实习2

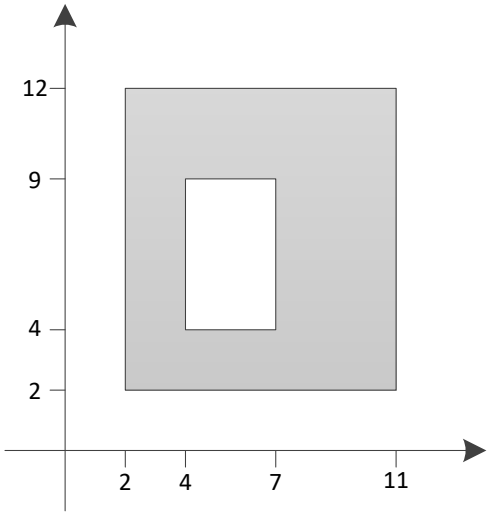
空间数据库创建和查询

实习2

- PostgreSQL，PostGIS，QGIS，OpenStreetMap
- 关键：学会查PostGIS帮助文档
  - 现在学的函数，可能等你毕业时已经全部更新了
  - ST_StartPoint等函数在PostGIS2.0版本以后已经不再支持MultiPolygon
- 主要问题
  - OGC SFA标准的基本概念
  - PostGIS空间查询函数
    - LineString vs. MultiLineString, Polygon vs. MultiPolygon
    - ST_Touches, ST_Crosses, ST_Overlaps, ST_Intersects
  - 空间计算核心概念：空间关联和时空查询
    - 空间距离、空间包含关联道路与交通事故等
    - 空间距离、拓扑关系判断等较为耗时，需要关注查询效率
    - 基于实际应用场景的时空数据分析(交通事故的星期与天数统计)

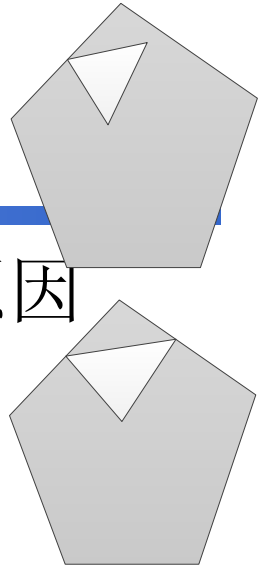
OGC SFA标准

- 1. 多边形的WKT表示
  - Polygon((2 2,11 2,11 12,2 12), (4 4,4 9,7 9,7 4))
  - Polygon((2 2,11 2,11 12,2 12,2 2), (4 4,4 9,7 9,7 4,4 4))
  - 课件Lecture 4：P104 Geometry表；P118 WKT举例
  - 课堂练习3的各类几何数据插入语句



http://portal.opengeospatial.org/files/?artifact_id=25355

OGC SFA标准

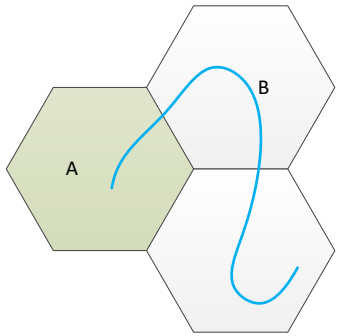


- 2. 基于6.1.11.1的assertions，给出无法表示原因
  - a) Polygons are topologically closed;
  - b) The boundary of a Polygon consists of a set of LinearRings that make up its exterior and interior boundaries;
  - c) No two Rings in the boundary cross and the Rings in the boundary of a Polygon may intersect at a Point but only as a tangent;
  - d) A Polygon may not have cut lines, spikes or punctures;
  - e) The interior of every Polygon is a connected point set;
  - f) The exterior of a Polygon with 1 or more holes is not connected. Each hole defines a connected component of the exterior.

OGC SFA标准

- 3.绿色多边形(A)和蓝色线(B)的Dimensionally Extended Nine-Intersection Model (DE-9IM)
  - DE-9IM 课件Lecture 4 P62，OGC Part 1 6.1.15.2

	I(B)	B(B)	E(B)
I(A)	1	0	2
B(A)	0	-1	1
E(A)	1	0	2



- 4.当a.Relate(b,“T*T***T**”)返回True时，请给出几何对象a和b所对应的空间关系
  - 9IM 课件Lecture 4 P59-67，OGC Part 1 6.1.15.2
  - 8种空间关系的九交矩阵字符串表示 OGC Part 1 6.1.15.3

空间关系	定义	九交矩阵
Overlaps	Dim(I(a)) = Dim(I(b)) = Dim(I(a) ∩ $I(b)$)，$a \cap b \neq a$，$a \cap b \neq b$	T*T***T** (点/点, 面/面) 1*T***T** (线/线)

OGC SFA标准

- 5.请给出空间关系Contains的九交矩阵(9IM)的字符串表示。
  - OGC Part 1 6.1.15.3
  - Contains：T*****FF*

空间关系	定义	九交矩阵
Within	$a \cap b = a$，$\mathrm{I}(a) \cap E(b) = \emptyset$	T*F**F***
Contains	a.Contains(b) ←→ b.Within(a)	?

PostGIS函数

- 2. ST_Distance函数说明： http://postgis.net/docs/ST_Distance.html
  - For geometry type Returns the 2D Cartesian distance between two geometries in projected units (based on spatial ref).
  - For geography type defaults to return minimum geodesic distance between two geographies in meters.
- 在空间参考系4326下，使用ST_Distance(geometry(Point, 4326), geometry(LineString, 4326))计算距离，返回什么距离和单位？
  - Geometry: 2D Cartesian distance, degrees(4326)
  - Geography: minimum geodesic distance, meters

```
--Geometry example - units in planar degrees 4326 is WGS 84 long lat unit=degrees
SELECT ST_Distance(
                ST_GeomFromText('POINT(-72.1235 42.3521)',4326),
                ST_GeomFromText('LINESTRING(-72.1260 42.45, -72.123 42.1546)', 4326)
        );
st_distance
------------------
0.00150567726382282
```

http://postgis.net/docs/reference.html

PostGIS函数

- 3.基于帮助文档，请回答=(等于号操作符)、ST_Equals和ST_OrderingEquals三个函数的异同
  - = http://postgis.net/docs/manual-2.2/ST_Geometry_EQ.html PostGIS 2.2
    - Returns TRUE if A's bounding box is the same as B's. Uses double precision bounding box.
  - = http://postgis.net/docs/ST_Geometry_EQ.html PostGIS 2.4 ~=
    - Returns TRUE if the coordinates and coordinate order geometry/geography A are the same as the coordinates and coordinate order of geometry/geography B.
  - ST_Equals http://postgis.net/docs/ST_Equals.html
    - Returns true if the given geometries represent the same geometry. Directionality is ignored.
  - ST_OrderingEquals http://postgis.net/docs/ST_OrderingEquals.html
    - Returns true if the given geometries represent the same geometry and points are in the same directional order.

PostGIS函数

- 3.基于帮助文档，请回答=(等于号操作符)、ST_Equals和ST_OrderingEquals三个函数的异同
  - PostGIS 2.2
    - A. LineString(0 0, 10 10)
    - B. Polygon((0 0, 10 0, 10 10, 0 10, 0 0))
    - C. LineString(0 0, 5 5, 10 10)
    - D. LineString(10 10, 0 0)

PostGIS函数

- 4. ST_Distance(Point, Polygon) <= 10和ST_DWithin(Point, Polygon, 10)功能上等价，而效率差异较大。基于帮助文档，分析效率差异的原因
  - ST_DWithin http://postgis.net/docs/ST_DWithin.html

```
--Find the nearest hospital to each school
--that is within 3000 units of the school.
-- We do an ST_DWithin search to utilize indexes to limit our search list
-- that the non-indexable ST_Distance needs to process
--If the units of the spatial reference is meters then units would be meters
SELECT DISTINCT ON (s.gid) s.gid, s.school_name, s.the_geom, h.hospital_name
        FROM schools s
                LEFT JOIN hospitals h ON ST_DWithin(s.the_geom, h.the_geom, 3000)
        ORDER BY s.gid, ST_Distance(s.the_geom, h.the_geom);

--The schools with no close hospitals
--Find all schools with no hospital within 3000 units
--away from the school.  Units is in units of spatial ref (e.g. meters, feet, degrees)
SELECT s.gid, s.school_name
        FROM schools s
                LEFT JOIN hospitals h ON ST_DWithin(s.the_geom, h.the_geom, 3000)
        WHERE h.gid IS NULL;
```

PostGIS函数

PostGIS 2.2 ST_Distance_Sphere

PostGIS 2.4 ST_DistanceSphere

- 5.基于帮助文档，请比较以下三个函数的异同。
  - ST_DistanceSphere http://postgis.net/docs/ST_DistanceSphere.html
    - Returns minimum distance in meters between two lon/lat geometries. Uses a spherical earth and radius derived from the spheroid defined by the SRID. Faster than ST_DistanceSpheroid, but less accurate.
  - ST_Distance http://postgis.net/docs/ST_Distance.html
    - For geometry type Returns the 2D Cartesian distance between two geometries in projected units (based on spatial ref). For geography type defaults to return minimum geodesic distance between two geographies in meters.
  - ST_DistanceSpheroid http://postgis.net/docs/ST_Distance_Spheroid.html
    - Returns the minimum distance between two lon/lat geometries given a particular spheroid.

PostGIS函数

- 6. 哪个函数可以将MultiXXX转换XXX，如MultiPolygon转换获得多个Polygon？
  - ST_Dump
    - geometry_dump[] ST_Dump(geometry g1);
    - Returns a set of geometry_dump (geom,path) rows, that make up a geometry g1.
    - http://postgis.net/docs/ST_Dump.html

Shapefile数据导入

- 通过PostgreSQL命令导入
  - shp2pgsql us_lakes.shp us_lakes > uslakes.sql
  - shp2pgsql usa_major_highways.shp usa_major_highways > ushighways.sql
  - 导入数据库
  - psql -U postgres -d usmap -f uslakes.sql
  - psql -U postgres -d usmap -f ushighways.sql
- 通过PostGIS 2.0 Shapefile and DBF Loader Exporter
- 通过QGIS连接数据库，导入shapefile文件
  - QGIS自带的SPIT工具无法对MultiLineString类型的ushighways数据进行导入

Shapefile数据导入

- Uslake.sql

```
CREATE TABLE "lakes" (gid serial PRIMARY KEY,
"area" numeric,
"name" varchar(25),
"shape_leng" numeric,
"shape_area" numeric);
SELECT AddGeometryColumn('', 'lakes', 'geom', '-1', 'MULTIPOLYGON', 2);
INSERT INTO "lakes" ("area","name","shape_leng","shape_area",geom) VALUES ('2.35944100000e+003','Great Salt Lake','4.97206617981e+000','6.55966373057e-001','010600000001…')
```

Highway和lake表

- Highway
  - Gid
  - Name
  - Full_name
  - The_gemo
- Lake
  - Gid
  - Name
  - Full_name
  - Shape_length
  - Shape_area
  - The_geom
- 单位是米，还是千米？

City表

- create table uscities (
  gid integer primary key,
  name varchar(100),
  state varchar(100),
  latitude numeric,
  longitude numeric);
- copy uscities from ‘…\uscity.txt’ delimiter '#';
  - 为什么不是’,’?

City表

- select ST_SRID(uslakes.geom, 4326) from uslakes;
- select AddGeometryColumn('public', 'uscities', 'geom', 0, 'POINT', 2);
- update uscities

  set geom = ST_Point(longitude, latitude);

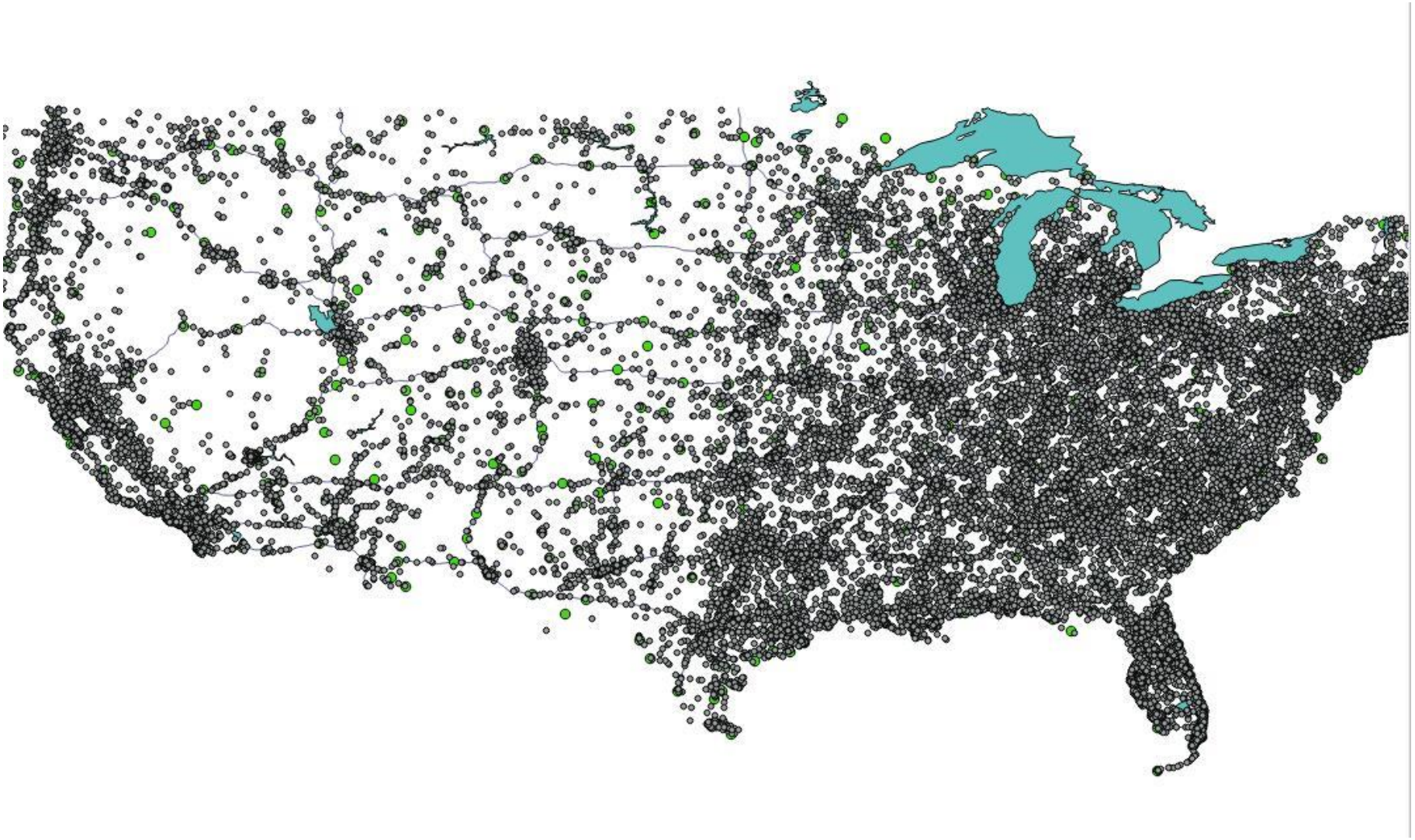
- 如何创建表时，直接创建几何属性？

Accidents表

- Accidents
  - 美国高速公路交通事故数据来源于美国交通局
  - STATE为美国56个州的ID
  - ST_CASE由州ID和交通事故编号组成
    - 同一天可能有多起交通事故，所以不能用ST_CASE统计天数
  - 交通事故发生在county和city
  - 时间为day, month, year, day_week, hour和minute
    - 2015.1.1号对应是星期四，day_week = 5
    - hour = 99?
  - 地点在latitude和longitud，geometry(Point, 4326)
  - 是否酒驾drunk_dr，大于0为酒驾
  - https://www.transportation.gov/fastlane/2015-traffic-fatalities-data-has-just-been-released-call-action-download-and-analyze

QGIS数据展示

- 练习GIS软件如何与数据库进行连接和图层展示



GIS分析与SQL查询

- 1. 查询伊利湖(Erie)几何中点的数目

```
select ST_NPoints(geom)
from uslakes
where name = 'Lake Erie';
```

ST_NumXXX vs. ST_NXXX区别

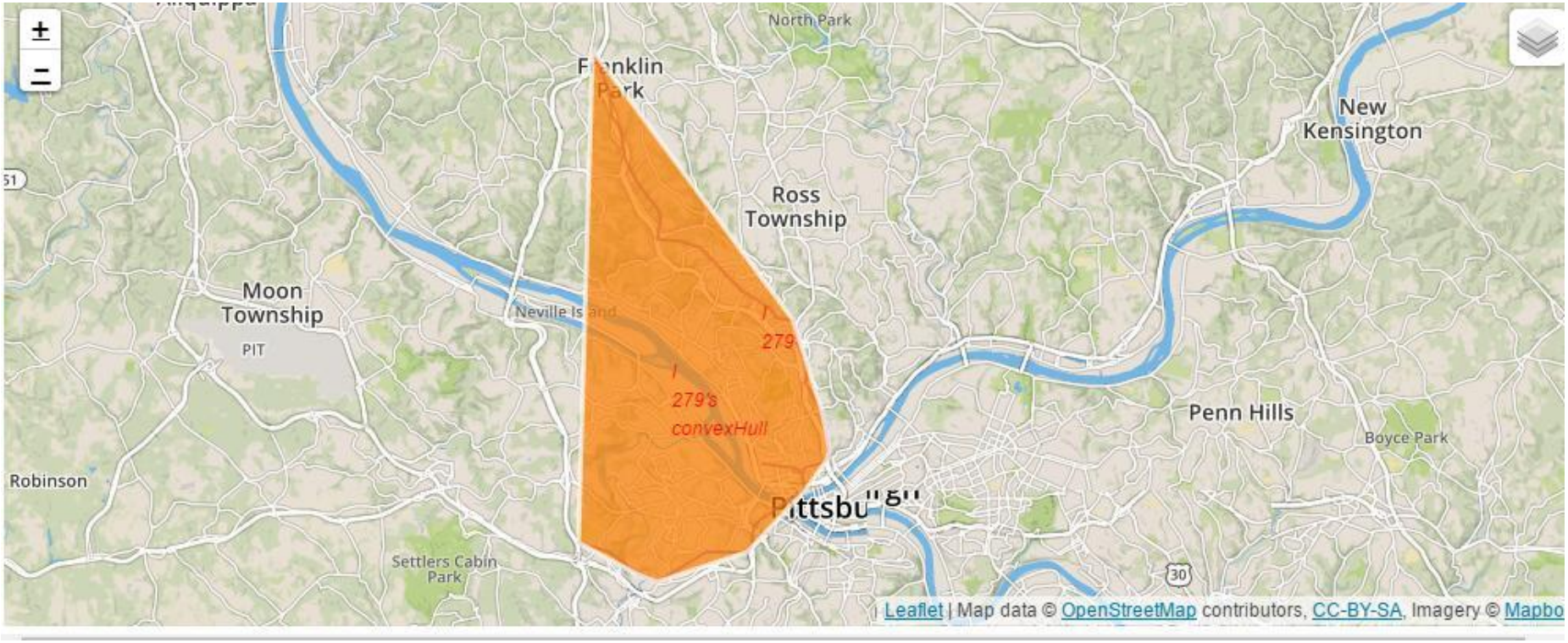
- ST_NumPoints
  - Return the number of points in an ST_LineString or ST_CircularString value
- ST_NPoints
  - Return the number of points (vertexes) in a geometry

http://postgis.net/docs/reference.html

GIS分析与SQL查询

- 2. 查询高速公路全称(full_name)为’I 279’的凸包

```
select gid, name, ST_ConvexHull(geom) as geom
from ushighways
where full_name = ‘I 279';
```



GIS分析与SQL查询

- 3. 查询哪些湖中有岛

```
select gid, name, geom
from uslakes
where ST_NRings(geom) > 1;

select gid, name, geom
from uslakes
where ST_NumInteriorRings(geom) > 0;
```

思考：如何修改上述语句？

GIS分析与SQL查询

- 3. 查询哪些湖中有岛

GIS分析与SQL查询

- 3. 查询哪些湖中有岛

GIS分析与SQL查询

- ST_NRings
  - If the geometry is a polygon or multi-polygon returns the number of rings.
- ST_NumInteriorRings
  - Return the number of interior rings of a polygon geometry
  - This method implements the SQL/MM specification. SQL-MM 3: 8.2.5
  - Changed: 2.0.0 - in prior versions it would allow passing a MULTIPOLYGON, returning the number of interior rings of first POLYGON.

GIS分析与SQL查询

- ST_NRings
- ST_NumInteriorRings
- SELECT ST_NRings(the_geom) As Nrings, ST_NumInteriorRings(the_geom) As ninterrings FROM (SELECT ST_GeomFromText('POLYGON((1 2, 3 4, 5 6, 1 2))') As the_geom) As foo;
- nrings | ninterrings
- --------+------------
- 1 | 0
- (1 row)

GIS分析与SQL查询

- 3. 查询哪些湖中有岛

```
select gid, geom, name
from lakes
where (ST_NRings(geom) - ST_NumGeometries(geom)) <> 0;

select gid, name, geom
from (select gid, name, (ST_Dump(geom)).geom as geom
        from lake ) as foo
group by gid, name, geom
having sum(ST_NumInteriorRings(geom)) <> 0;
```

GIS分析与SQL查询

- 4. 查询湖的面积属性是否准确(绝对误差小于1e-6)，列出面积属性不准确的湖及其误差
  - 相对误差 vs. 绝对误差
  - 面积属性不准确的湖及其误差

GIS分析与SQL查询

- 5. 查询最短的高速公路及其长度（单位为千米）
- 方法一

```
select gid, full_name, ST_Length(A.geom)
from ushighways as A
where ST_Length(A.geom) <= all(select
ST_Length(B.geom) from ushighways as B);
```

- N条高速公路，共计算了多少次长度？
  - 几何函数计算时间较长，仅可能减少几何函数的调用次数

GIS分析与SQL查询

- 5. 查询最短的高速公路及其长度（单位为千米）
- 方法二

```
select full_name, ST_Length(geom)
from ushighways
where ST_Length(geom) =
(select min(ST_Length(geom)) from ushighways);
```

- N条高速公路，共计算了多少次长度？
  - 几何函数计算时间较长，仅可能减少几何函数的调用次数

GIS分析与SQL查询

- 5. 查询最短的高速公路及其长度（单位为千米）
- 方法三

```
select gid, full_name, geom, len
from ushighways A,
        (select gid, ST_Length(geom) as len
         from ushighways) as B
where A.gid = B.gid and
          B.len = (Select min(len) from B);
```

GIS分析与SQL查询

- 5. 查询最短的高速公路及其长度（单位为千米）
- 方法三

```
with B as (select gid, ST_Length(geom) as len
                from ushighways)

select gid, full_name, geom, len
from ushighways, B
where A.gid = B.gid and
        B.len = (Select min(len) from B);
```

- N条高速公路，共计算了多少次长度？
  - 几何函数计算时间较长，仅可能减少几何函数的调用次数

GIS分析与SQL查询

- 5. 查询最短的高速公路及其长度（单位为千米）
- 方法三

```
select gid, full_name, geom, len
from ushighways A,
        (select min(ST_Length(geom)) as len
         from ushighways) as B
where ST_Length(geom) = len
```

- N条高速公路，共计算了多少次长度？
  - 几何函数计算时间较长，仅可能减少几何函数的调用次数

GIS分析与SQL查询

- 5. 查询最短的高速公路及其长度（单位为千米）
- 方法四

```
select gid, full_name, geom, L.len
from ushighways A,
        (select gid, ST_Length(geom) as len
         from ushighways) as L
where A.gid = B.gid
order by L.length desc
limit 1
```

GIS分析与SQL查询

- 5. 查询最短的高速公路及其长度（单位为千米）
  - ST_Length(geom)
    - 4326对应的是经纬度，所以距离单位也是度
    - Geometry example - units in planar degrees 4326 is WGS 84 long lat unit=degrees
  - ST_Length(ST_Transform(geom, 26986))
    - 空间参考系26969
    - units in meters (SRID: 26986 Massachusetts state plane meters) (most accurate for Massachusetts)
    - units in meters (SRID: 2163 US National Atlas Equal area) (leas t accurate)
  - ST_Length(geom :: geography)
    - 将geometry类型转成geography，距离单位也是米，same but note units in meters - use sphere for slightly faster less accurate
  - ST_Length(geom, true/false) 几何类型自动转换(C++)

GIS分析与SQL查询

- 6. 查询与苏必利尔湖的质心距离最近的城市
  - ST_Distance
  - ST_Centroid
- 7.查询距离ST_CASE = 10001交通事故最近的城市
  - ST_Distance
  - name || ‘in’ || state

GIS分析与SQL查询

- 8. 查询94号公路与哪些高速公路联通，不包括94号公路，并求取总长度

```
SELECT H.id, H.full_name, H.geom
FROM ushighways H94, ushighways H
WHERE H94.full_name like '%94%' and H.gid <> H94.gid and
        (ST_Crosses(H94.geom, H.geom) or
         ST_Touches(H94.geom, H.geom))
```

H94.gid = 94

H94.name = '94'

GIS分析与SQL查询

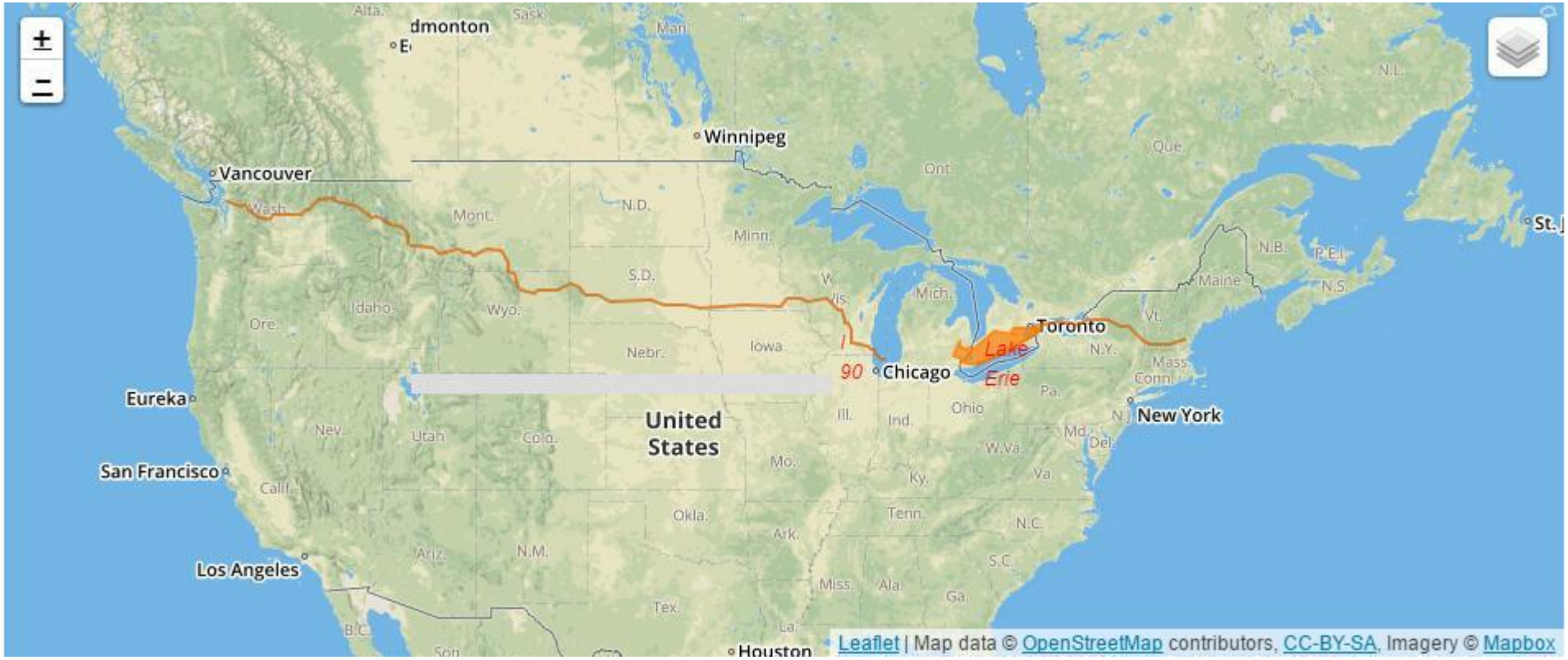
- 8. 查询94号公路与哪些高速公路联通，不包括94号公路，并求取总长度
- T_Intersects / ST_Disjoint
  - 如果相交没有生成空集，则 ST_Intersects 返回 1 或 t (TRUE)
- ST_Crosses
  - 如果交集形成了一个维度比两个源几何的最大维度小一的几何，并且交集位于两个源几何内部，则 ST_Crosses 返回 1 或 t (TRUE)
- ST_Touches
  - 如果两个几何的公共点都不与两个几何的内部相交，则 ST_Touches 返回 1 或 t (TRUE)
  - 线的边界和内部是什么？

GIS分析与SQL查询

- 8. 查询94号公路与哪些高速公路联通，不包括94号公路，并求取总长度
- ST_Intersects / ST_Disjoint
- ST_Crosses
- ST_Touches
- ST_Geometry 的空间关系函数
  - http://help.arcgis.com/zh-cn/ARCGISDESKTOP/10.0/HELP../index.html#//006z000000001z000000

GIS分析与SQL查询

- 9. 查询与伊利湖(Erie)距离最近的高速公路
  - 最小值问题
  - ST_Distance



GIS分析与SQL查询

- 10. 查询哪个城市最偏僻

```
select C.gid, C.name, C.geom

from uscities C, ushighways H,
(select min(ST_Distance(A.geom, B.geom)) as mindis
from ushighways A, uscities B
group by B.gid) D

where ST_Distance(C.geom, H.geom) = (select
max(mindis) from D)
```

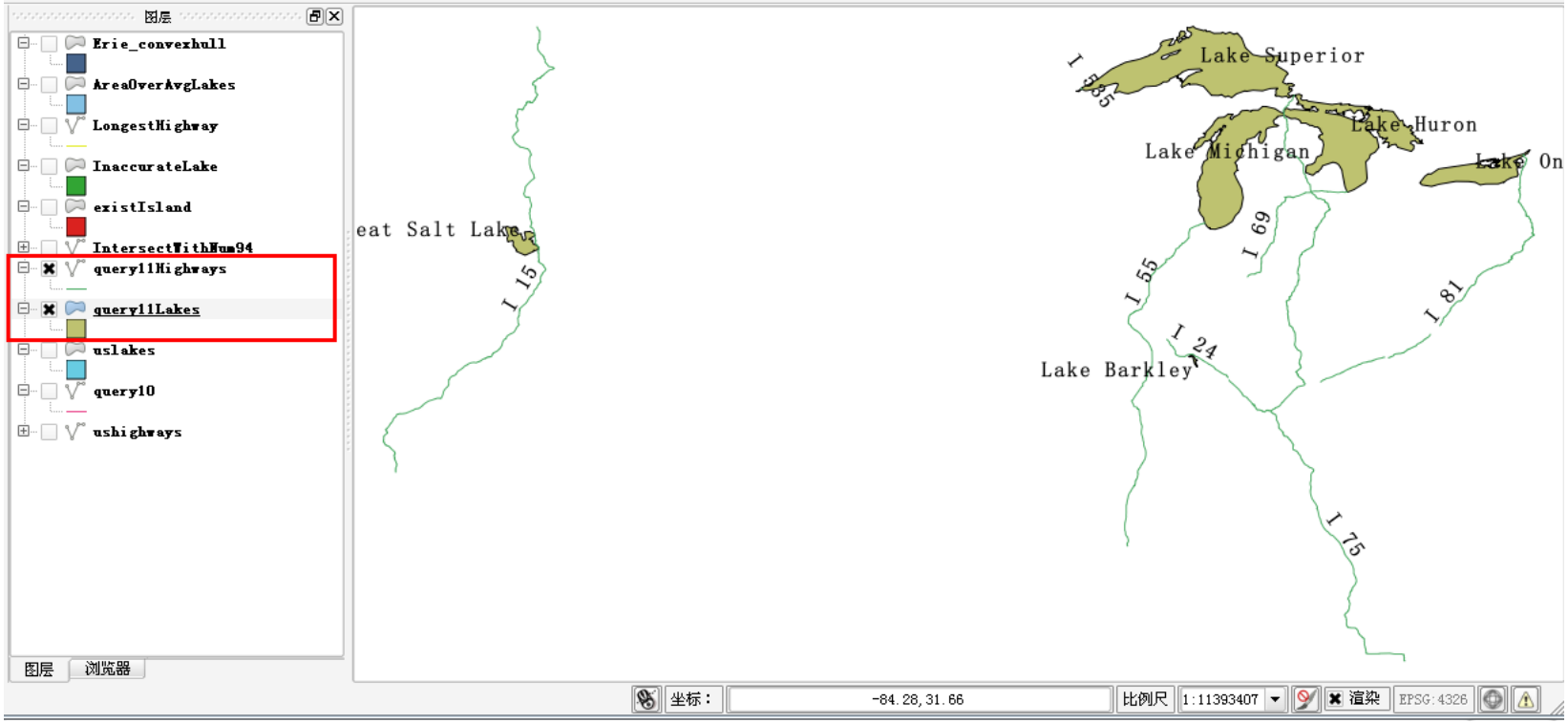
GIS分析与SQL查询

- 11. 查询哪些高速公路穿越湖，列出高速公路及其在湖中的长度，按长度从长到短排列

```
select H.full_name, L.name,
ST_Length(ST_Intersection(H.geom, L.geom), true)
as len
from ushighways H, uslakes L
where ST_Crosses(H.geom, L.geom)
order by len desc
```

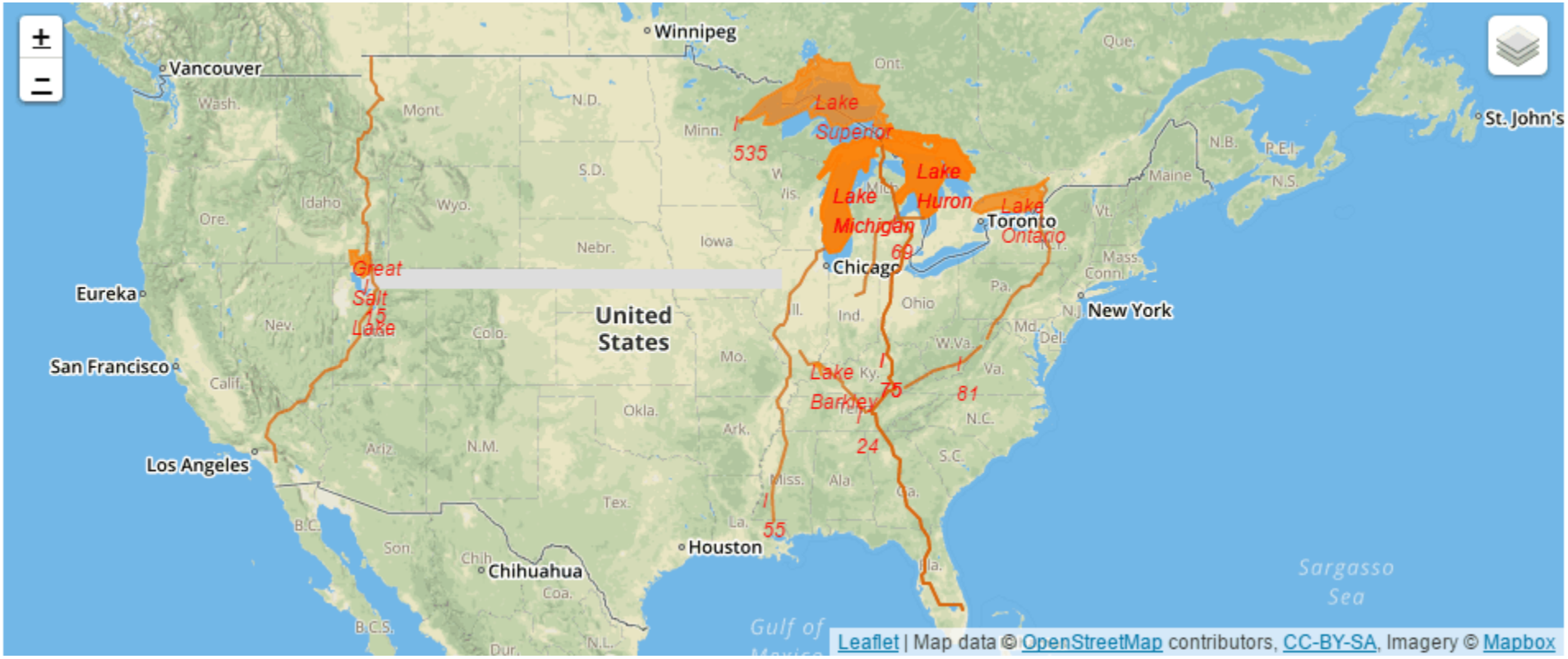
GIS分析与SQL查询

- 11. 查询哪些高速公路穿越湖，列出高速公路及其在湖中的长度，按长度从长到短排列
  - QGIS中直接显示可能会导致gid相同，即同一条路穿越多个湖，可以把路id+湖id作为gid进行显示



GIS分析与SQL查询

- 11. 查询哪些高速公路穿越湖，列出高速公路及其在湖中的长度，按长度从长到短排列



GIS分析与SQL查询

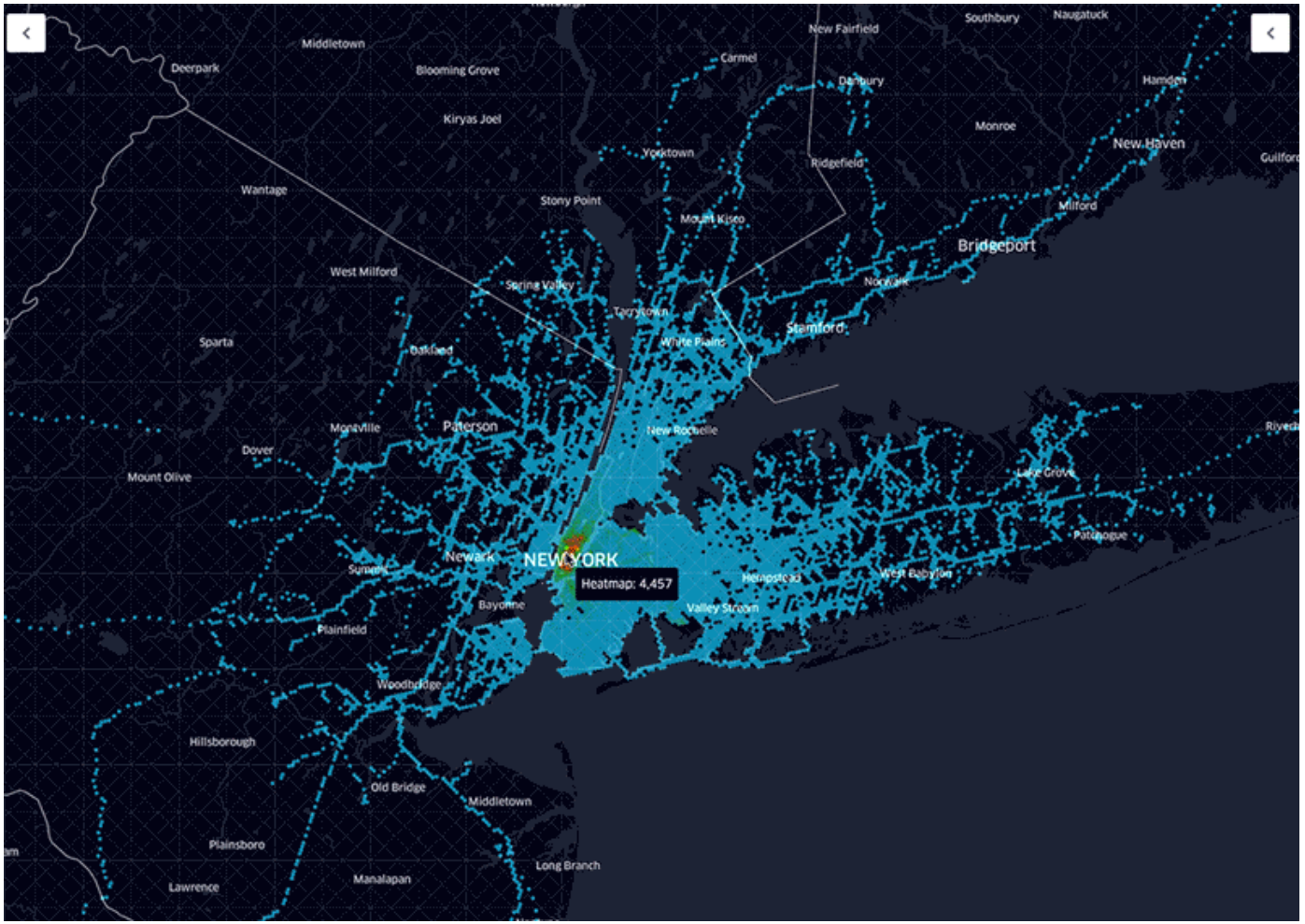
- 12. 将交通事故与高速公路基于空间距离进行关联，查询哪条高速公路上的交通事故最多

```
select H.gid, H.full_name, H.geom, count(*)
from ushighways H, usaccidents A
where ST_DWithin(A.geom, H.geom, 500, true) and month between 7 and 8
group by H.gid, H.full_name, H.geom
having count(*) >= (….)
```

- N条高速公路和M个城市，共计算了多少次距离?
- 用with和limit减少距离计算次数
- 类似问题
  - 哪条道路上加油站最多？哪条道路上车辆最多？哪个湖中船最多？

GIS分析与SQL查询

- 13. 加州交通事故方格统计
  - Uber/滴滴运力分析 https://eng.uber.com/data-viz-intel/



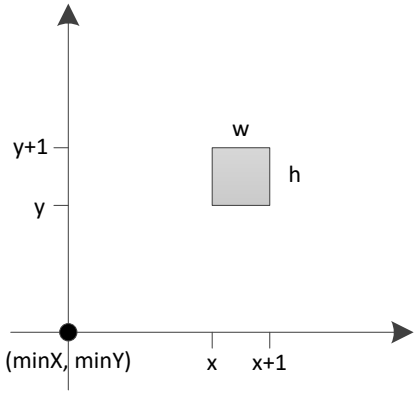
GIS分析与SQL查询

- 13. 加州交通事故方格统计
  - 生成方格，与几何求交，统计交通事故，可视化

```
with calBoundary as (
select ST_XMin(geom) minX, ST_YMin(geom) minY,
        (ST_XMax(geom) – ST_XMin(geom)) / 50.0 w,
       (ST_YMax(geom) – ST_YMin(geom)) / 46.0 h
from cal),
```

GIS分析与SQL查询

- 13. 加州交通事故方格统计
  - 生成方格，与几何求交，统计交通事故，可视化




```
with calBoundary as (
select minX, minY, w, h from cal),
grid as (select y * 50 + x gid,
            ST_MakeEnvelope(
            minX + (x - 1) * w, minY + (y - 1) * h,
            minX + x * w, minY + y * h, 4326) geom
          from generate_series(1, 50) X(x),
               generate_series(1, 46) Y(y),
               calBoundary),
```

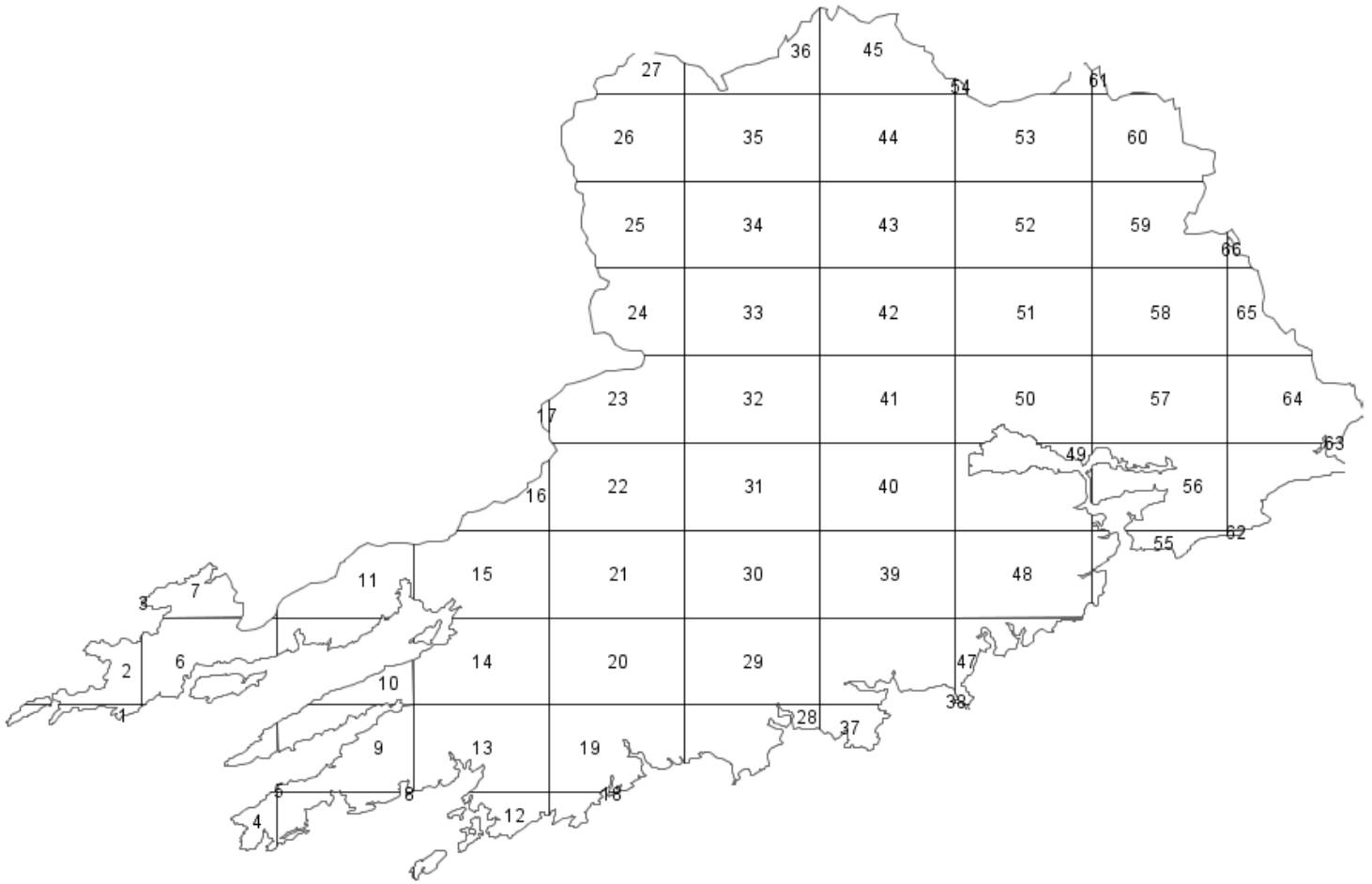
GIS分析与SQL查询

- 13. 加州交通事故方格统计
  - 生成方格，与几何求交，统计交通事故，可视化

```
with calBoundary as (
select minX, minY, w, h from cal),
grid as (select gid geom from X(x),  Y(y), calBoundary),
grid_intersection as (select G.gid as gid,
        ST_Intersection(G.geom, C.geom) as geom
        from grid G, cal C
        where ST_Intersects(G.geom, C.geom)),
```

GIS分析与SQL查询

- 13. 加州交通事故方格统计
  - 生成方格，与几何求交，统计交通事故，可视化



GIS分析与SQL查询

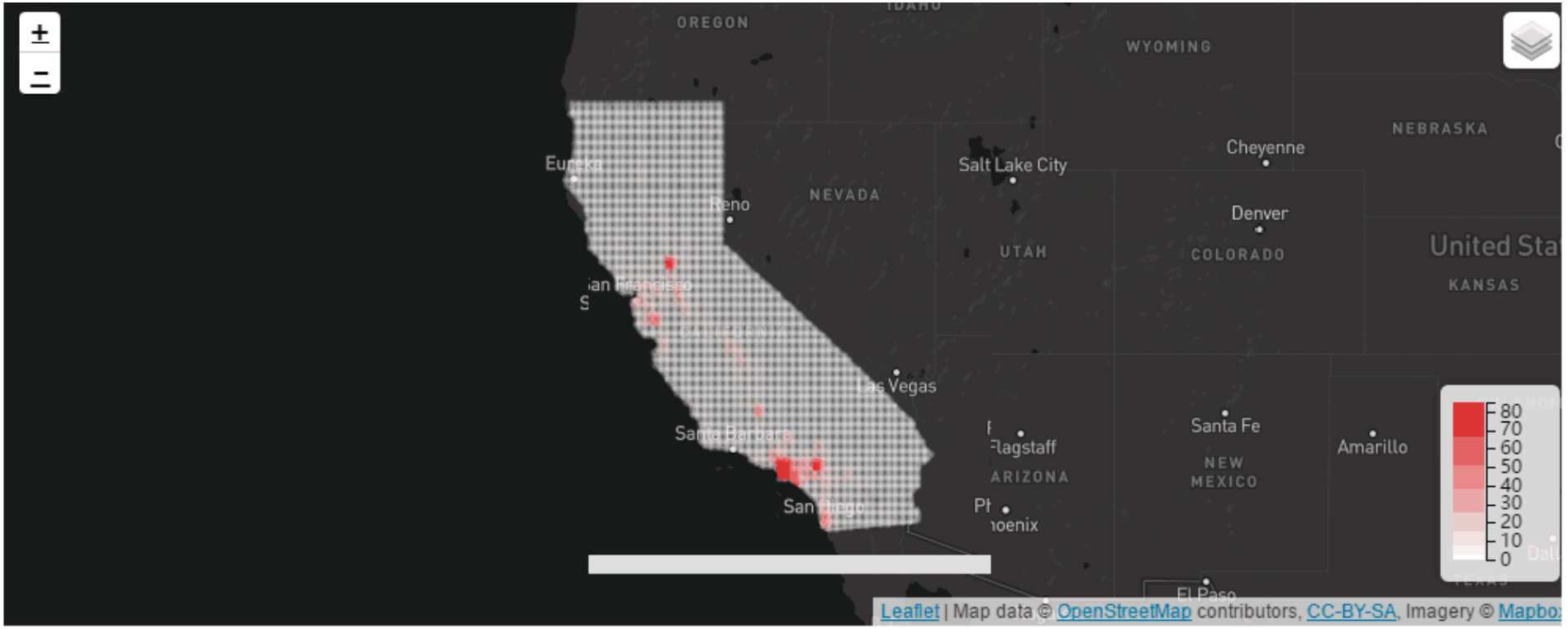
- 13. 加州交通事故方格统计
  - 生成方格，与几何求交，统计交通事故，可视化

```
with calBoundary as (
select minX, minY, w, h from cal),
grid as (select gid geom from X(x),  Y(y), calBoundary),
grid_intersection as (select gid, geom from grid, cal),

select G.gid, G.geom, count(*) value
from grid_intersection G, usaccidents A
where ST_Within(A.geom, G.geom)
group by G.gid, G.geom
```

GIS分析与SQL查询

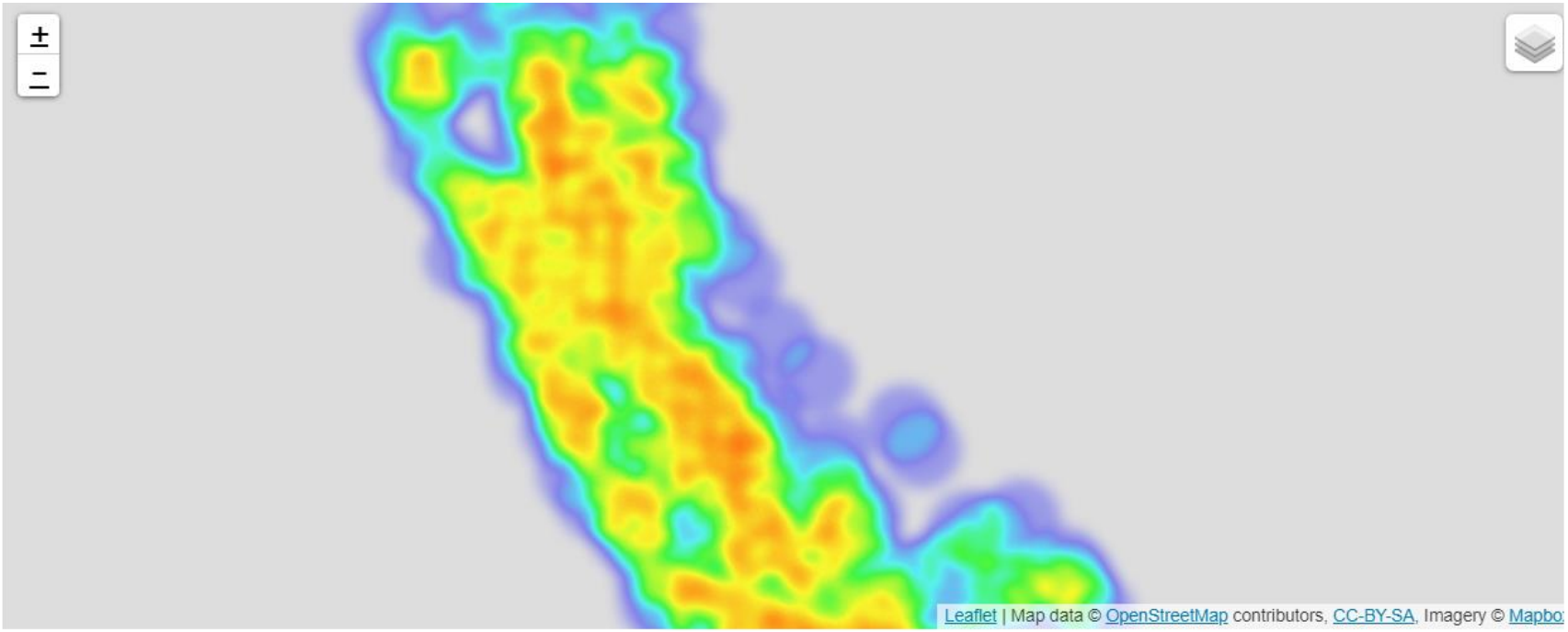
- 13. 加州交通事故方格统计



GIS分析与SQL查询

- 14. 查询在加州范围内的交通事故，通过heatMap进行可视化

```
select A.gid, 'accident' as name, A.geom
from cal, usaccidents A
where ST_Intersects(A.geom, cal.geom)
```



GIS分析与SQL查询

- 15. 酒驾是否在周末时发生的概率较高？
  - day_week对应星期几，天数统计

```
select round(weekday_accident_count * 1.0 / weekday_day_count, 4) as avg_weekday_count,
from
(select count(*) as weekday_accident_count from usaccidents where drunk_dr > 0 and day_week between 2 and 6) as weekday1,
(select count(distinct(month || '-' || day)) as weekday_day_count from usaccidents where day_week between 2 and 6) as weekday2,
```

GIS分析与SQL查询

- 16. 酒驾在工作日和休息日主要发生在哪个时间段？
- 17. 酒驾交通事故次数在每个小时上是否与总的交通事故次数成正比？

hour	avg_weekday_count	avg_weekend_count
0.0	1.2261	3.0865
1.0	1.1916	3.7019
2.0	1.0881	4.2788
3.0	0.7318	3.2115
4.0	0.4406	1.7981
5.0	0.3870	1.2885
6.0	0.3257	1.0481
7.0	0.3295	0.7115
8.0	0.1609	0.3558
9.0	0.1762	0.3462
10.0	0.1686	0.4135
11.0	0.2720	0.4904
12.0	0.3027	0.4231
13.0	0.3908	0.6731
14.0	0.4176	0.7885
15.0	0.5785	1.0288
16.0	0.7165	1.1923
17.0	0.8812	1.9135
18.0	1.1303	1.9519
19.0	1.1801	2.1154
20.0	1.2644	2.4038
21.0	1.4674	2.3173
22.0	1.4483	2.0577
23.0	1.3678	2.2788

hour	drunk_count	total_count	round
0.0	641	1252	0.5120
1.0	696	1200	0.5800
2.0	729	1188	0.6136
3.0	525	936	0.5609
4.0	302	741	0.4076
5.0	235	988	0.2379
6.0	194	1185	0.1637
7.0	160	1131	0.1415
8.0	79	906	0.0872
9.0	82	948	0.0865
10.0	87	1019	0.0854
11.0	122	1117	0.1092
12.0	123	1205	0.1021
13.0	172	1376	0.1250
14.0	191	1489	0.1283
15.0	258	1669	0.1546
16.0	311	1624	0.1915
17.0	429	1826	0.2349
18.0	498	1878	0.2652
19.0	528	1758	0.3003
20.0	580	1836	0.3159
21.0	624	1792	0.3482
22.0	592	1538	0.3849
23.0	594	1315	0.4517

GIS分析与SQL查询

- 18.分析周末与非周末哪些高速公路上发生酒驾概率有较大差异

```
select H.gid as gid, H.full_name as name,
count(*) as value, H.geom as geom
from ushighways H, usaccidents A
where ST_DWithin(A.geom, H.geom, 500, true) and
day_week between 2 and 6 and drunk_dr > 0
group by H.gid, H.full_name, H. geom
```

GIS分析与SQL查询

- 18.分析周末与非周末哪些高速公路上发生酒驾概率有较大差异

'ResultSet' object has no attribute 'has_key'