Bringing your legacy data into the future.

Using MS Access and ArcGIS together.

Melissa Albino Hegeman
October 19, 2016
NEARC Fall Meeting
Bringing your legacy data into the future.

Using MS Access and ArcGIS together.
The Goal

DMR wants all of our data digitized, so we can:

• Standardize
• Automate
• Reduce our errors
• Make better “stuff”
But...

What is the best way to handle the historic points, lines and polygons we have hanging around?
What to do with all the old data?

How do we handle spatial data from the pre-ArcGIS days?

• Enter data in to Access
• Use Python to query Access and to create feature classes from that database
Database Requirements

- Dead simple
- No crazy stuff for me to manage
- The data entry process should be very easy
- Everyone has to be able to access it
Aquaculture Leases

- One permit holder can have multiple leases
- Varying lease shapes: circles, squares and polygons
- One lease site can have a varying number of coordinates.
- ~75 lease sites
- WGS84
# Database Tables

<table>
<thead>
<tr>
<th>Permit Holders</th>
<th>Lease Info</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>PermitHolderID</td>
<td>LeaseInfoID</td>
<td>CoordinatesID</td>
</tr>
<tr>
<td>PermitNum</td>
<td>PermitNum</td>
<td>LeaseID</td>
</tr>
<tr>
<td>PermitHolder</td>
<td>Embayment</td>
<td>Latitude</td>
</tr>
<tr>
<td></td>
<td>Access</td>
<td>Longitude</td>
</tr>
<tr>
<td></td>
<td>Area</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LeaseID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ShapeType</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Radius(ft)</td>
<td></td>
</tr>
</tbody>
</table>
The Geodatabase

1. Set up an empty geodatabase for the new features
2. Create an empty template feature classes in the right projection, with the right fields
The Process

1. Import arcpy, pandas, pyodbc
2. Connect to MS Access database
3. Copy the data into a pandas data frame
4. Separate that data frame into groups based on shape field
5. Write the circles to the feature class
6. Write the polygons to the feature class
7. Append the attributes.
Get the data out of Access

dbPath = 'K:/MRCD/aquacultureDB/shellfishManagement.accdb'
accessDriver = 'DRIVER={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=' + dbPath
cnxn = pyodbc.connect(accessDriver)

query = """""
SELECT Lease_Info2.LeaseID, Lease_Info2.ShapeType,
Coordinates2.Latitude, Coordinates2.Longitude
FROM Coordinates2
INNER JOIN Lease_Info2
ON Coordinates2.LeaseID = Lease_Info2.LeaseID;"""

sourceData = pandas.read_sql(query, cnxn)
Get the data out of Access

dbPath = 'K:/MRCD/aquacultureDB/shellfishManagement.accdb'
accessDriver = 'DRIVER={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=' + dbPath
cnxn = pyodbc.connect(accessDriver)

query = """SELECT Lease_Info2.LeaseID, Lease_Info2.ShapeType, Coordinates2.Latitude, Coordinates2.Longitude FROM Coordinates2 INNER JOIN Lease_Info2 ON Coordinates2.LeaseID = Lease_Info2.LeaseID;""

sourceData = pandas.read_sql(query, cnxn)
Get the data out of Access

dbPath = 'K:/MRCD/aquacultureDB/shellfishManagement.accdb'
accessDriver = 'DRIVER={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=' + dbPath

cnxn = pyodbc.connect(accessDriver)

query = """SELECT Lease_Info2.LeaseID, Lease_Info2.ShapeType, Coordinates2.Latitude, Coordinates2.Longitude
FROM Coordinates2
INNER JOIN Lease_Info2
ON Coordinates2.LeaseID = Lease_Info2.LeaseID;""

sourceData = pandas.read_sql(query, cnxn)
The Process

1. Connect to MS Access database
2. Copy the data into a pandas data frame
3. Separate that data frame into groups based on shape field
4. Write the circles to the feature class
5. Write the polygons to the feature class
6. Append the attributes.
circles = sourceData[sourceData.ShapeType=='Circle']
polygons = sourceData[sourceData.ShapeType=='Polygon']
The Process

1. Connect to MS Access database
2. Copy the data into a pandas data frame
3. Separate that data frame into groups based on shape field
4. Write the circles to the feature class
5. Write the polygons to the feature class
6. Append the attributes.
Circles

1. Create an empty point feature class for the center of each circle in WGS84
2. Open an insert cursor on the feature class we just made
3. Iterate through our circles dataframe and cursor.insertRow for each row
4. Reproject that feature to NAD83
5. Create a 250’ buffer
```python
cursor = arcpy.da.InsertCursor(tempFileWGS84,["SHAPE@XY", 'LeaseID'])
for x in circles.iterrows():
    #returns a tuple: index# and row
    #all of our information is in second part of the tuple.
    data = x[1]
    LeaseID = data[0]
    lon = data[3]
    lat = data[2]
    point = lon, lat
    cursor.insertRow([point,LeaseID])

del cursor
```
Circles

1. Create an empty point feature class for the center of each circle in WGS84
2. Open an insert cursor on the feature class we just made
3. Iterate through our circles dataframe and cursor.insertRow for each row
4. Reproject that feature to NAD83
5. Create a 250’ buffer
The Process

1. Connect to MS Access database
2. Copy the data into a pandas data frame
3. Separate that data frame into groups based on shape field
4. Write the circles to the feature class
5. Write the polygons to the feature class
6. Append the attributes.
Polygons

1. Create a polygon feature class in WGS84
2. Get a list of all the unique lease IDs
3. Open an insert cursor
4. For each lease ID
   a. Create a df of only the cords belonging to the lease
   b. Create an empty array
   c. Go through each coord pair and add it to the array
   d. Use the array to create a polygon, insert the row
5. Reproject that feature to NAD83
6. Append to the circles feature class
The Process

1. Connect to MS Access database
2. Copy the data into a pandas data frame
3. Separate that data frame into groups based on shape field
4. Write the circles to the feature class
5. Write the polygons to the feature class
6. Append the attributes.
Attribute Table

1. Create a table, based on a template
2. Query the Access database into a pandas dataframe
3. Create an insert cursor for the empty table
4. For each row in the pandas dataframe, insert row into table
5. Append the table to the feature class
Thank You

• Melissa Albino Hegeman
• Data and GIS Coordinator
• melissa.albino@dec.ny.gov
• (631) 444-0464
• https://github.com/mhegeman/arcgisandaccess_nearc

Connect with us:
Facebook: www.facebook.com/NYSDEC
Twitter: twitter.com/NYSDEC
Flickr: www.flickr.com/photos/nysdec