cost=0.28..263.59 rows=10
width=61)
  ->  Index Scan using
idx_latest_drone_telemetry_geom_g
ist on latest_drone_telemetry
(cost=0.28..2633056.28
rows=100000 width=61)
        Order By: (latest_geom <-
>
'0101000020E61000003B014D840DCF42
409C33A2B437E04B40'::geometry)
```

Output | Result 12

10 rows | CSV

	d...	latest_recorded_at	latest_geom	distance_...
1	54181	2023-12-01 09:00:00.000000	0101000020E610000…	476141.63270863
2	53554	2023-12-01 09:00:00.000000	0101000020E610000…	476989.02686474
3	12719	2023-12-01 09:00:00.000000	0101000020E610000…	475071.88728588
4	59844	2023-12-01 09:00:00.000000	0101000020E610000…	478365.63731183
5	44992	2023-12-01 09:00:00.000000	0101000020E610000…	482994.08823383
6	91223	2023-12-01 09:00:00.000000	0101000020E610000…	479036.12661429
7	98285	2023-12-01 09:00:00.000000	0101000020E610000…	477018.90379506
8	50550	2023-12-01 09:00:00.000000	0101000020E610000…	483144.820276
9	66754	2023-12-01 09:00:00.000000	0101000020E610000…	480339.37269975
10	93292	2023-12-01 09:00:00.000000	0101000020E610000…	479109.95749884

Geo Viewer



COMPARATIVE ANALYSIS: QUERYING DISTANCE TO TARGET WITH AND WITHOUT SPATIAL CONSTRAINTS

Query with circles

```
10 rows retrieved starting from 1 in 882 ms
(execution: 664 ms, fetching: 218 ms)
```

Query with `<->` operator

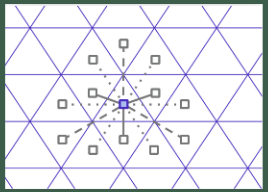
```
10 rows retrieved starting from 1 in 80 ms
(execution: 60 ms, fetching: 20 ms)
```

10x Performance Speed-Up

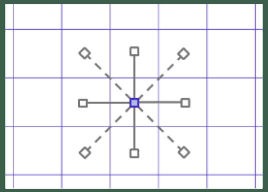
CASE 3: WHICH DRONES HAVE ENTERED RESTRICTED ZONES? WITH H3

UNDERSTANDING H3 GEOSPATIAL INDEXING: GRIDS, RESOLUTIONS, AND COVERAGE

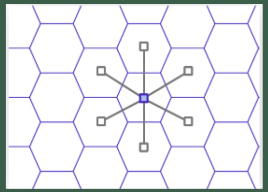
TRIANGLE



SQUARE



HEXAGON



Res	Total number of cells	Number of hexagons	Number of pentagons
0	122	110	12
1	842	830	12
2	5,882	5,870	12
3	41,162	41,150	12
4	288,122	288,110	12
5	2,016,842	2,016,830	12
6	14,117,882	14,117,870	12
7	98,825,162	98,825,150	12
8	691,776,122	691,776,110	12
9	4,842,432,842	4,842,432,830	12
10	33,897,029,882	33,897,029,870	12
11	237,279,209,162	237,279,209,150	12
12	1,660,954,464,122	1,660,954,464,110	12
13	11,626,681,248,842	11,626,681,248,830	12
14	81,386,768,741,882	81,386,768,741,870	12
15	569,707,381,193,162	569,707,381,193,150	12

UNDERSTANDING H3 GEOSPATIAL INDEXING: GRIDS, RESOLUTIONS, AND COVERAGE

Full video: https://www.youtube.com/watch?v=ay2uwtRO3QE

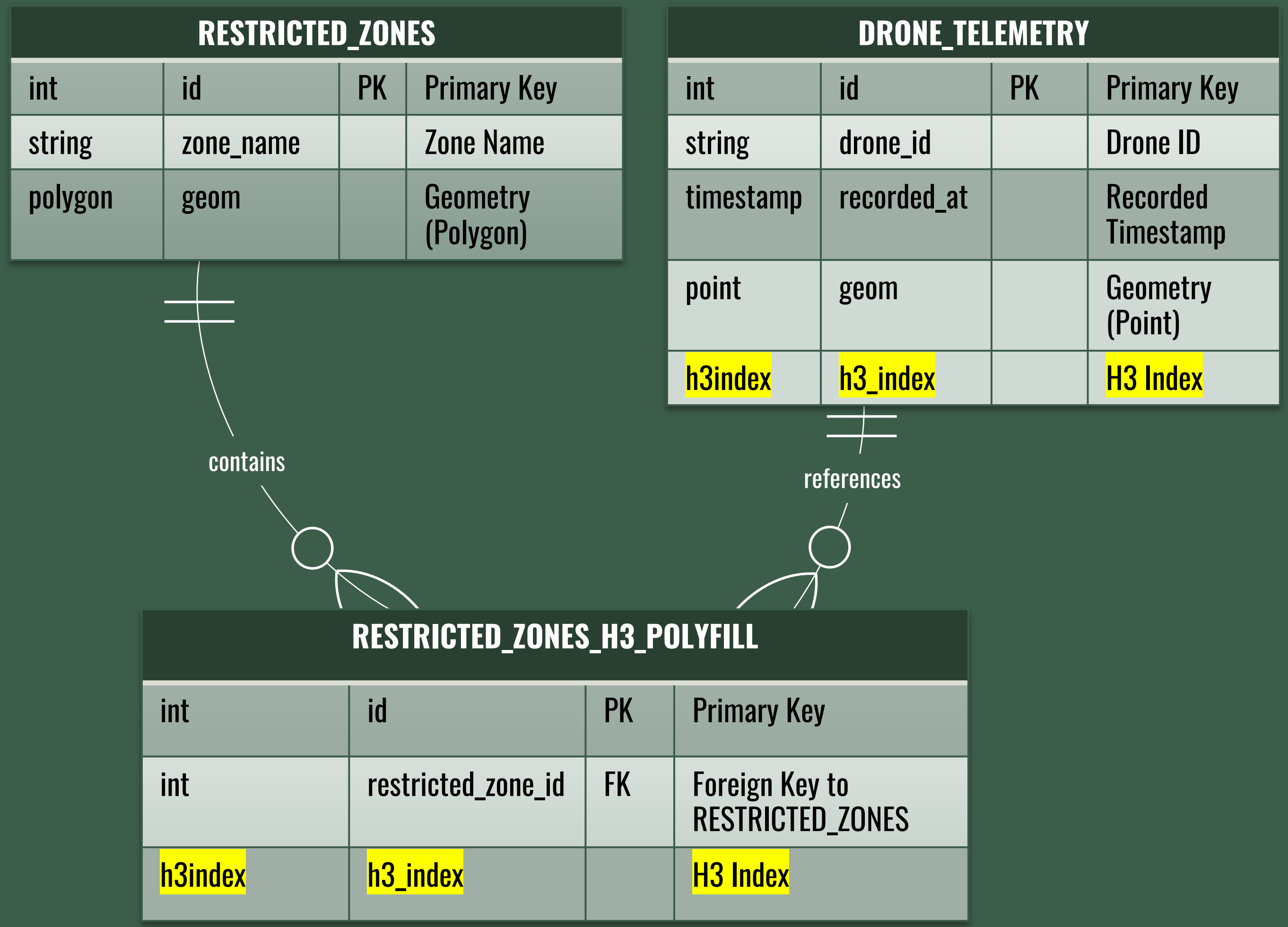
HARNESSING H3 IN POSTGRESQL

Installation & Availability:

- H3 is a powerful geospatial indexing system and can be integrated into PostgreSQL
- To install, use the CREATE EXTENSION h3; command
- H3 is supported in cloud platforms, such as AWS RDS, enhancing its accessibility.

FUNCTION	DESCRIPTION
h3_lat_lng_to_cell	Converts latitude and longitude to an H3 index
h3_cell_to_boundary	Returns the boundary coordinates of an H3 index cell
h3_get_resolution	Determines the resolution level of an H3 index
h3_cell_to_parent	Provides the parent cell of a given H3 index at a specified resolution
h3_cell_to_children	Lists the children cells of a given H3 index at a specified resolution
h3_polygon_to_cells	Converts a polygon to a set of H3 cells covering it

DATABASE SCHEMA: SPATIAL ANALYSIS OF DRONE TELEMETRY AND RESTRICTED ZONES

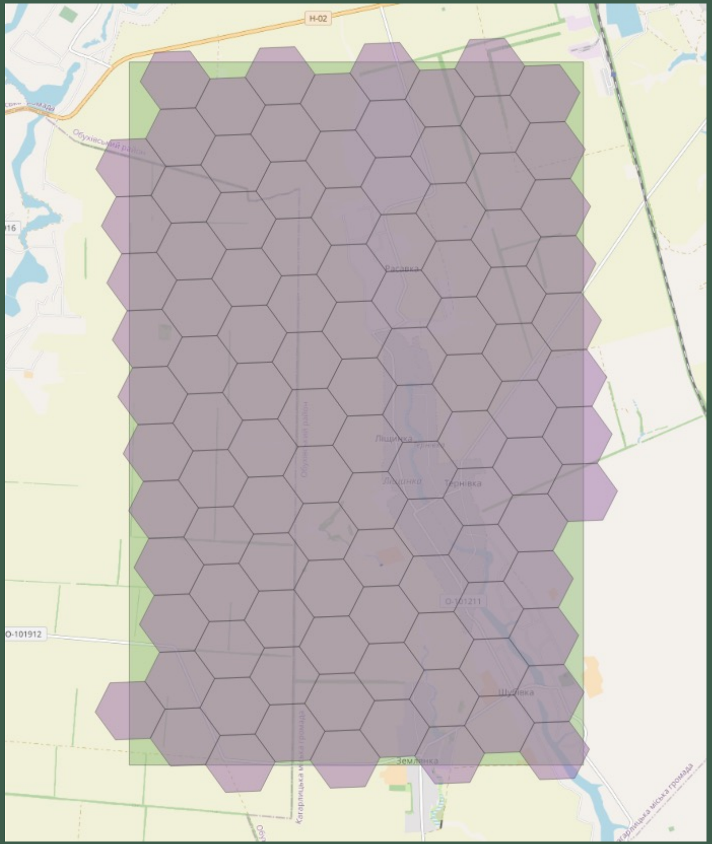


This diagram presents the database schema for our spatial analysis system.
It includes three main tables:

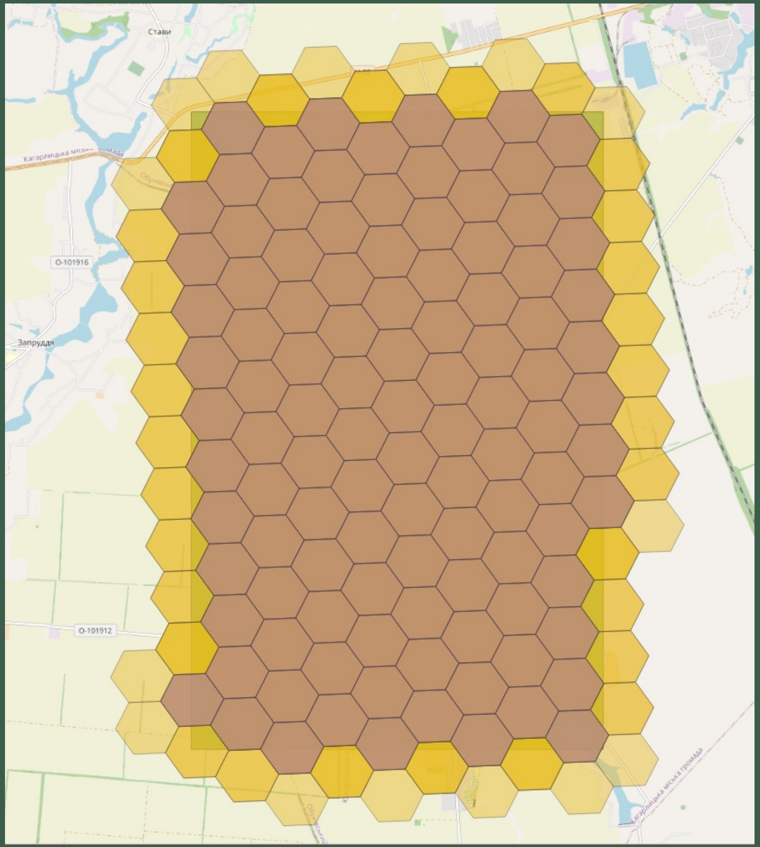
- **RESTRICTED_ZONES** detailing geographically restricted areas
- **DRONE_TELEMETRY** capturing telemetry data from drones
- **RESTRICTED_ZONES_H3_POLYFILL** linking restricted zones with H3 geospatial indexes

The schema illustrates the relationships between restricted zones and drone telemetry data, emphasizing the integration of H3 indexing for advanced spatial queries and analysis.

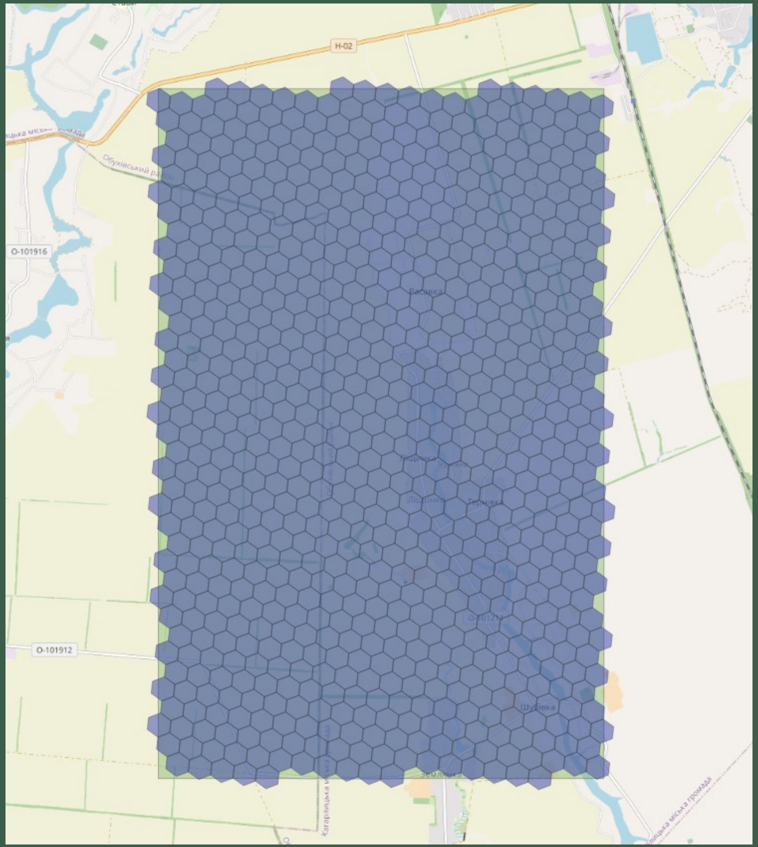
UNDERSTANDING H3 POLYFILLING: METHODOLOGY AND POTENTIAL PITFALLS



h3_polygon_to_cells(geom, 8)



h3_grid_disk(
h3_polygon_to_cells(geom, 8)
, 1)



h3_polygon_to_cells(geom, 9)

IDENTIFYING DRONES IN RESTRICTED ZONES USING H3 INDEXES

```
SELECT dt.drone_id,
       dt.recorded_at,
       rz.zone_name,
       dt.geom as drone,
       rz.geom as restricted_zone
FROM spatial.drone_telemetry dt
INNER JOIN spatial.restricted_zones_h3_polyfill rzhp
     ON dt.h3_index = rzhp.h3_index
INNER JOIN spatial.restricted_zones rz
     ON rzhp.restricted_zone_id = rz.id;
```

```
290 rows retrieved starting from 1 in 436 ms
    (execution: 376 ms, fetching: 60 ms)
```

35x Performance Speed-Up

CASE 4: FINDING DEVICES AT THE SAME PLACE AND TIME

AGGREGATING DRONE TELEMETRY DATA FOR SPATIO-TEMPORAL ANALYSIS

DRONE_TELEMETRY

int	id	PK	Primary Key
string	drone_id		Drone ID
timestamp	recorded_at		Recorded Timestamp
geometry	geom		Geometry (Point)
h3index	h3_index		H3 Index

DRONE_TELEMETRY_AGGREGATED

int	id	PK	Primary Key
string	drone_id		Drone ID
timestamp	time_block		Time Block
h3index	h3_parent_index		Parent H3 Index

```sql
INSERT INTO spatial.drone_telemetry_aggregated
(drone_id, time_block, h3_parent_index)
SELECT
    drone_id,
    date_trunc('minute', recorded_at) -
(EXTRACT(MINUTE FROM recorded_at)::integer % 15) *
interval '1 minute' AS time_block,
    h3_cell_to_parent(h3_index) AS h3_parent_index
FROM
    spatial.drone_telemetry
GROUP BY
    drone_id,
    time_block,
    h3_parent_index;
```

Original Concept by [Andriy Zabavskyy]

DETECTING SIMULTANEOUS DRONE PRESENCE WITH AGGREGATED TELEMETRY DATA

```sql
SELECT
    h3_parent_index,
    time_block,
    COUNT(DISTINCT drone_id) AS unique_drones_count,
    ARRAY_AGG(DISTINCT drone_id) AS drones
FROM
    spatial.drone_telemetry_aggregated
GROUP BY
    h3_parent_index,
    time_block
HAVING
    COUNT(DISTINCT drone_id) > 1;
```

293 ms (execution: 243 ms, fetching: 50 ms)

	h3_parent_index	time_block	unique_drones_count	drones
1	872d76d6effffff	2023-12-01 04:00:00.000000	2	{39858,57636}
2	872d76d6effffff	2023-12-01 07:00:00.000000	2	{76582,91035}
3	872d76d6dffffff	2023-12-01 01:00:00.000000	2	{27930,78132}
4	872d76d6dffffff	2023-12-01 08:00:00.000000	3	{10105,4891,65405}
5	872d76d6cffffff	2023-12-01 01:00:00.000000	3	{13442,76009,76914}

CASE 5: TRACING DEVICES THAT TRAVELED TOGETHER

IDENTIFYING DRONES WITH SHARED FLIGHT PATHS

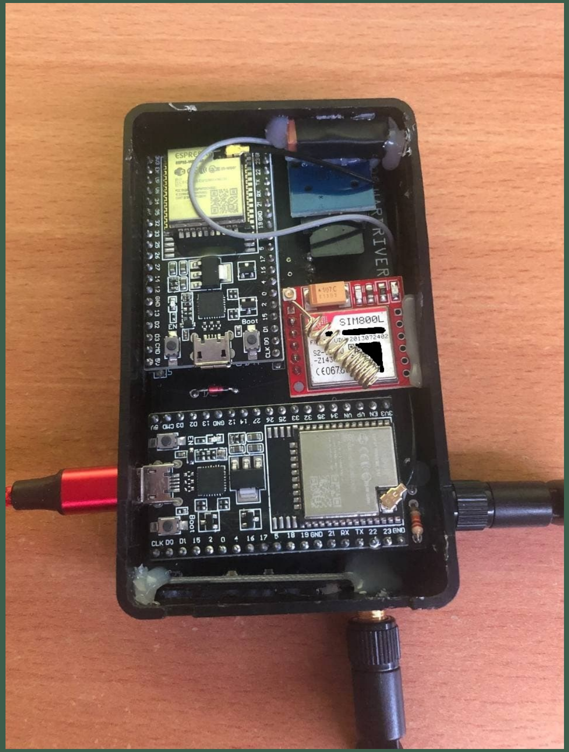
```
WITH PairedMovements AS (
    SELECT
        a.drone_id AS drone_id_a,
        b.drone_id AS drone_id_b,
        a.time_block,
        a.h3_parent_index
    FROM
      spatial.drone_telemetry_aggregated
a
    INNER JOIN
spatial.drone_telemetry_aggregated b
     ON a.h3_parent_index =
            b.h3_parent_index
        AND a.drone_id <> b.drone_id
        AND a.time_block = b.time_block
)
```

```
SELECT
    drone_id_a,
    drone_id_b,
    COUNT(*) AS pair_count,
    ARRAY_AGG(DISTINCT h3_parent_index)
AS h3_cells
FROM
    PairedMovements
GROUP BY
    drone_id_a, drone_id_b
HAVING
    COUNT(*) > 1
    AND COUNT(DISTINCT h3_parent_index) >
1;
```

	drone_id_a	drone_id_b	pair_count	h3_cells
1	14242	70247	2	{871e714b4ffffff,871e70a8effffff}
2	16207	80144	2	{871e6e48cffffff,871e49946ffffff}

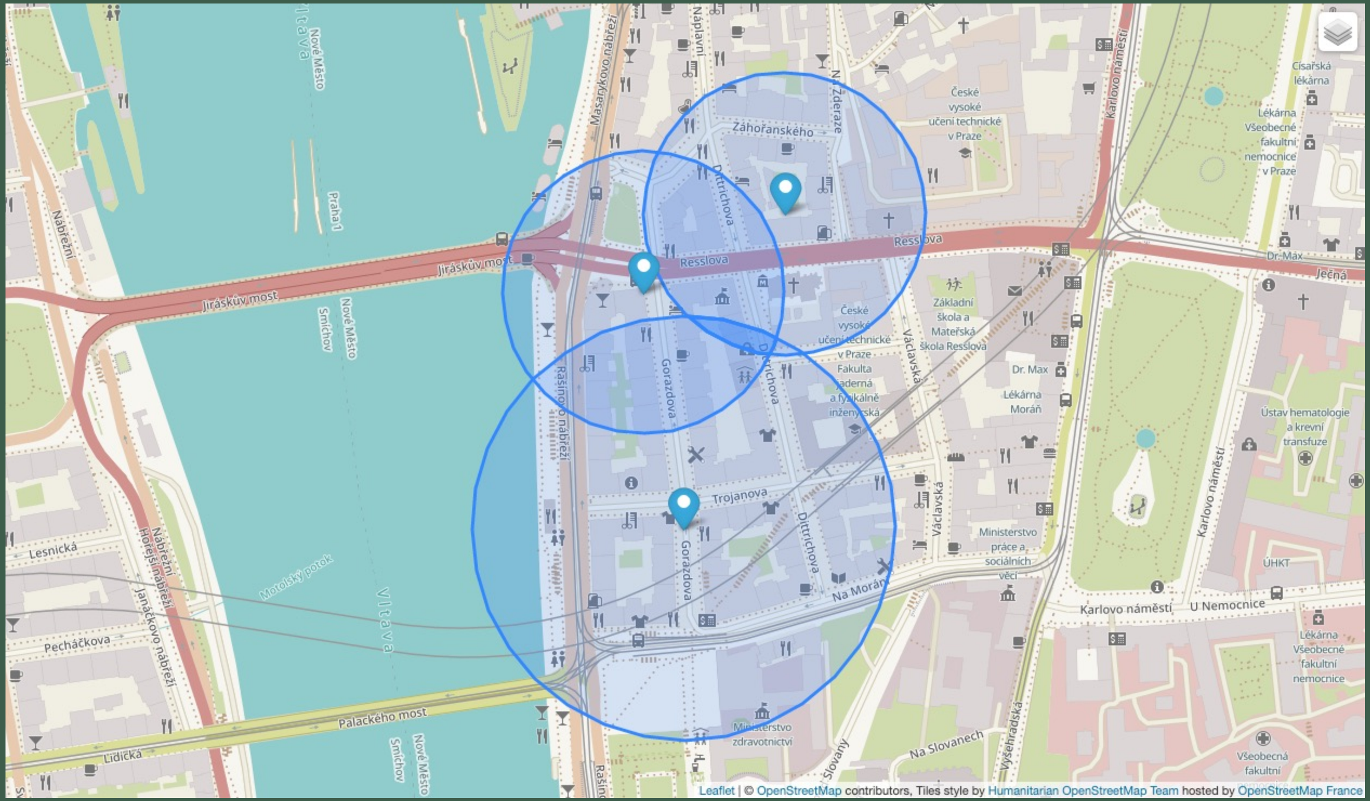
CASE 6: CALCULATING TRILATERATION IN POSTGRESQL

WARDRIVING



Source: https://wigle.net/

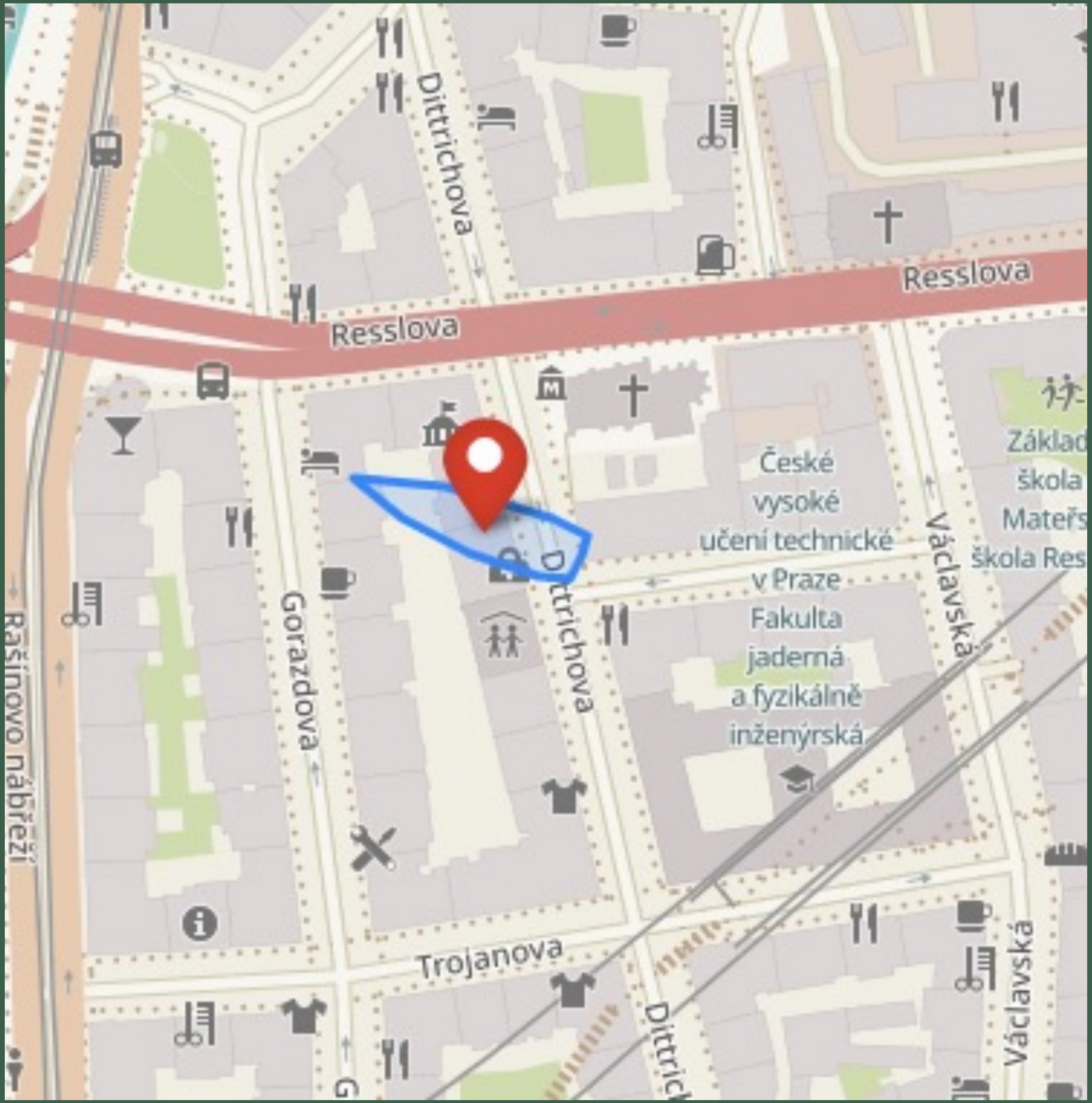
GEOSPATIAL TRIANGULATION




```
SELECT
    id,
    geom,
ST_Buffer(geom::geography,
radius)::geometry AS
circle
FROM spatial.points;
```

TRIANGULATION ANALYSIS: CALCULATING INTERSECTION

```
WITH circles AS
         (
             SELECT id,
                    geom,
                    St_buffer(geom::geography, radius)::geometry AS circle
             FROM spatial.points),
-- Find intersections between each pair of circles intersections
     as
         (
          select c1.id                                AS id1,
             c2.id                                AS id2,
          st_intersection(c1.circle, c2.circle) AS intersection
           FROM       circles c1
           CROSS JOIN circles c2
           WHERE      c1.id < c2.id),
-- Find a common intersection area of all three circles common_intersection
AS
(
       SELECT st_intersection(a.intersection, b.intersection) AS common_area
       FROM   intersections a,
              intersections b
       WHERE  a.id1 = b.id1
       AND    a.id2 != b.id2
       AND    st_intersects(a.intersection, b.intersection) ),
-- Extract the point from the common intersection area and calculate the centroid
triangulation
AS
(
       SELECT (st_dump(st_intersection(common_area, c.circle))).geom AS triangulation_point,
              st_centroid(st_intersection(common_area, c.circle))    AS centroid
       FROM   common_intersection,
              circles c
       WHERE  st_intersects(common_area, c.circle)
       AND    c.id NOT IN
              (
                     SELECT id1
                     FROM   intersections
                     WHERE  id2 = c.id) )
SELECT st_astext(triangulation_point) AS triangulation_point,
       st_astext(centroid)            AS centroid_point
FROM   triangulation limit 1;
```



SUMMARY

Powerful Geodata Handling

PostgreSQL excels in managing and analyzing geospatial data

Rich PostGIS Features

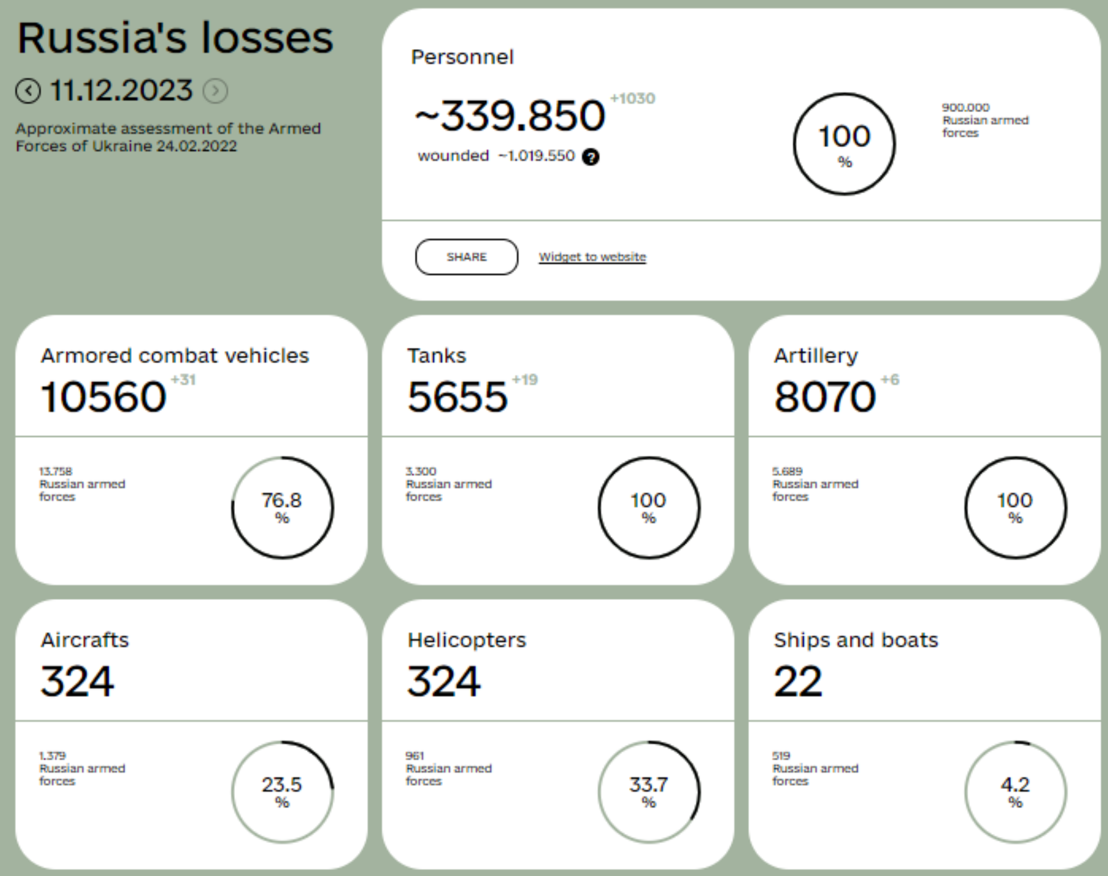
PostGIS provides an extensive toolkit for geoanalytics

Future insights

Looking forward to exploring these advanced topics in upcoming sessions

Beyond the Presentation

Further capabilities like Citus for geo-data sharding, PgRouting, geocoding, and leveraging OpenStreetMap (OSM) data in PostgreSQL



Source: https://www.minusrus.com/en

CONNECT ON SOCIAL MEDIA

Taras Kloba

Associate Director, Big Data & Analytics,
SoftServe Inc.

linkedin.com/in/kloba

ANY QUESTIONS?