Geospatial data acquisition & analysis

With KnowWhereGraph & QGIS

Part 1:
Geospatial data acquisition

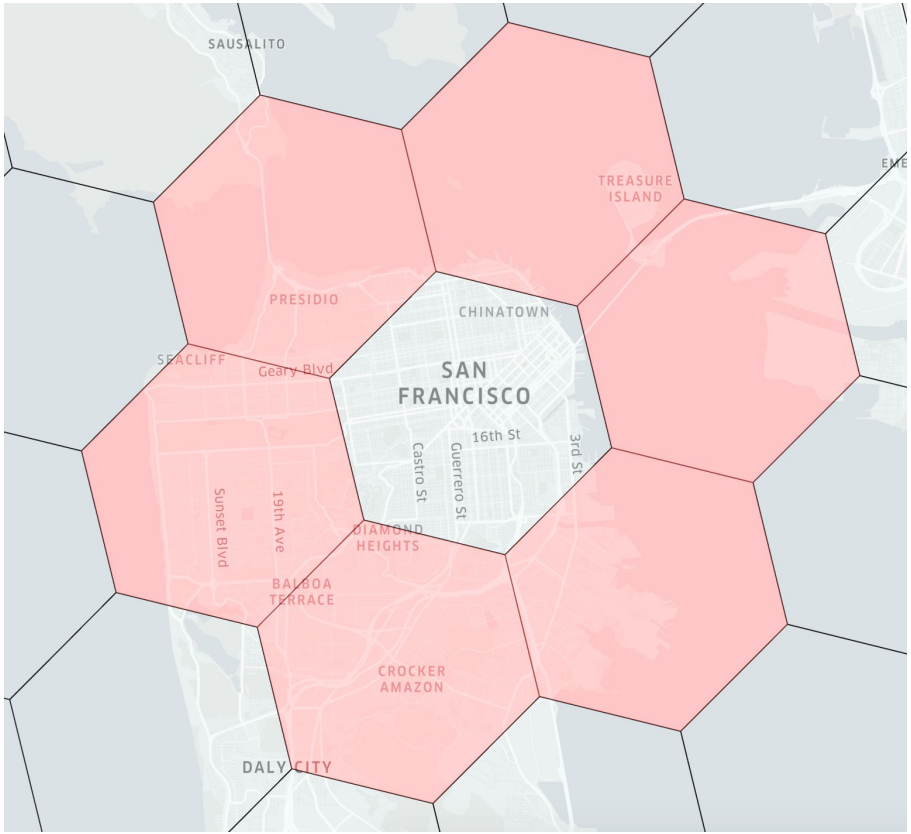
With KnowWhereGraph

Spatial Indexing

- Data structures for fast geospatial lookup
- Tree data structures commonly used in databases (R-Tree, KD-Tree, Quadtree, Octree)
  - Not the kind we're going to use here
- Grid systems
  - Hierarchical grids that cover the globe
    - Can query for fine grained areas, or larger ones
  - Can use spatial relations (Regional Connection Calculus) with grids
    - Paraphrased relations:
      - Grids can touch each other, overlap, be contained within, etc
        - These relations trickle down into queries

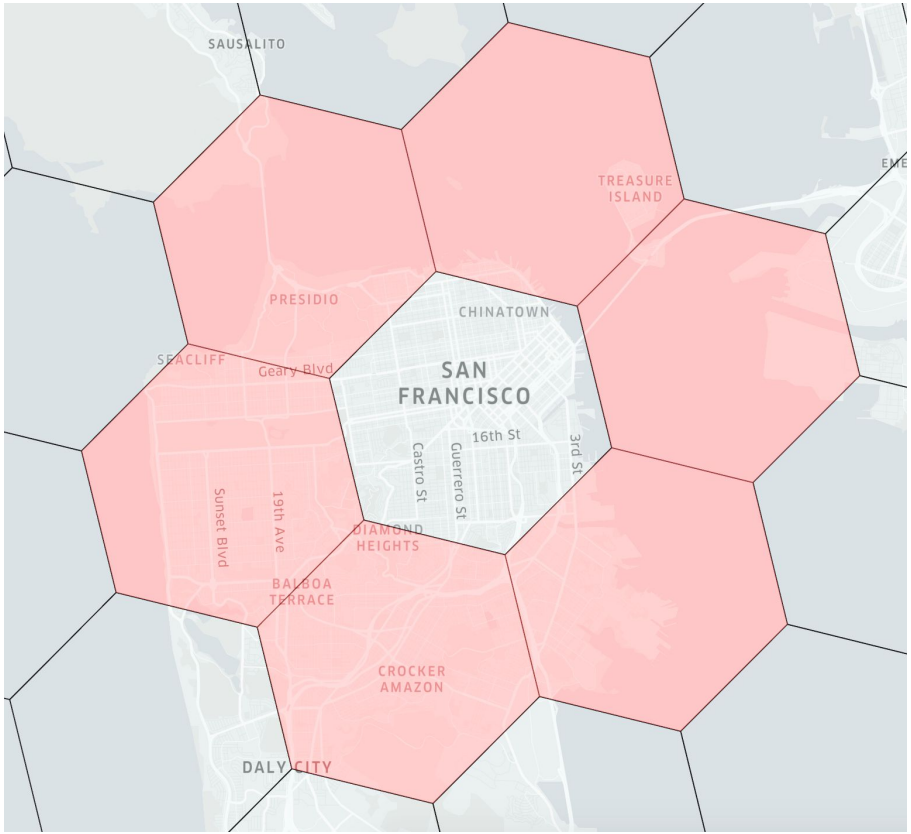
Spatial Indexing: H3

- Spatial index from Uber
- Each hexagon has a unique id
- Imagine you're in the center, on google maps, asking for nearby restaurants
- If you were an engineer at google how would you find nearby restaurants?



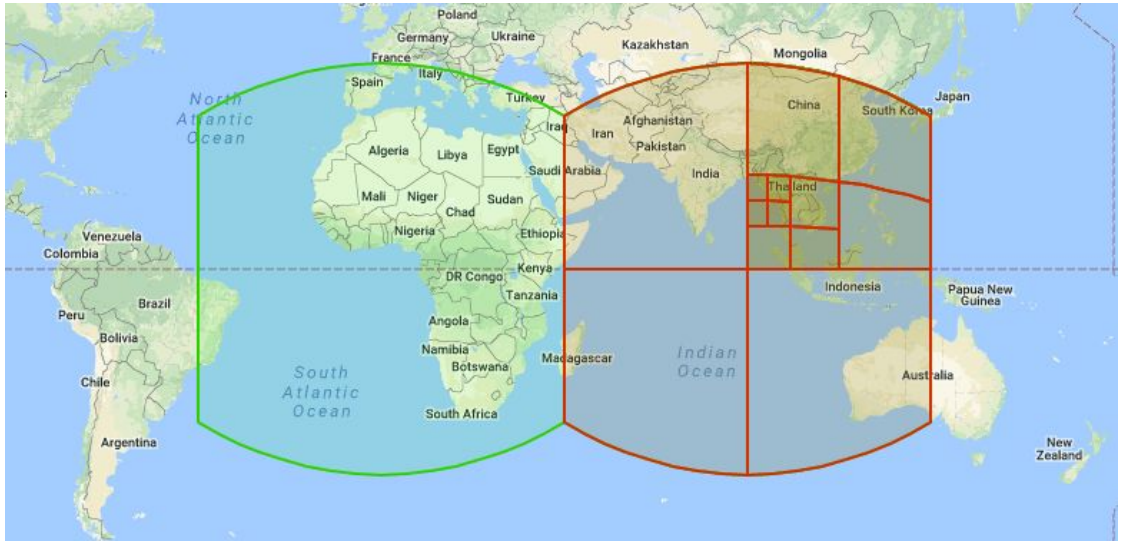
Spatial Indexing: H3

- The most common grid index?
- Each hexagon has a unique id
- Imagine you're in the center, on google maps, asking for nearby restaurants
- If you were an engineer at google how would you find nearby Restaurants?
- Find restaurants within the center hexagon, then find restaurants within the cells connected to the center one. Suggest these last



Spatial Indexing: S2 (What KWG uses)

- Spatial index from Google
- Square-ish rather than hex grids (Hilbert curve)
- Same ideas as H3
- Smaller squares are within bigger squares
    - This allows for more specific Queries
    - Allows for more general (what are the restaurants within the s2 cell that contains the one I'm in)



KnowWhereGraph Geospatial Backbone

- Uses S2 as the index
  - Queries can ask for things contained in
    - Cells that touch another
    - Cells that are contained in another
    - Maybe more?
- Uses the concept of administrative regions in a similar way
  - AdministrativeRegion_0: The world
  - AdministrativeRegion_1: Nation level
  - AdministrativeRegion_2: States within a nation
  - AdministrativeRegion_3: County/district within a state
  - AdministrativeRegion_{4/5/6}: Regions within counties

- Imagine doing RCC with administrative regions

KnowWhereGraph Admin Regions Example: 1

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX kwg-ont: <http://stko-kwg.geog.ucsb.edu/lod/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select * where {
    ?state rdf:type kwg-ont:AdministrativeRegion_2 .
    ?state rdfs:label ?label .
} limit 100
```

	state	label
1	kwgr:Earth.North_America.United_States.USA.10_1	"Florida"
2	kwgr:Earth.North_America.United_States.USA.11_1	"Georgia"
3	kwgr:Earth.North_America.United_States.USA.12_1	"Hawaii"
4	kwgr:Earth.North_America.United_States.USA.13_1	"Idaho"
5	kwgr:Earth.North_America.United_States.USA.14_1	"Illinois"
6	kwgr:Earth.North_America.United_States.USA.15_1	"Indiana"

KnowWhereGraph Admin Regions Example

What's in a State node?

kwgr:Earth.North_America.United_States.USA.10_1	kwg-ont:sfWithin	kwgr:Earth.North_America.United_States.USA
kwgr:Earth.North_America.United_States.USA.10_1	geosparql:hasDefaultGeometry	kwgr:geometry.multipolygon.North_America.United_Sta
kwgr:Earth.North_America.United_States.USA.10_1	rdf:type	kwg-ont:AdministrativeRegion_2
kwgr:Earth.North_America.United_States.USA.10_1	rdf:type	geosparql:Feature
kwgr:Earth.North_America.United_States.USA.10_1	rdfs:label	"Florida"
kwgr:Earth.North_America.United_States.USA.10_1	owl:sameAs	wd:Q812

KnowWhereGraph Spatial Types

- Things you can ask for
- Big idea (the good): Spatial *things* are one way or another geosparql:SpatialObject
  - Example: usgs:Hospital is a subclass of a BuiltUpArea, which is a subclass of a SpatialObject
  - Example: kwg-ont:Hospital is a subclass of geosparql:Feature which is a subclass of SpatialObject

KnowWhereGraph Spatial Types

Is defined by
http://www.opengis.net/ont/geosparql#

has super-classes
Spatial Object Spatial Object [C]

has sub-classes
Administrative Region [C], Bluesky Modeled Wildfire [C], Disaster from FEMA [C], Drought Zone [C], Earthquake [C], Federal Judicial District [C], Hospital [C], MTBS Fire [C], Metropolitan and micropolitan statistical areas [C], NIFC Fire [C]
NOAA Hazard [C], National Weather Zone [C], Nielsen Market Zone [C], Road Segment [C], Road Segment Node [C], Smoke Plume Snapshot [C], Soil Map Unit [C], Storm Track [C], Storm Tracklet [C], US Climate Division [C], Zip Code
Area [C]

IRI: http://www.opengis.net/ont/geosparql#SpatialObject

Anything spatial (having or being a shape, position or an extent).

Is defined by
http://www.opengis.net/ont/geosparql#

has sub-classes
Built Up Area [C], Cell [C], Feature Feature [C], Geometry [C], Geometry Collection [C], Surface Water [C], Terrain [C]

is in domain of
contains [OP], crosses [OP], disjoint [OP], equals [OP], intersects [OP], overlaps [OP], touches [OP], within [OP]

is in range of
contains [OP], crosses [OP], disjoint [OP], equals [OP], intersects [OP], overlaps [OP], touches [OP], within [OP]

KnowWhereGraph Spatial Types

- Important Note!
  - The classes in the previous slides *probably* have subclasses
  - Use the subclasses for getting specific data out
  - With hierarchy, you can create general queries
    - “Give me all the nodes of type kwg-ont:Fire”
    - Powerful, but also difficult for deep dives because different Fire subclasses are described with different predicates

KnowWhereGraph Geometries

- Geometries are described with the geosparql ontology
- geosparql:Features have geometries attached to them via relations
  - geosparql:hasDefaultGeometry
  - geosparql:hasGeometry
- *You can query all geometries the same way* (for geosparql:Feature)
- Geometries are stored in the Well Known Text (WKT)
  - Most geospatial libraries support this format

KnowWhereGraph: Get hospitals within Ohio

- Step 1: Locate Ohio
  - We know that we're looking for a *State*. Which means an AdministrativeRegion_2
  - We know the name (which is hopefully the rdfs:label - so use regex)

```
PREFIX kwgr: <http://stko-kwg.geog.ucsb.edu/lod/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX kwg-ont: <http://stko-kwg.geog.ucsb.edu/lod/ontology/>
select * where {
    ?ohio a kwg-ont:AdministrativeRegion_2 .
    ?ohio rdfs:label ?state_name .
    FILTER (REGEX(?state_name, "Ohio"))
} limit 100
```

Run

keyboard shortcuts

Table | Raw Response | Pivot Table | Google Chart

Download as

Filter query results

Showing results from 1 to 1 of 1. Query took 0.1s, moments ago.

	ohio	state_name
1	kwgr:Earth.North_America.United_States.USA.36_1	"Ohio"

KnowWhereGraph: Get hospitals within Ohio

- Step 2: Ride the spatial backbone
  - Use RCC logic to find the hospitals *within* the ohio node

```
PREFIX kwgr: <http://stko-kwg.geog.ucsb.edu/lod/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX kwg-ont: <http://stko-kwg.geog.ucsb.edu/lod/ontology/>
select ?hospital ?hospital_name where {
    ?ohio a kwg-ont:AdministrativeRegion_2 .
    ?ohio rdfs:label ?state_name .
    FILTER (REGEX(?state_name, "Ohio"))
    ?hospital kwg-ont:sfWithin ?ohio .
    ?hospital a kwg-ont:Hospital .
    ?hospital rdfs:label ?hospital_name .
} limit 100
```

KWG-V2-Vienna | Login

Table | Raw Response | Pivot Table | Google Chart

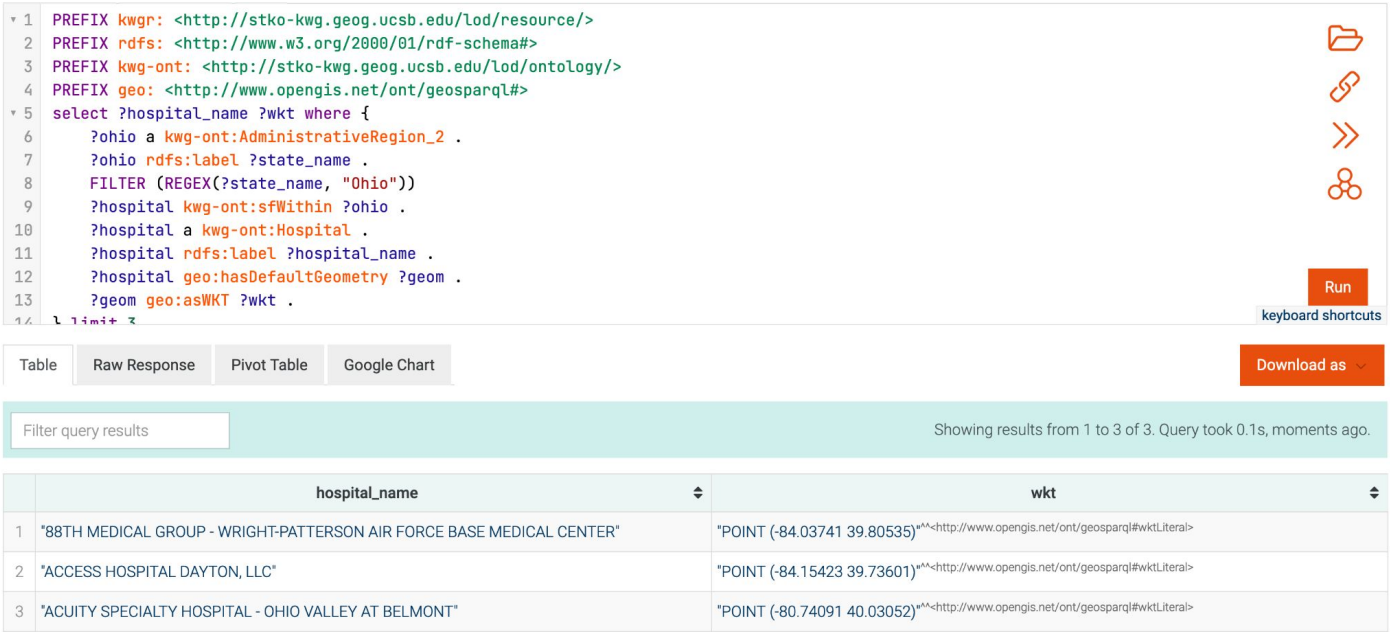
Filter query results

Showing results from 1 to 100 of 100. Query took 0.1s, mo

	hospital	hospital_name
1	kwgr:hospital.88THMEDICALGROUPWRIGHTPATTERSONAIRFORCEBASEMEDICALCENTER	"88TH MEDICAL GROUP - WRIGHT-PATTERSON AIR FORCE BASE MEDICAL CENTER"
2	kwgr:hospital.ACCESSHOSPITALDAYTONLLC	"ACCESS HOSPITAL DAYTON, LLC"
3	kwgr:hospital.ACUITYSPECIALTYHOSPITALOHIOVALLEYATBELMONT	"ACUITY SPECIALTY HOSPITAL - OHIO VALLEY AT BELMONT"
4	kwgr:hospital.ACUITYSPECIALTYOHIOVALLEY	"ACUITY SPECIALTY OHIO VALLEY"
5	kwgr:hospital.ACUTECARESPECIALTYHOSPITALAULTMAN	"ACUTE CARE SPECIALTY HOSPITAL - AULTMAN"
6	kwgr:hospital.ADAMSCOUNTYREGIONALMEDICALCENTER	"ADAMS COUNTY REGIONAL MEDICAL CENTER"
7	kwgr:hospital.ADENAGREENFIELDMEDICALCENTER	"ADENA GREENFIELD MEDICAL CENTER"
8	kwgr:hospital.ADENAREGIONALMEDICALCENTER	"ADENA REGIONAL MEDICAL CENTER"
9	kwgr:hospital.ADVANCEDSPECIALTYHOSPITALOFTOLEDO	"ADVANCED SPECIALTY HOSPITAL OF TOLEDO"

KnowWhereGraph: Get hospitals within Ohio

- Get the geometries (limited to 3 results for readability)



```
PREFIX kwgr: <http://stko-kwg.geog.ucsb.edu/lod/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX kwg-ont: <http://stko-kwg.geog.ucsb.edu/lod/ontology/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
select ?hospital_name ?wkt where {
    ?ohio a kwg-ont:AdministrativeRegion_2 .
    ?ohio rdfs:label ?state_name .
    FILTER (REGEX(?state_name, "Ohio"))
    ?hospital kwg-ont:sfWithin ?ohio .
    ?hospital a kwg-ont:Hospital .
    ?hospital rdfs:label ?hospital_name .
    ?hospital geo:hasDefaultGeometry ?geom .
    ?geom geo:asWKT ?wkt .
} limit 3
```

Run

keyboard shortcuts

Table | Raw Response | Pivot Table | Google Chart

Download as

Filter query results

Showing results from 1 to 3 of 3. Query took 0.1s, moments ago.

	hospital_name	wkt
1	"88TH MEDICAL GROUP - WRIGHT-PATTERSON AIR FORCE BASE MEDICAL CENTER"	"POINT (-84.03741 39.80535)"^^<http://www.opengis.net/ont/geosparql#wktLiteral>
2	"ACCESS HOSPITAL DAYTON, LLC"	"POINT (-84.15423 39.73601)"^^<http://www.opengis.net/ont/geosparql#wktLiteral>
3	"ACUITY SPECIALTY HOSPITAL - OHIO VALLEY AT BELMONT"	"POINT (-80.74091 40.03052)"^^<http://www.opengis.net/ont/geosparql#wktLiteral>

Homework

- Write a python script to download the geometry of all hospitals in Ohio
- Save them in a text file (csv or similar)

Part 2:
Geospatial Analysis

With QGIS

QGIS Overview

- Cool, free software for doing spatial *things*
  - This may be planning
  - Drawing shapes on a map
  - Advanced functionality comes from *Plugins*
    - Network analysis
    - Spatial analysis
    - Temporal integrations
    - Street view
    - Support for more data formats
    - Etc
  - Two ways of working
    - Click this menu, then click this button, then this other button
    - Python

QGIS: Loading Hospital Geometries

- Big Picture
  - Mortality rate increases ~ %1 per 10 km away from a hospital
  - Load data into QGIS
    - Hospital points
    - Center of counties (point geometry)
  - Connect each county to the closest hospital
- QGIS supports loading WKT
  - -> Add Delimited Text Layer (choose wkt option)
  - Google how to if you forget

QGIS: Loading Hospital Geometries

Get county geometries

```
PREFIX kwg-ont: <http://stko-kwg.geog.ucsb.edu/lod/ontology/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select ?wkt where {
    BIND(<http://stko-kwg.geog.ucsb.edu/lod/resource/Earth.North_America.United_States.USA.36_1> as ?ohio)
    ?town a kwg-ont:AdministrativeRegion_3 .
    ?town kwg-ont:sfWithin ?ohio .
    ?town rdfs:label ?name .
    ?town geo:hasGeometry ?geo .
    ?geo geo:asWKT ?wkt .
} limit 10000
```

QGIS: Loading Hospital Geometries

Get county geometries

```
from shapely import wkt

wkt_file = open('res.csv', 'r')
wkt_lines = wkt_file.readlines()
for wkt_line in wkt_lines:
    g = wkt.loads(wkt_line)
    print(g.centroid)
```

```
POINT (-84.58601785660353 40.855316223070325)
POINT (-84.16704414724646 39.42745506021709)
POINT (-81.88827771192568 40.829065494505876)
POINT (-84.58866239497604 41.56075345073124)
POINT (-83.6235259790934 41.36179684833092)
POINT (-83.3040092887956 40.84254458847032)
POINT (-84.57550628649356 39.43852087563047)
POINT (-81.31433513976843 41.97363682717935)

Process finished with exit code 0
```

QGIS: Loading Hospital Geometries

Load into QGIS: Live demo here