AVIAN GEOSEARCH

UTILIZING CITIZEN-SCIENCE TO LOCATE BIRD SPECIES IN PROTECTED AREAS

KIRK MCPHAIL AND DORN MOORE – GEOGRAPHY 574, SPRING 2017

SECTION 1: INTRODUCTION & OBJECTIVES

In the age of online collaboration, the scientific research community at large has gained the ability to share their findings with the public readily. Likewise, so has the general population; thus, the concept of citizen science has thrived. One of our primary sources, eBird, is a prime example of an organization that utilizes data from these self-titled citizen scientists. The Cornell Lab of Ornithology and National Audubon Society pooled their resources in 2002, resulting in what is now "…one of the largest and fastest growing biodiversity data resources in existence." (eBird.org).

There is a considerable community of dedicated people who use technology to network with fellow birders seeking opportunities to observe more birds in their home town or favorite vacation spot. However, for novice birders or those with a passing interest, it can be difficult to discover where to see a particular species or find a list of common species seen a particular area. For example, at the International Crane Foundation, where one of the authors works, it is a common question from visitors to our site or website to ask "Where can I see cranes near me?" Our goal with this project was to help bring together the over 274 million eBird species sightings in the United States to help people find birds.

RESEARCH QUESTIONS

From the early planning stages of this project, our primary objective has been to create a platform that answers variations of three questions:

- What bird species can I see nearby?
- Where can I see a certain species nearby?
- When can I see a certain species nearby?

The first question demonstrates the minimum requirement when building a proper query statement for use in our database; a user must designate a search location. Building on this stipulation, the second and third questions allow for bird species as an additional parameter. More detailed queries may include specifying a certain month, natural area, or any combination of the variables above. This project focuses on solutions to these concerns through the design, creation, and implementation of a spatial database, along with custom functions, joins, and relationships.

DATA DESCRIPTION AND SOURCES

Three main sources form our database: the eBird Basic Dataset from Cornell, the Protected Areas Database of the United States from USGS, and TIGER/Line Shapefiles from the U.S. Census Bureau. The download for the former source consists of a tab-delimited text file, imported using the SQL copy

command, and assigned point geometry derived from longitude and latitude field values. The latter two sources were obtained as shapefiles and uploaded using the PostGIS shp2pgsql and psql command line expressions.

EXPECTATIONS

Our goal in combining these resources is to develop a bird species search tool that is both comprehensive and practical at a local scale. Although the raw data we obtained has the potential for extensive analytical research, our database will primarily cater to the average outdoor enthusiast and thus many of the original fields proved unnecessary. We encountered several challenges working with such a vast dataset of species sightings. Fortunately, we were able to resolve several of these through query experimentation and will discuss those challenges and solutions in the sections below.

SECTION 2: DATABASE DESIGN, IMPLEMENTATION & MANIPULATION

We translated our source data into four spatial entities (see the **ebird**, **area**, **state**, and **county** entities in Figure 1 and tables in Figure 2) placed in three groups, which are later expressed as schemas during the database implementation phase. The first consists of species observations depicted as points, another contains protected lands with polygon geometry, and the last group holds both states and counties represented as polygons. Given the amount of overlap in field values within each entity, minimizing redundancy via normalization became the next priority. The bulk of non-spatial entities and attributes seen in Figure 1 originate from the **area** entity. Attributes for some might seem obvious based on names like manager_name and manager_type (e.g. values *SFW*, *State Fish and Wildlife* for mang_name, d_mang_nam relate to *STAT*, *State* for mang_type, d_mang_typ), but others may need further explanation. The most relevant for casual users is **access**. Composed of four domains, this entity details the level of public access permissible:

usgs_pad.access
OA - *Open Access*
RA - *Restricted Access*
XA - *Closed*
UK - *Unknown*

Some useful, though more niche entities derived from **area** include:

usgs_pad.designation
AGRE - *Agricultural Easement*
…
NT - *National Scenic or Historic Trail*
…
WSR - *Wild and Scenic River*

usgs_pad.gap_status

1 - *managed for biodiversity - disturbance events proceed or are mimicked*

2 - *managed for biodiversity - disturbance events surpressed*

3 - *managed for multiple uses - subject to extractive (e.g. mining or logging) or OHV use*

4 - *no known mandate for protection*

usgs_pad.iucn_cat

Ia - *Strict nature reserves*

Ib - *Wilderness areas*

II - *National park*

III - *Natural monument or feature*

IV - *Habitat / species management*

V - *Protected landscape / seascape*

VI - *Protected area with sustainable use of natural resources*

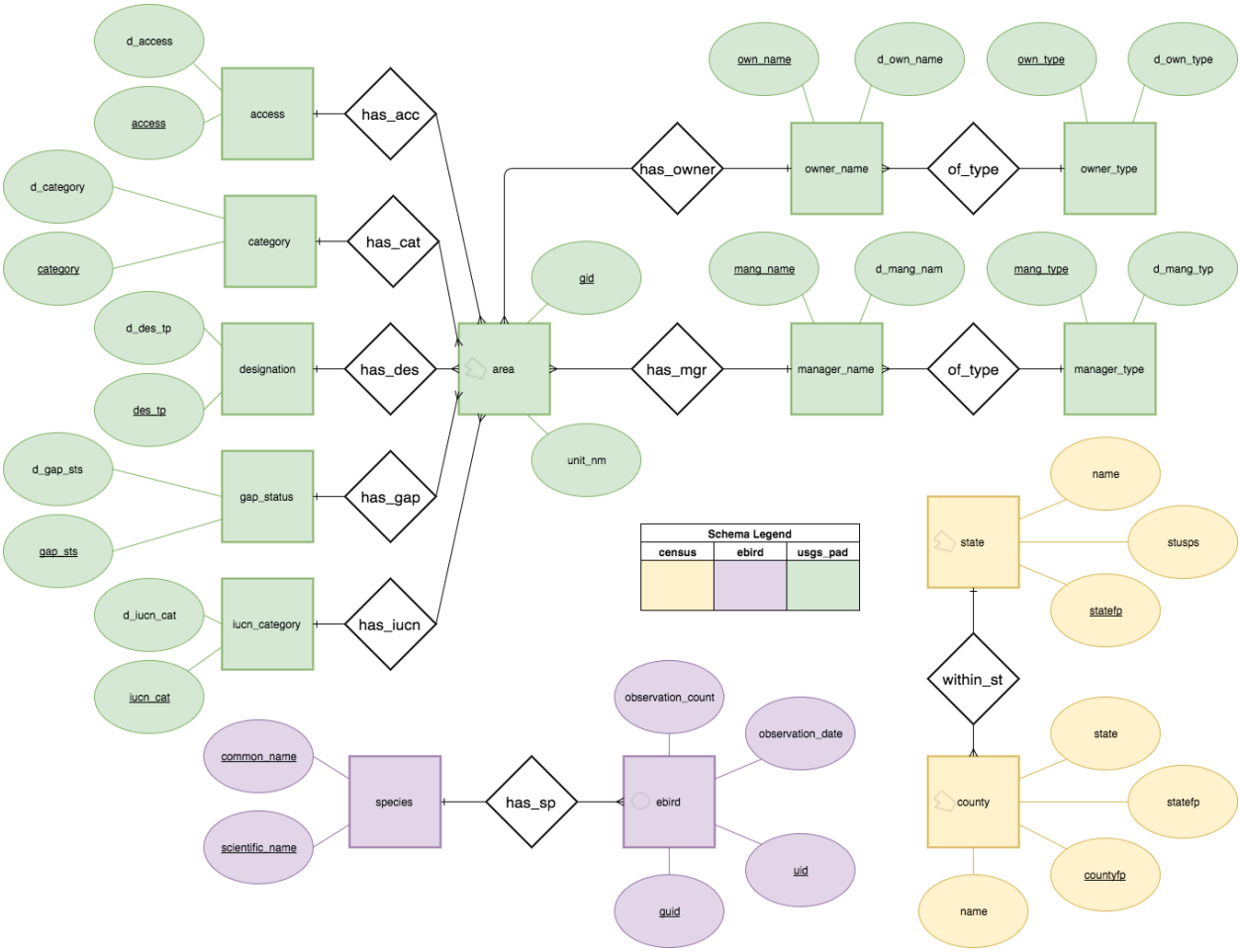

Figure 1. Entity-Relationship diagram

Figure 2. Logical diagram

DATABASE IMPLEMENTATION

We used data from existing (third-party) databases for this project, namely eBird, the USGS Protected Area Database (USGS PAD), and the US Census Tiger data. In each case, we created separate schemas to hold these databases. As mentioned above, our research questions do not require that we keep all of the attributes in each table, so we decided to eliminate fields that were not useful for our questions. Further, the database providers often included duplicative fields to make it easier for outside users read and query. For example, the USGS PAD includes both a coded value and descriptive value for eight fields (16 in total). In the case of the USGS PAD, we created list tables that are foreign keys and reduce the number of fields stored in the core table. We created a similar table in eBird schema to hold common and scientific names.

To import the eBird data, we used an example found on the web (https://github.com/weecology/retriever/issues/90) to create a properly formatted table. Once completed, we uploaded the US eBird data downloaded from their website to the database. The SQL query **create_ebird_table.sql** details the process and is attached to this report.

Data for the **census** and **usgs_pad** schemas were uploaded using the PostGIS SHP2PGSQL command. Once imported, the USGS PAD table (**usgs_pad.area**) was used to create several additional list tables to eliminate 'duplicate' attributes, as described above. Please reference the ER Model (see Figure 1) and Logical Diagram (see Figure 2) for details. We used the state name (**census.state.name**) as a foreign key

in the USGS PAD table (**usgs_pad.area.state_nm**). There are no longer direct linkages between the tables in different schemas; all other connections are spatial in nature.

For each table, we created indexes on fields pertinent to our core queries. The list includes:

ebd.ebird

```
CREATE INDEX ebird_geom_idx ON ebd.ebird USING GIST (geom);
CREATE INDEX ebird_common_name_idx ON ebd.ebird (common_name);
CREATE INDEX ebird_scientific_name_idx ON ebd.ebird (scientific_name);
CREATE INDEX ebird_month_idx ON ebd.ebird
(date_part('month',observation_date);
CREATE INDEX ebird_year_idx ON ebd.ebird
(date_part('year',observation_date);
CREATE INDEX ebird_obsdate_idx ON ebd.ebird (observation_date);
```

usgs_pad.area

```
CREATE INDEX area_geom_idx ON usgs_pad.area USING GIST (geom);
CREATE INDEX area_loc_nm_idx ON usgs_pad.area (loc_nm);
CREATE INDEX area_mang_nm_idx ON usgs_pad.area (mang_nm);
CREATE INDEX area_unit_nm_idx ON usgs_pad.area (unit_nm);
```

census.state

```
CREATE INDEX state_geom_idx ON census.state USING GIST (geom);
CREATE INDEX county_name_idx ON census.state (name);
CREATE INDEX county_statefp_idx ON census.state (statefp);
CREATE INDEX county_stusps_idx ON census.state (stusps);
```

census.county

```
CREATE INDEX county_geom_idx ON census.county USING GIST (geom);
CREATE INDEX county_name_idx ON census.county (name);
```

DATABASE MANIPULATION

To answer our core questions, we created several example queries. As noted above, our core questions for this database are:

- What bird species can I see nearby?
- Where can I see a certain species nearby?
- When can I see a certain species nearby?

Although we have created some samples of queries to help answer the questions above, the examples below are hardly comprehensive.

FIND PROTECTED AREAS (WITHOUT ACCESS RESTRICTIONS) IN A PARTICULAR COUNTY THAT HAVE RECORDS OF A PARTICULAR SPECIES.

```
**QUERY**
-- A query to find accessible protected areas with certain species.
WITH local_sp AS (
SELECT eb.geom FROM ebd.ebird AS eb
JOIN census.county AS cn ON ST_INTERSECTS(eb.geom, cn.geom)
WHERE
      cn.name = 'Dane' AND
      cn.state = 'WI' AND
      eb.common_name = 'Sandhill Crane'
)
SELECT up.unit_nm as protected_area, ac.d_access as access_type
FROM usgs_pad.area AS up
JOIN local_sp ON ST_INTERSECTS(local_sp.geom, up.geom)
JOIN usgs_pad.access as ac on up.access=ac.access
WHERE up.access NOT IN ('UK','XA')
GROUP BY up.unit_nm, ac.d_access
ORDER BY ac.d_access, up.unit_nm;


**RESULTS**
            protected_area              | access_type
----------------------------------------+-------------
 Black Earth Creek Fishery Area         | Open Access
 Capitol Springs Centennial State Park  | Open Access
 Cherokee Marsh Fishery Area            | Open Access
 Cross Plains State Park                | Open Access
 Dane County Waterfowl Production Area  | Open Access
 Door Creek                             | Open Access
 Dorn Creek Fishery Area                | Open Access
 E-Way                                  | Open Access
 Glacial Drumlin - Cattell              | Open Access
 Glacial Drumlin State Trail            | Open Access
 ... list continues ...
```

CREATE A LIST OF SPECIES YOU MIGHT EXPECT TO SEE IN A SPECIFIC PROTECTED AREA.

```
**QUERY**
-- A query to create a bird list for a protected area in a particular
month
--    In this example, since Devil's Lake State Park has several
polygons and not just one,
--    I used LIKE in the WHERE clause to capture them all
SELECT
      e.common_name,
      e.scientific_name
FROM
      ebd.ebird AS e
JOIN usgs_pad.area AS pa ON ST_Intersects (pa.geom, e.geom)
WHERE
      LOWER (pa.unit_nm) LIKE 'devils lake%'
      AND pa.state_nm = 'WI'
      AND date_part('month', e.observation_date) = '5'
GROUP BY
      common_name,
      scientific_name
ORDER BY
      e.common_name;



**RESULTS**
      common_name      |    scientific_name
-----------------------+-----------------------
 Acadian Flycatcher    | Empidonax virescens
 Accipiter sp.         | Accipiter sp.
 Alder Flycatcher      | Empidonax alnorum
 American Coot         | Fulica Americana
 American Crow         | Corvus brachyrhynchos
 American Goldfinch    | Spinus tristis
 American Kestrel      | Falco sparverius
 American Redstart     | Setophaga ruticilla
 American Robin        | Turdus migratorius
 American Tree Sparrow | Spizelloides arborea
 ... list continues ...
```

FIND ALL OF THE PROTECTED AREAS WITH WHOOPING CRANE SIGHTINGS SINCE 2011.

```
**QUERY**
SELECT DISTINCT pa.unit_nm,pa.state_nm
FROM usgs_pad.area as pa
JOIN ebd.ebird as e ON ST_Intersects(pa.geom,e.geom)
WHERE common_name='Whooping Crane'
     and observation_date>'2011-01-01'
GROUP BY pa.unit_nm, pa.state_nm
ORDER BY pa.state_nm, pa.unit_nm;


**RESULTS**
                  unit_nm                    | state_nm
---------------------------------------------+----------
 Joe Wheeler State Park                      | AL
 Mallard-fox Creek Wildlife Management Area  | AL
 State-owned Submerged Lands                 | AL
 Wheeler Closing Order Boundary              | AL
 Wheeler National Wildlife Refuge            | AL
 Wheeler Reservoir Retained Land             | AL
 Arbuckle Airfield                           | FL
 Avon Park AF Range                          | FL
 Blackwater Creek Preserve                   | FL
 Blue Cypress Conservation Area              | FL
 ... list continues ...
```

FIND THE NEAREST PUBLICLY ACCESSIBLE PROTECTED AREA WHERE ONE MIGHT FIND A PARTICULAR SPECIES THIS MONTH.

QUERY

```
-- A query to find the nearest PA where a species can be seen (this
month and using data since 2015)
SELECT
      up.unit_nm AS pa_name,
      up.state_nm AS state,
      ST_Distance (up.geom :: geography, ST_SetSRID(ST_POINT(-
89.7485322, 43.4687975),4326)) /
               1000 AS distance_km
FROM
      usgs_pad.area AS up
JOIN ebd.ebird e ON ST_INTERSECTS (e.geom, up.geom)
WHERE
      e.common_name = 'American Bittern'
      AND up.access NOT IN ('XA', 'UK')
      AND e.observation_date >= '2015-01-01'
      AND date_part('month', e.observation_date) = date_part('month',
now())
GROUP BY up.unit_nm, up.state_nm, up.geom
ORDER BY
      up.geom <-> ST_SetSRID (
            ST_POINT (- 89.7485322, 43.4687975),
            4326
      )
LIMIT 10;
```

RESULTS

pa_name	state	distance_km
Statewide Natural Area	WI	3.60143062977
Dane County Waterfowl Production Area	WI	32.53233368161
Quincy Bluff And Wetlands Natural Area	WI	42.59313832168
Grand River Marsh Wildlife Area	WI	47.31264756116
Germania Wildlife Area	WI	59.74288461056
Hook Lake/grass Lake Wildlife And Natural Area	WI	65.19609125151
Necedah National Wildlife Refuge	WI	68.56909126759
Leola Marsh Wildlife Area	WI	79.36120575309
White River Marsh Wildlife Area	WI	69.42757766956
Wood County Forest	WI	89.32880702701

(10 rows)

GIVEN THE COMPLEXITY OF THE ABOVE QUERY, WE CREATED A FUNCTION THAT OUTPUTS THE SAME INFORMATION BUT SIMPLIFIES CRITERIA ENTERED BY THE USER.

```
CREATE FUNCTION ebd.near_pa_for_sp(float8, float8, varchar, int4)
RETURNS SETOF ebd._near_pa AS $BODY$SELECT
      up.unit_nm AS pa_name,
      up.state_nm AS state,
      ST_Distance (up.geom :: geography, ST_SetSRID(ST_POINT($1,
$2),4326)) / 1000 AS distance_km
FROM
      usgs_pad.area AS up
JOIN ebd.ebird e ON ST_INTERSECTS (e.geom, up.geom)
WHERE
      e.common_name = $3
      AND up.access NOT IN ('XA', 'UK')
      AND e.observation_date >= '2014-01-01'
      AND date_part('month', e.observation_date)::int = $4
GROUP BY up.unit_nm, up.state_nm, up.geom
ORDER BY
      up.geom <-> ST_SetSRID (
            ST_POINT ($1, $2),
            4326
      )
LIMIT 10;$BODY$
LANGUAGE 'sql' VOLATILE COST 100
ROWS 10
;
```

The query ends up looking like this:

```
SELECT ebd.near_pa_for_sp(-89.4021755,43.075816,'Sandhill Crane',5);
```

SECTION 3: RESULTS & CONCLUSION

Our database implementation proved successful when running the queries noted above. Both authors hope to reuse the database implementation in a future course. The International Crane Foundation will use the data for a more specific education/outreach message by identifying protected areas that play a key role as nesting, foraging, wintering and migration stopover locations for the Whooping Crane, an endangered species.

Even so, we faced several challenges as part of this project. The largest and most challenging hurdle was the sheer size of the eBird dataset. There are 274,141,000 eBird observation records in the dataset we used for this project. Loading the data to a cloud implementation of PostgreSQL forced us to play with a variety of tools beyond the scope of the course. Indexes of various types helped us tune the database to improve its speed. We employed an index based on the ST_geohash() command and used that index to

cluster the eBird records. This ensured sightings near one another spatially are also close to one another within the database, which improved spatial query response time while using the **ebd.ebird** table.

We were already too far along in our work on the project when we reached the modules on using MongoDB. It would make an interesting project to compare and contrast using MongoDB vs. PostgreSQL to answer these questions.

SECTION 4: REFERENCES

eBird Basic Dataset, Version: EBD_US_relFeb-2017. Cornell Lab of Ornithology, Ithaca, New York. Feb 2017.https://ebird.org/ebird/data/download Accessed Mar 2017.

Protected Areas Database of the United States (PAD-US), Version: PADUS1_4Shapefile. U.S. Geological Survey, Gap Analysis Program (GAP). May 2016. https://gapanalysis.usgs.gov/padus/data/download/ Accessed Mar 2017.

TIGER/Line Shapefiles, Versions: tl_2016_us_state, tl_2016_us_county. U.S. Census Bureau. 2016. https://www.census.gov/geo/maps-data/data/tiger-line.html Accessed Apr 2017.