Spatial SQL II

CASA0025: Building Spatial Applications
with Big Data

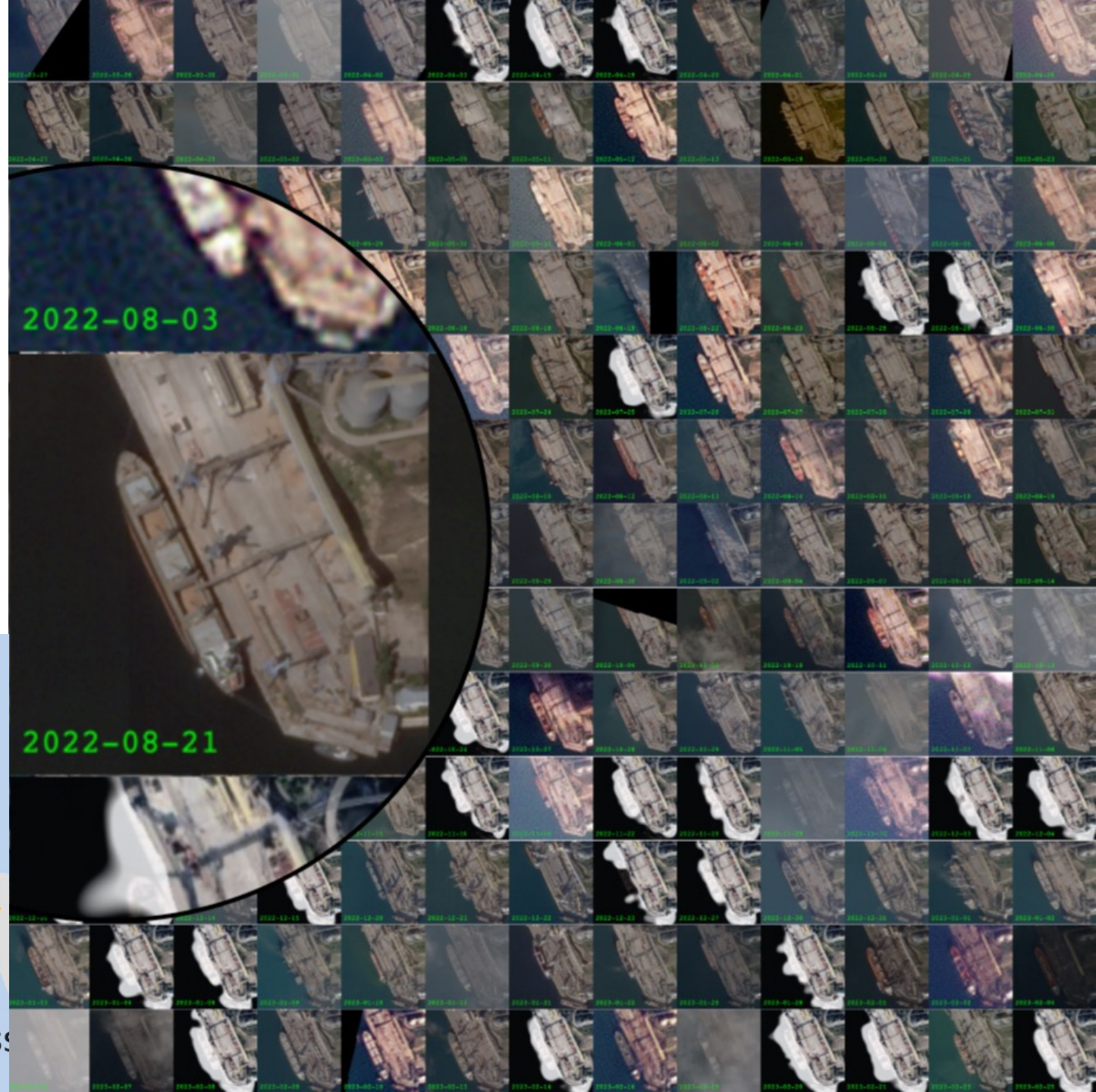
Ollie Ballinger

Grain Theft in Ukraine

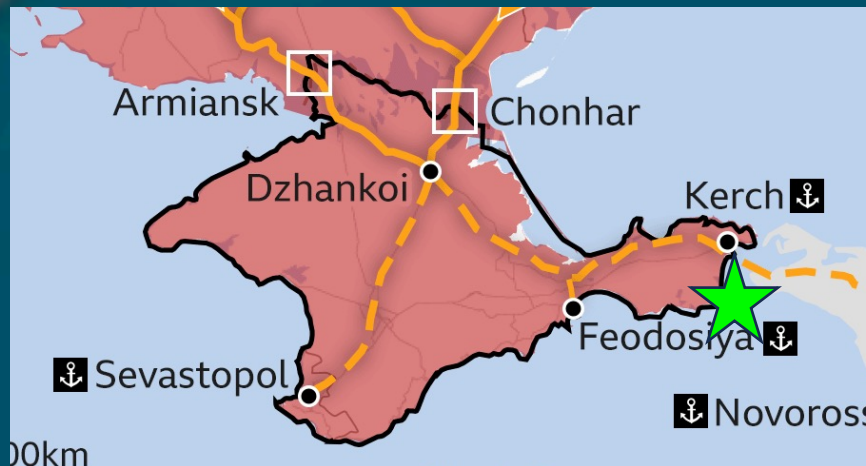
- "They take grain to the annexed Crimea first, where they transport it to Kerch or Sevastopol [ports], then they load Ukrainian grain on Russian ships and go to the **Kerch Strait**. There, in the **Kerch Strait** [between Crimea and Russia], they transfer Ukrainian grain from small ships on to bulk carriers, where it is mixed with grain from Russia - or in some cases, they sail to this area just to give the appearance they are loading up with Russian grain."

Step 1: Load Grain

- Smaller feeder vessels turn off AIS and sail to the Avlita Grain Terminal in Sevastopol.
- Using optical and radar satellite imagery, we identified **180 ships with AIS turned off** at this terminal in the first year of the war.



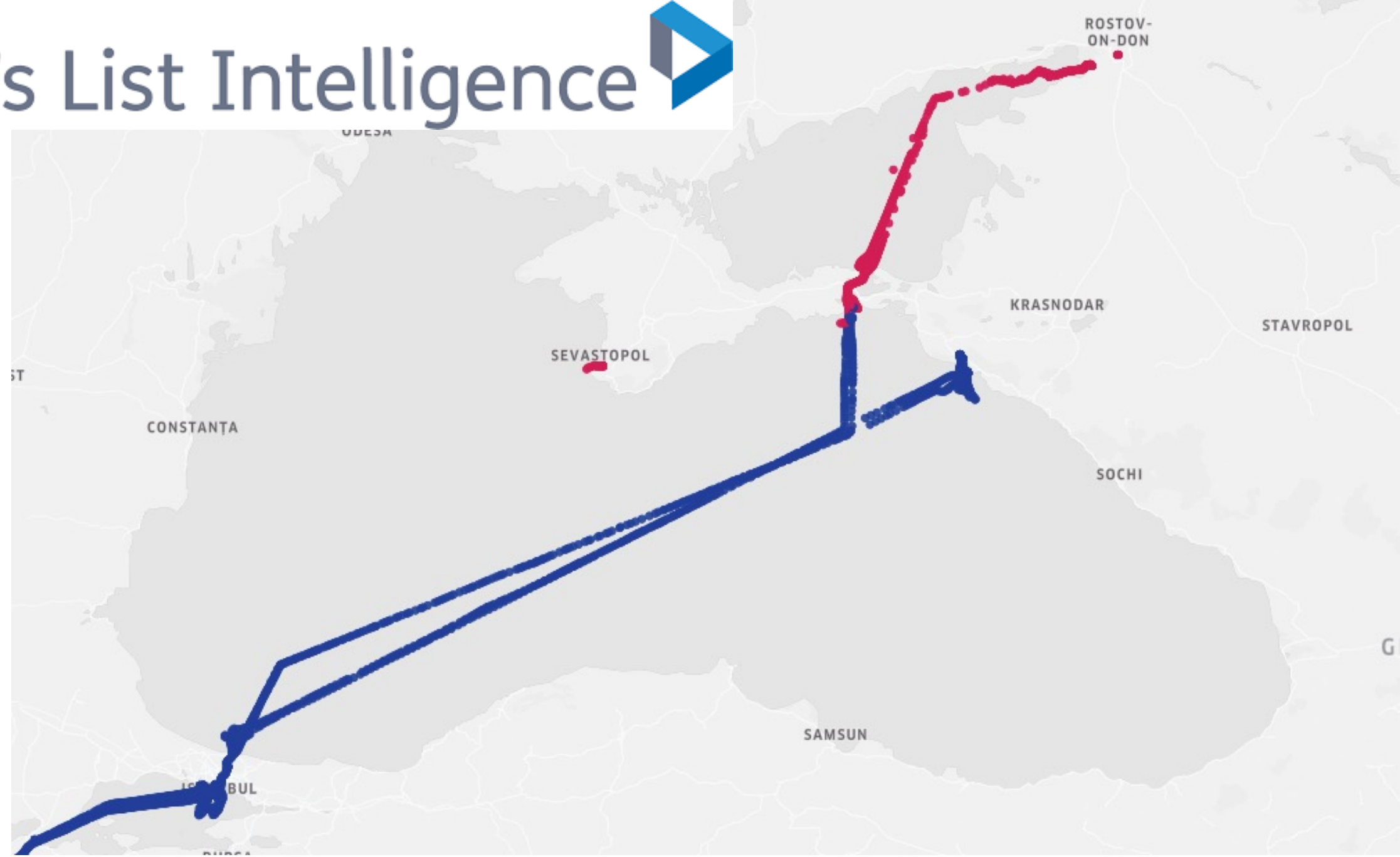
Step 2: Ship-to-Ship Transfer




02 JUN 2018
East China Sea
SHANG YUAN BAO
Hoses connecting SHANG YUAN BAO and
MYONG RYU 1 for ship-to-ship transfer
MYONG RYU 1


ARMAN 114
S TINOS
KRIBI
IMO

Lloyd's List Intelligence



LLOYD's LIST. N° 560

FRIDAY, January 2. 1740.

THIS List, which was formerly publish'd once a Week, will now continue to be publish'd every *Tuesday* and *Friday*, with the Addition of the Stocks Course of Exchange, &c.——Subscriptions are taken in at Three Shillings per Quarter, at the Bar of *Lloyd*'s Coffee-House in *Lombard-Street*.

Such Gentlemen as are willing to encourage this Undertaking, shall have them carefully deliver'd according to their Directions.

London *Exchanges* on

Amst.	34 11 a 10
Ditto Sight	34 7½a8
Rott.	35 a 1
Antw.	35 11 a 36
Hamb:	33 10 2Ua11 2½
Paris —	32 5/8
Ditto at 2U	32 3/8
Bourdeaux ½ Usance	32 5/8
Cadiz —	42 5/8
Madrid	42 5/8
Bilboa	41 5/8
Leghorn	51¼
Genoa	55
Venice	51¼
Lisbon	5 4 7/8a5
Oporto	5 4½
Dublin	8

Aids in the Exchequer		*Given for*	*Paid off*
18th 2 Shilling	1739	1000000	926800
18th 4 Ditto	1740	2000000	482600
Malt——	1739	750000	501014
Salt——	1734	1000000	910500

Gold in Coin	3	18	1
Ditto in Barrs	3	18	
Ps of Eight: Pillar large	per Oz 0	5	7 5/8
Ditto Small	per Oz 0	5	6 1/2
Mexico large	per Oz 0	5	7 1/2
Ditto Small	per Oz 0	5	6 5/8
Silver in Barrs	0	5	7 5/8

Annuities

14*l.* per Cent at 22¼ Years Purchase
1704 to 1708 Inclusive 24¼ ditto
3½ per Cent. 1 per Cent. præm.
3 per Cent. 5¼ Disc.

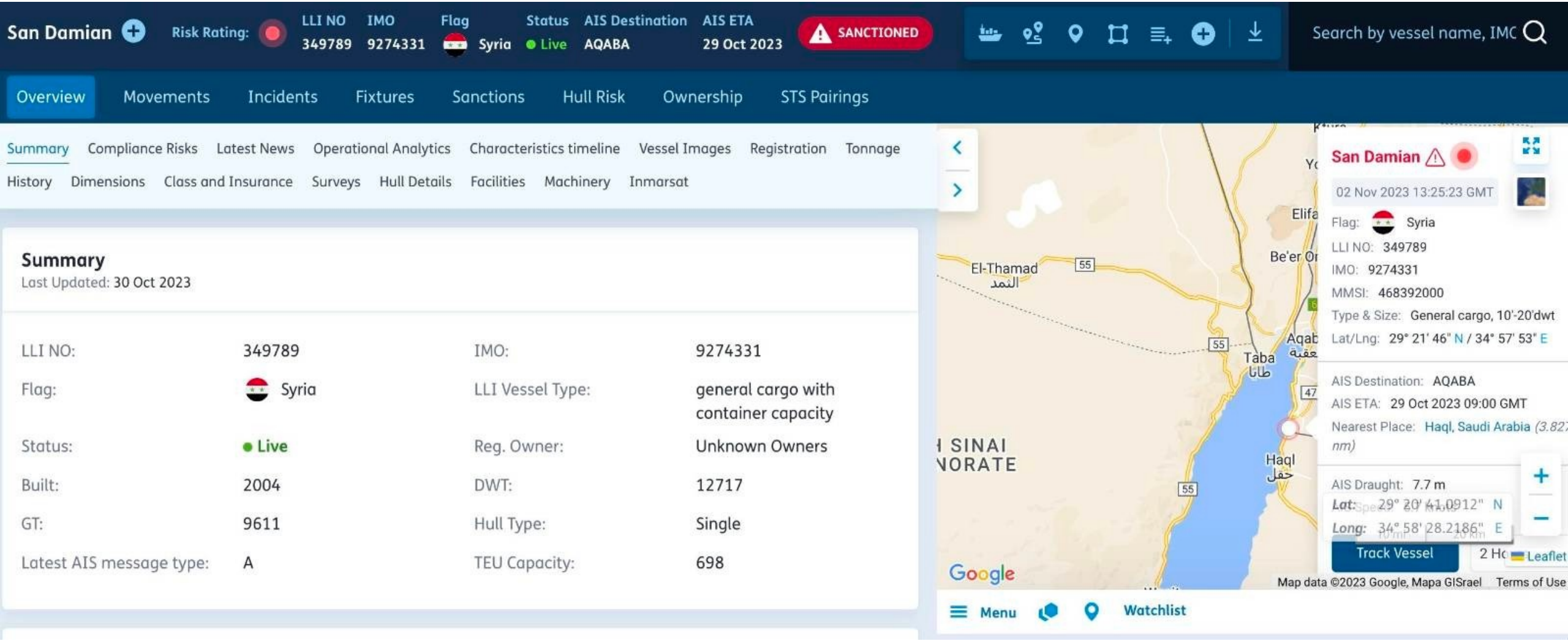
Lottery 1710.

Prizes for 3 Years from *Michaelmas* last are in course of Payment
Blanks for 3 Years from *Michaelmas* last 1l. 10s *per* Set.

Price of Stocks	*Wednesday*	*Thursday*	*Friday*
Bank Stock	138½a¾		138½
East India		156	156a56¼
South Sea	98¼		98¼
Ditto Anuity Old	110½a10	110½	110½
Ditto——New	110½a¾	110¼	110¼
3 *per* Cent. 1726			
Annuity 1731			99⅝
Million Bank	113	113	113
Equivalent	112	112	112
R. Aff. 100l paid in			
L: Aff. 13l paid in	10¼	10¼	10¼
7 per Cent Em. Loan	98	98	98
5 per Cent. Ditto	74¾	74¾	75
Bank Circulation	2l 10s od	2l 10s od	2l 10s od
Lottery Tickets	5l 16s od	5l 17s od	6l 00s od

India Transfer Books open the 19th of January
Royal Assurance the 20th of January
South Sea New Annuity the 22d of January, 3 per Cent Annuities the 21st and 22d of January
South Sea Stock the 4th of February
The 5 per Cent Emperor's Loan, sells as above without the six Months Interest of 2 and a quarter per Cent, and 5 per Cent. part of the Principal to be paid of both, are now paying at the Bank
The India Dividend will be paid the 29th of January, South Sea New Annuities the 29th ditto, and the S. Sea Stock the 6th and 7th of February,
Navy and Victualling Bills to the 30th June last are in course of Payment.

Interest per Cent	*Wednesday*	*Thursday*	*Friday*
3 India Bonds new	79	80	80 Shill: Præm:
4 Salt Tallies	¼ a ⅜	¼ a ⅜	¼ a ⅜

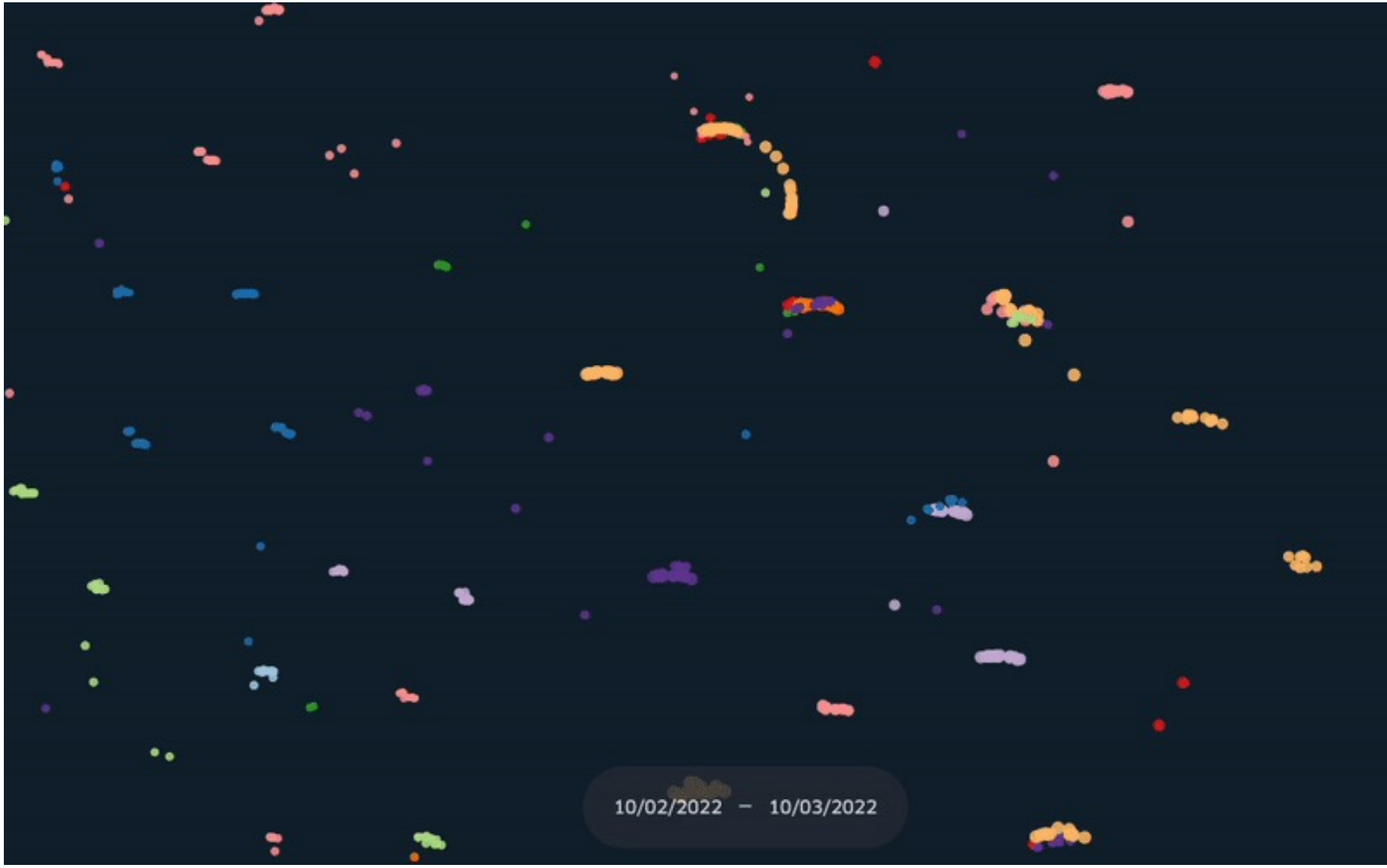

San Damian
Risk Rating:
LLI NO 349789
IMO 9274331
Flag Syria
Status Live
AIS Destination AQABA
AIS ETA 29 Oct 2023
SANCTIONED
Search by vessel name, IMC
Overview
Movements
Incidents
Fixtures
Sanctions
Hull Risk
Ownership
STS Pairings
Summary
Compliance Risks
Latest News
Operational Analytics
Characteristics timeline
Vessel Images
Registration
Tonnage
History
Dimensions
Class and Insurance
Surveys
Hull Details
Facilities
Machinery
Inmarsat
Summary
Last Updated: 30 Oct 2023
LLI NO: 349789
IMO: 9274331
Flag: Syria
LLI Vessel Type: general cargo with container capacity
Status: Live
Reg. Owner: Unknown Owners
Built: 2004
DWT: 12717
GT: 9611
Hull Type: Single
Latest AIS message type: A
TEU Capacity: 698
San Damian
02 Nov 2023 13:25:23 GMT
Flag: Syria
LLI NO: 349789
IMO: 9274331
MMSI: 468392000
Type & Size: General cargo, 10'-20'dwt
Lat/Lng: 29° 21' 46" N / 34° 57' 53" E
AIS Destination: AQABA
AIS ETA: 29 Oct 2023 09:00 GMT
Nearest Place: Haql, Saudi Arabia
AIS Draught: 7.7 m
Lat: 29° 20' 41.0912" N
Long: 34° 58' 28.2186" E
Track Vessel
El-Thamad
Taba
Haql
Google
Map data ©2023 Google, Mapa GISrael
Terms of Use
Leaflet
Menu
Watchlist

imo	vessel_name	vsl_descr	dwt	v_length	vesselid	draught	date	geometry
9372846.0	15 Temmuz	general cargo	7827.0	113.0	378595	4.4	2021-09-12 01:29:54	POINT (4063634.892855323 5636163.296415932)
9372846.0	15 Temmuz	general cargo	7827.0	113.0	378595	4.4	2021-09-12 01:36:43	POINT (4063613.00002214 5638822.792206503)
9372846.0	15 Temmuz	general cargo	7827.0	113.0	378595	4.4	2021-09-12 01:43:43	POINT (4062958.626948753 5641425.484822003)
9372846.0	15 Temmuz	general cargo	7827.0	113.0	378595	6.9	2021-09-17 00:38:34	POINT (4062702.035522474 5638123.540483131)
8956188.0	30 Let Pobedy	general cargo	5150.0	NaN	350053	3.4	2021-06-05 02:26:10	POINT (4064299.099150392 5643560.996257269)
...	...	...	...	...	...	...	...	...
9232151.0	Zeynep C	bulk carrier	53806.0	183.0	312965	6.2	2023-08-16 13:32:48	POINT (4071296.271647753 5646968.081009985)
9232151.0	Zeynep C	bulk carrier	53806.0	183.0	312965	6.2	2023-08-16 13:35:52	POINT (4070342.263611655 5643114.654042991)
9232151.0	Zeynep C	bulk carrier	53806.0	183.0	312965	6.2	2023-08-16 13:36:03	POINT (4070341.891804556 5643113.601468305)
9232151.0	Zeynep C	bulk carrier	53806.0	183.0	312965	6.2	2023-08-16 14:39:18	POINT (4071619.468864959 5645507.330939625)
9232151.0	Zeynep C	bulk carrier	53806.0	183.0	312965	6.2	2023-08-19 11:36:03	POINT (4070322.967491121 5643163.574441582)

Identifying STS in AIS data

A simple algorithm:

- **If** two or more ships
- **Spend** two or more hours
- **Within** 500m of each other
- **Then** link them together



```sql
SELECT
    a1.vesselid AS ship1,
    a2.vesselid AS ship2,
    a1.date AS start,
    a2.date AS end
FROM
    ais AS a1
JOIN
    ais AS a2
ON ST_DWithin(a1.geom, a2.geom, 500)
     AND ABS(EXTRACT(EPOCH FROM (a2.date - a1.date))) > 7200;
     AND a1.vesselid <> a2.vesselid
```

Spatial Joins

What neighborhood is the 'Broad St' station in?

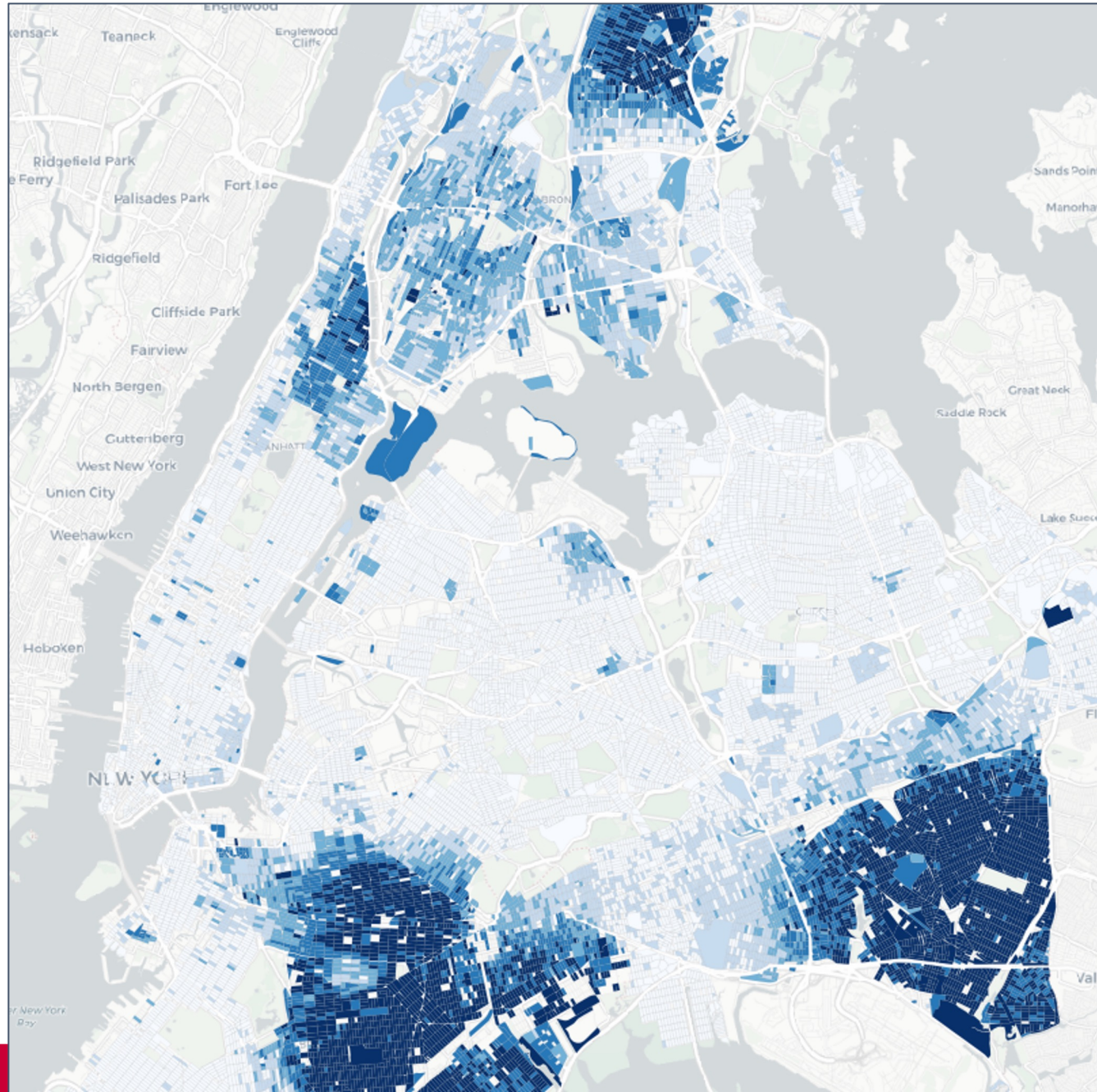
Remember...

```
SELECT name, boroname
FROM nyc_neighborhoods
WHERE
   ST_Intersects(
      geom,
      ST_GeomFromText(
         'POINT(583571 4506714)',
         26918));
```

Do it in one step, with a spatial join!

```
SELECT s.name, n.name, n.boroname
FROM nyc_neighborhoods AS n
JOIN nyc_subway_stations AS s
  ON ST_Contains(
     n.geom,
     s.geom
  )
WHERE s.name = 'Broad St';
```

What is the population and racial make-up of the neighborhoods of Manhattan?

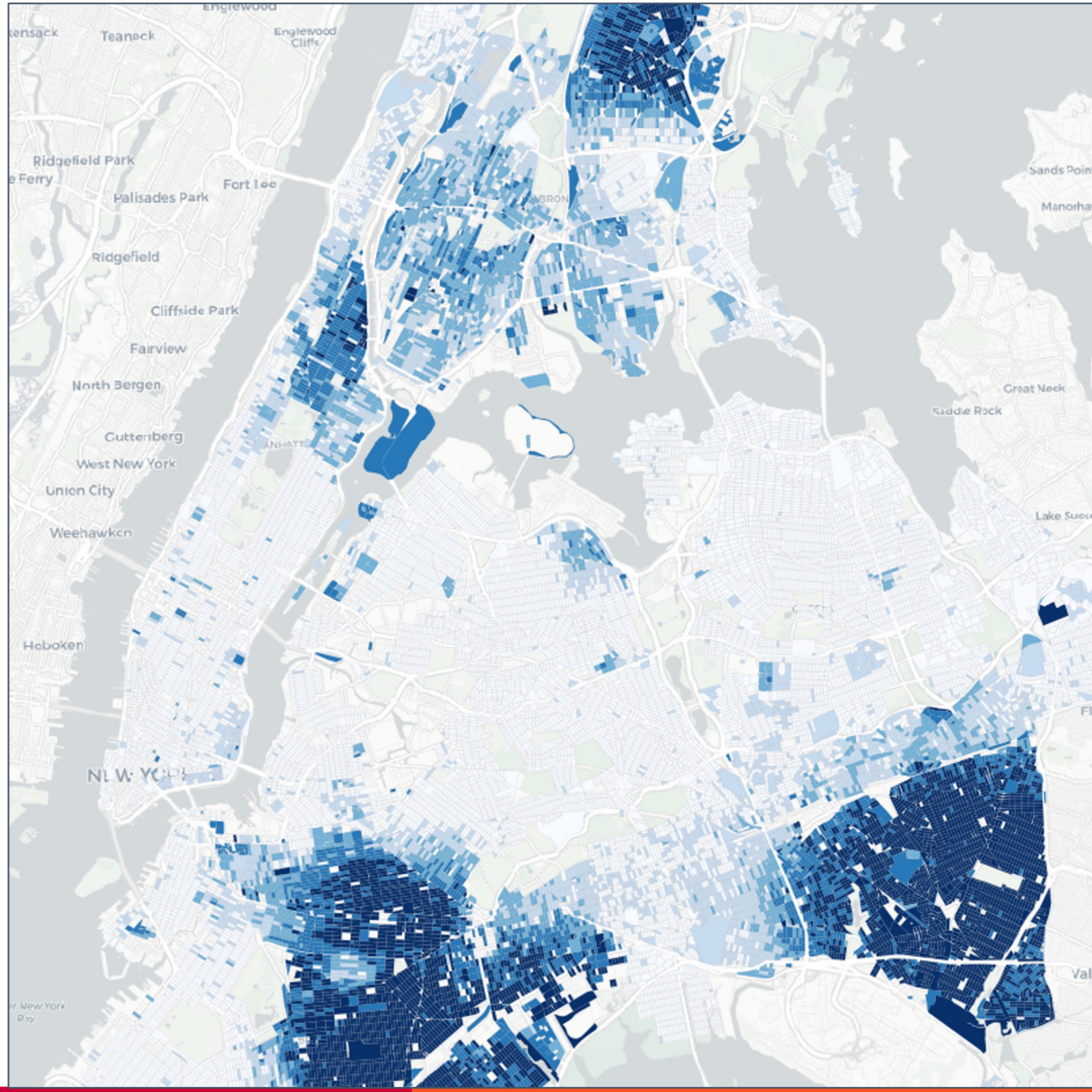


```sql
SELECT
  n.name AS neighborhood_name,
  SUM(c.popn_total) AS population,
  100*SUM(c.popn_white)/SUM(c.popn_total) AS white_pct,
  100*SUM(c.popn_black)/SUM(c.popn_total) AS black_pct
FROM nyc_neighborhoods AS n
JOIN nyc_census_blocks AS c
  ON ST_Intersects(
    n.geom,
    c.geom
  )
WHERE n.boroname = 'Manhattan'
GROUP BY n.name
ORDER BY white_pct DESC;
```

neighborhood_name	popn	white %	black %
Carnegie Hill	18763	90.1	1.4
West Village	26718	87.6	2.2
North Sutton Area	22460	87.6	1.6
Upper East Side	203741	85.0	2.7
Soho	15436	84.6	2.2
Greenwich Village	57224	82.0	2.4
Central Park	46600	79.5	8.0
Tribeca	20908	79.1	3.5
Gramercy	104876	75.5	4.7
Murray Hill	29655	75.0	2.5
Chelsea	61340	74.8	6.4
Upper West Side	214761	74.6	9.2
Midtown	76840	72.6	5.2
Battery Park	17153	71.8	3.4

neighborhood_name	popn	white %	black %
Financial District	34807	69.9	3.8
Clinton	32201	65.3	7.9
East Village	82266	63.3	8.8
Garment District	10539	55.2	7.1
Morningside Heights	42844	52.7	19.4
Little Italy	12568	49.0	1.8
Yorkville	58450	35.6	29.7
Inwood	50047	35.2	16.8
Washington Heights	169013	34.9	16.8
Lower East Side	96156	33.5	9.1
East Harlem	60576	26.4	40.4
Hamilton Heights	67432	23.9	35.8
Chinatown	16209	15.2	3.8
Harlem	134955	15.1	67.1

Let's explore the racial geography of New York City...



Overall Racial Make-up of NYC

```
SELECT
  100.0*SUM(popn_white)/SUM(popn_total) AS white_pct,
  100.0*SUM(popn_black)/SUM(popn_total) AS black_pct,
  SUM(popn_total) AS popn_total
FROM nyc_census_blocks;
```

```
     white_pct      |     black_pct      | popn_total
--------------------+--------------------+------------
 44.00395007628105 | 25.546578900241613 |    8175032
```

What is the racial make-up of the areas served by the A train?

First, we must determine where the A train stops.

Our routes are comma-separated strings!

```
SELECT DISTINCT routes
FROM nyc_subway_stations;
```

```
4,5
N,Q,R,W
J
B,M,Q,R
D,F,N,Q
J,M
E,F
...
```

Postgres string function: strpos()

`strpos(routes,'A')` returns a non-zero number if 'A' is in the `routes` field

Check out https://www.postgresql.org/docs/current/functions-string.html

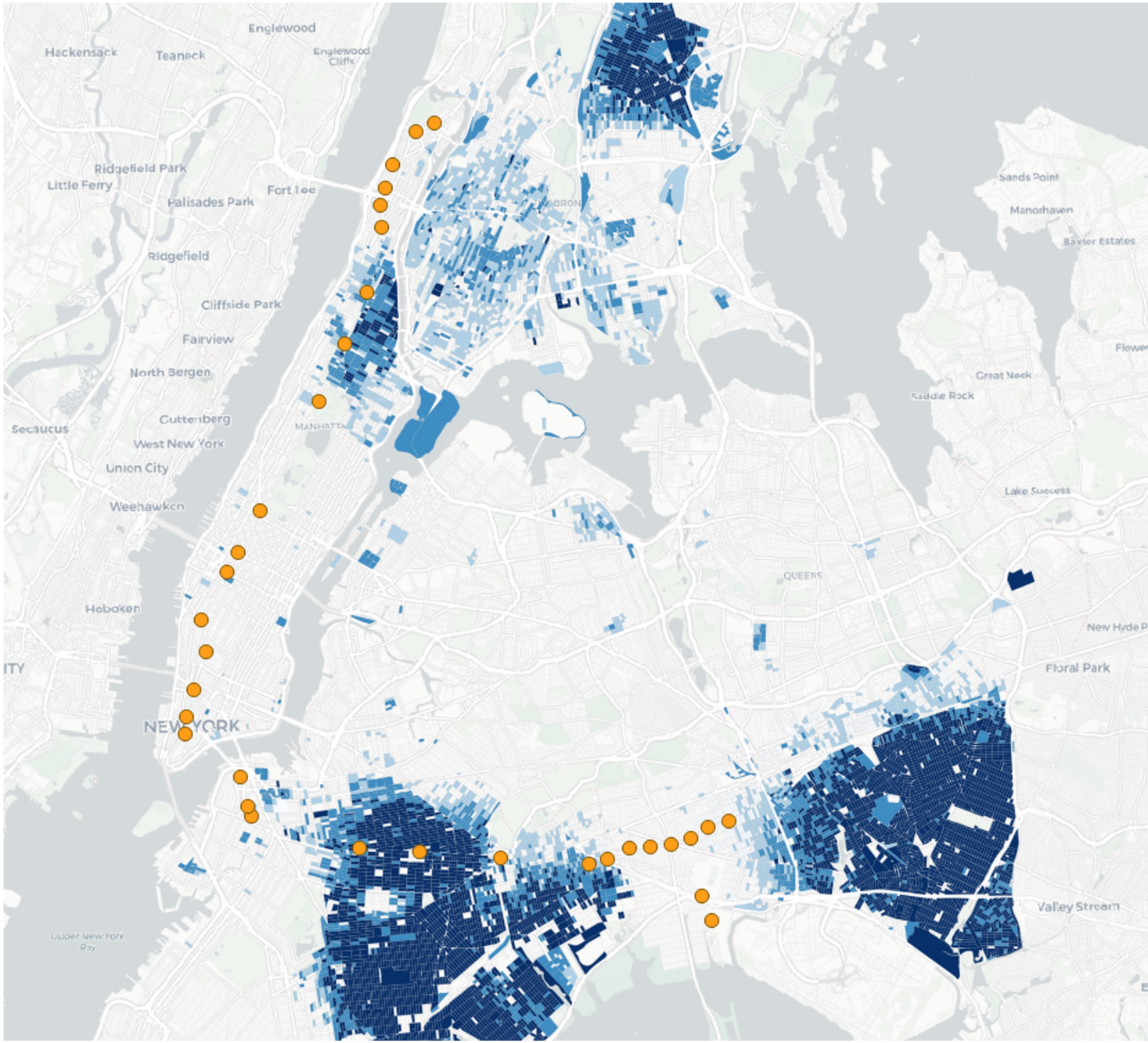
Find all routes with an “A”

```
SELECT DISTINCT routes
FROM nyc_subway_stations AS subways
WHERE strpos(subways.routes,'A') > 0;
```

```
A,C
A,B,C,D
A,C,E,L
A,C,F
A,B,C
A,S
A,C,E
...
```

The route of the A train.

What is the racial makeup within 200 meters of each stop? Who is served by the A train?



Summarize population 200m from A train stops

```
SELECT
  100*SUM(c.popn_white)/SUM(c.popn_total) AS white_pct,
  100*SUM(c.popn_black)/SUM(c.popn_total) AS black_pct,
  SUM(popn_total) AS popn_total
FROM nyc_census_blocks AS c
JOIN nyc_subway_stations AS s
  ON ST_DWithin(
    c.geom,
    s.geom,
    200
  )
WHERE strpos(s.routes,'A') > 0;
```

New York

44.00% white

25.55% black

A Train

45.59% white

22.09% black

Spatial Indexing

A spatial database has…

- **Spatial Data Types**
  - geometry, geography
- **Spatial Indexes**
  - r-tree, quad-tree, kd-tree
- **Spatial Functions**
  - ST_Length(geometry), ST_X(geometry)

- NOTE: Currently, only PostGIS supports spatial indexing; DuckDB does not (but it’s under development)!

A spatial index speeds spatial query …

- **Join two tables of 10,000 records each**

Without Index	With Index
10,000 * 10,000 = **100,000,000** comparisons	10,000 + 10,000 = **20,000** comparisons

To prove it... remove the index.

```
DROP INDEX nyc_census_blocks_geom_idx;
```

Run a spatial join.

```
SELECT b.blkid
FROM nyc_census_blocks b
JOIN nyc_subway_stations s
  ON ST_Contains(b.geom, s.geom)
WHERE s.name LIKE 'B%';
```

~300 ms

Create the index again.

```
CREATE INDEX nyc_census_blocks_geom_idx
ON nyc_census_blocks USING GIST (geom);
```

Run the join again.

```
SELECT blocks.blkid
FROM nyc_census_blocks b
JOIN nyc_subway_stations s
  ON ST_Contains(b.geom, s.geom)
WHERE s.name LIKE 'B%';
```

~90 ms

Spatial Index Cliff Notes

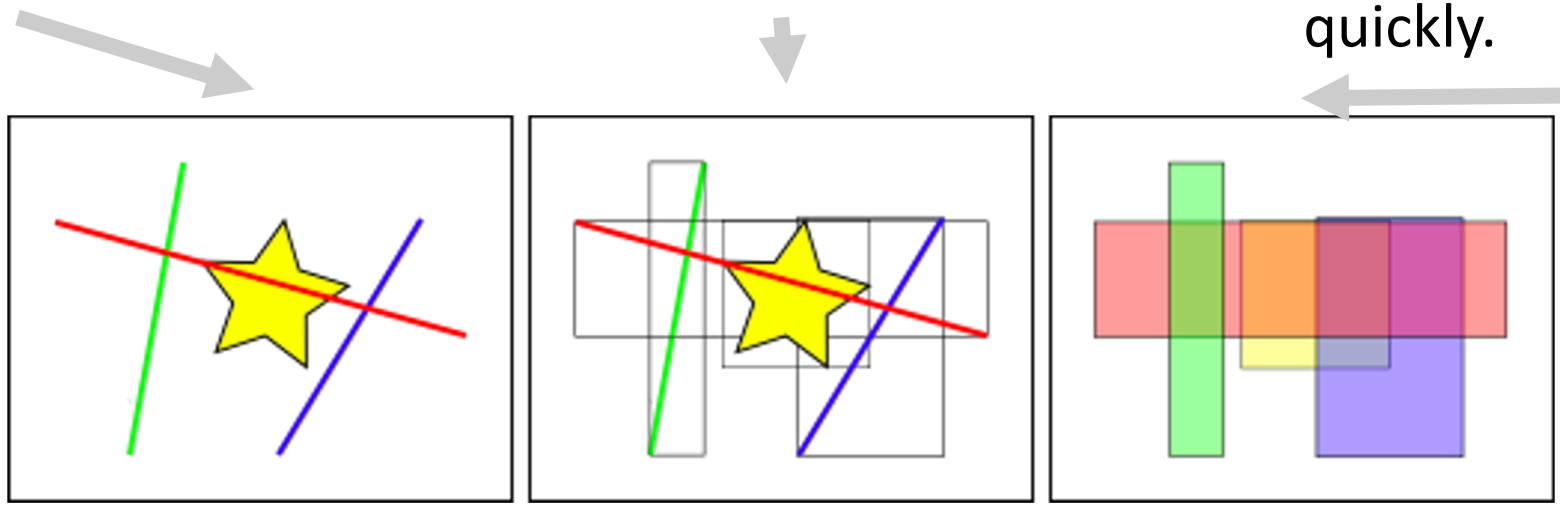
- CREATE INDEX index_name ON table_name USING GIST (geom)
- Use a “spatially indexed function” in JOIN or WHERE clause
  - ST_Intersects(A, B), ST_Contains(A, B), ST_Within(A, B)
  - ST_DWithin(A, B, R)

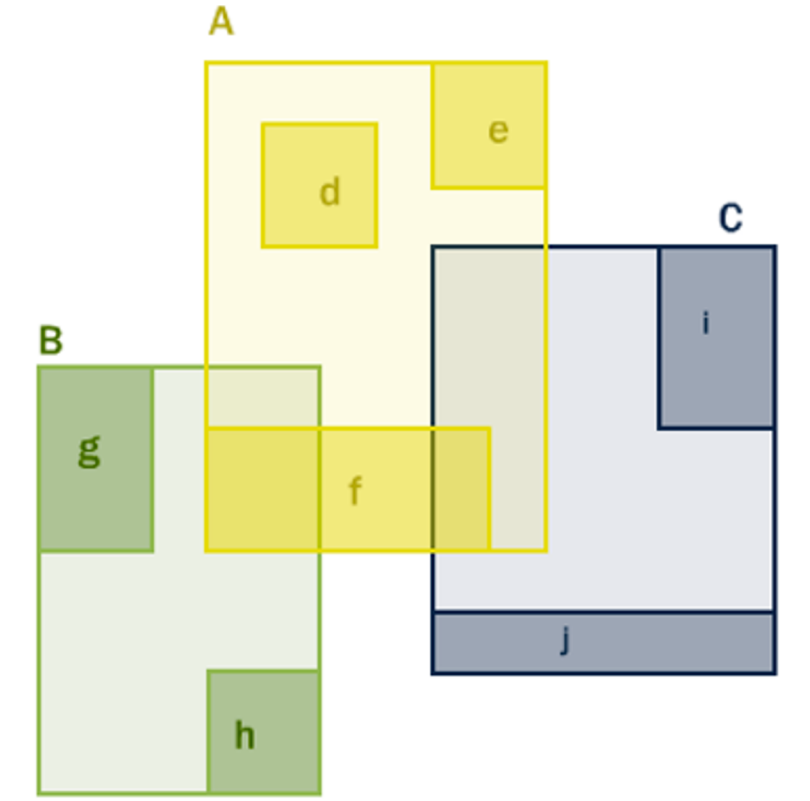
How do spatial indexes work?

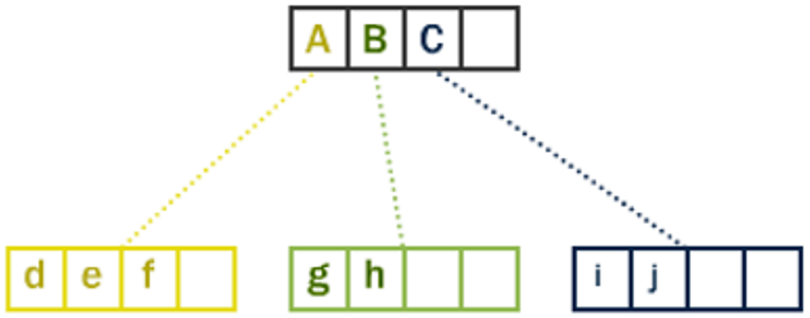
Some spatial objects (like the star) are quite large and complex. Comparing complex objects is expensive!

Instead of indexing objects directly, spatial indexes work on the bounding boxes of the objects.

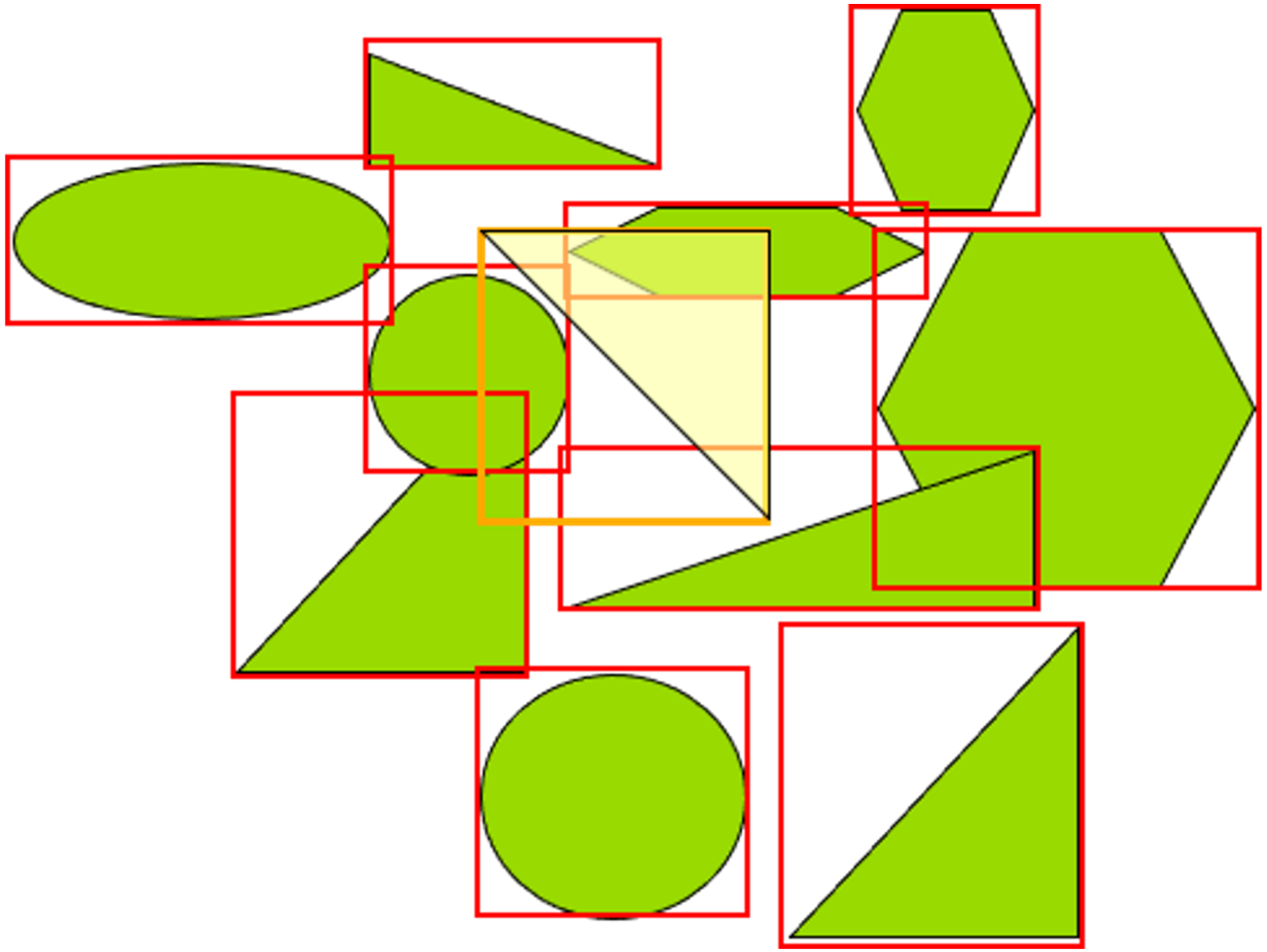
The boxes are of uniform size, and can be compared to determine spatial relationships very quickly.

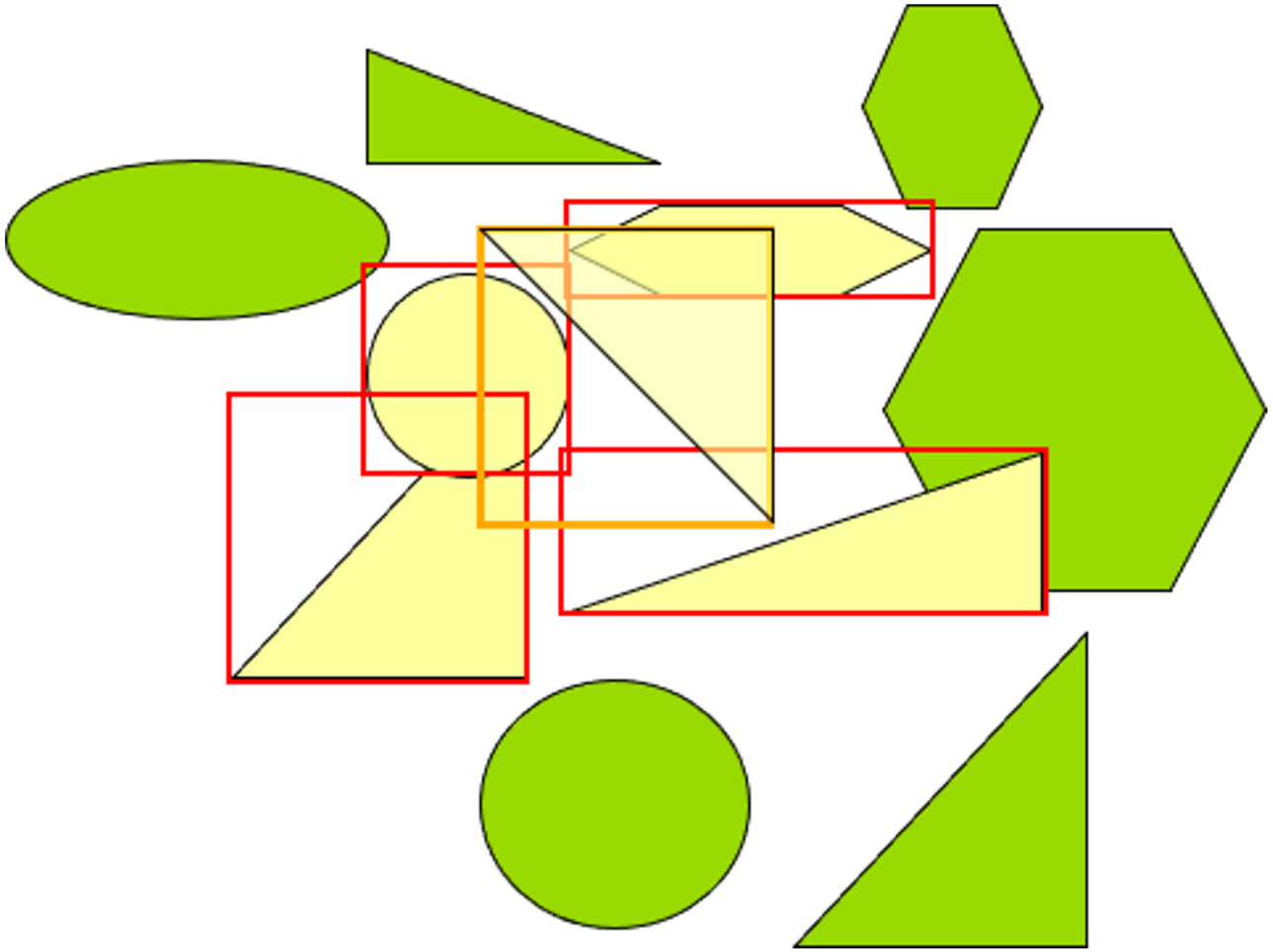


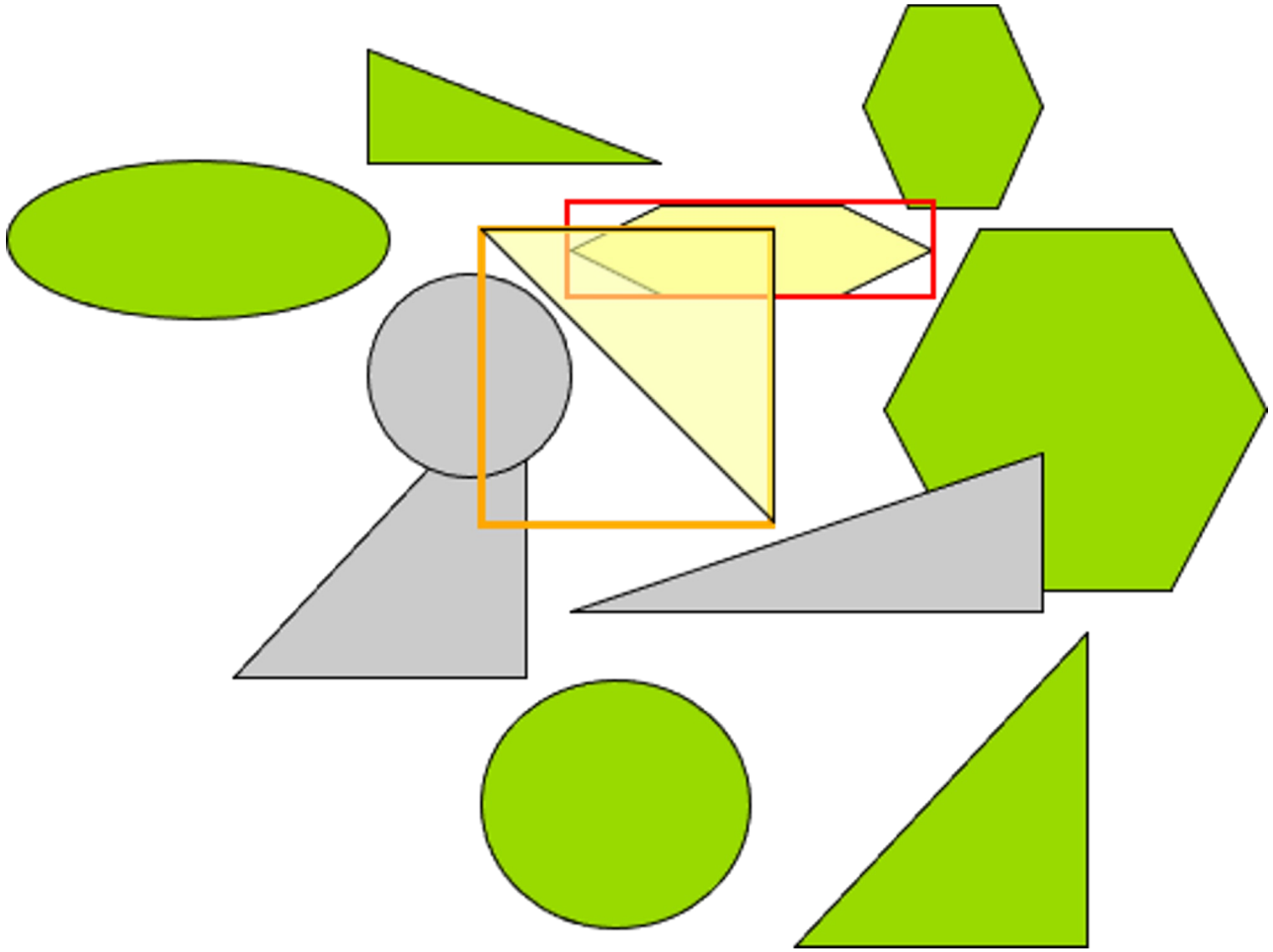




The boxes can be arranged in a hierarchy, so that a query can quickly discard portions of the search space that will not interact with a query box. Depending on the algorithm, different hierarchies can be build. PostGIS uses an “R*tree” algorithm.







Index-enabled Spatial Functions

- **ST_Intersects()**
- **ST_Contains()**
- **ST_Within()**
- **ST_DWithin()**
- ST_ContainsProperly()
- ST_CoveredBy()
- ST_Covers()
- ST_Overlaps()
- ST_Crosses()
- ST_DFullyWithin()
- ST_3DIntersects()
- ST_3DDWithin()
- ST_3DDFullyWithin()
- ST_LineCrossingDirection()
- ST_OrderingEquals()
- ST_Equals()

Index-only queries

geom_a && geom_b

The “&&” operator is the “bounding boxes overlap” operator.

It returns “true” when the bounds of the left and right arguments overlap.

Operators like “=” or “>” are symbols that express relationships between the left- and right-hand side arguments. “&&” is just another operator like any other.

What is the population of the West Village?

```
SELECT Sum(blk.popn_total)
FROM nyc_neighborhoods nh
  JOIN nyc_census_blocks blk
  ON nh.geom && blk.geom
WHERE nh.name = 'West Village';
```

49821

What is the population of the West Village?

```
SELECT Sum(blk.popn_total)
FROM nyc_neighborhoods nh
   JOIN nyc_census_blocks blk
   ON ST_Intersects(nh.geom, blk.geom)
WHERE nh.name = 'West Village';
```

26718

Spatial Indexes

```
VACUUM ANALYZE nyc_census_blocks;
```

The database planner can only know when to use indexes if the tables have been “analyzed”.

The database executor will run more efficiently if dead tuple bloat has been removed with “vacuum”. Most important on tables with bulk data changes (insert/update/delete).

Any Questions?

o.ballinger@ucl.ac.uk