# Lab 1 Report

Title: ETL pipelines using web APIs
Notice: Dr. Bryan Runck
Author: Diego Osorio
Date: 10/5/2022

**Project Repository:** https://github.com/osori050/GIS5571/tree/main/Lab1
**Google Drive Link:**
**Time Spent:** 26

## Abstract

The Minnesota Geospatial Commons (MGC), Google Places, and North Dakota Weather Network (NDAWN) Center web APIs were deconstructed through reverse engineering to depict their conceptual models. It was found that the most straightforward API is MGC while the most complex is Google Places. Additionally, a script in ArcPro and ArcOnline was developed to build ETL pipelines able to download a pair of datasets from each API and integrate them by using spatial joins. The results were verified by comparing the shapefile coordinates to ArcPro, and the spatial join attribute tables to those of the inputs which confirmed the production of high-quality results.

## Problem Statement

This lab seeks to decompose spatial web API interfaces of the Minnesota Geospatial Commons (MGC), Google Places, and the North Dakota Agricultural Weather Network Center (NDAWN) to understand the conceptual models by doing reverse engineering. These models are to be compared to then create ETL (extract, transform, and load) routines. Scripts on Jupyter Notebooks are created to automatically download a pair of datasets from these websites, transform the data to the same coordinate reference system (both geographic and projected), spatially join them, print to screen the head of the attribute table showing the joining, and finally storing the integrated data in a geodatabase. Figure 1 illustrates the ETL pipeline.

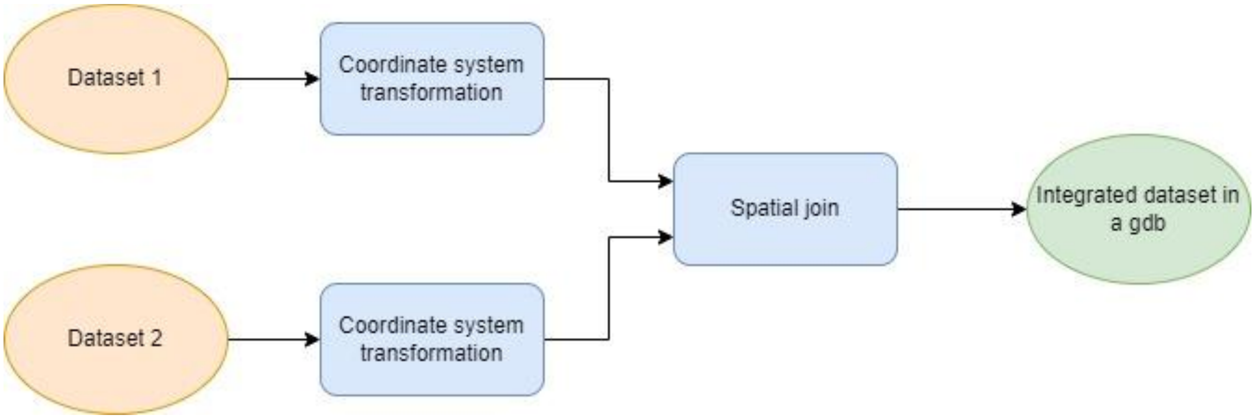

Dataset 1
Coordinate system transformation
Dataset 2
Coordinate system transformation
Spatial join
Integrated dataset in a gdb

*Figure 1. ETL pipeline*

*Table 1. Data required for spatial analysis*

| # | Requirement | Defined As | (Spatial) Data | Attribute Data | Dataset | Preparation |
|---|---|---|---|---|---|---|
| 1 | Geological units of Minnesota | Raw input polygon vector layer with ecological land type associations | Geological areas | Land type | Mn GeoSpatial Commons | Project to WGS 1984 UTM Zone 15N |
| 2 | Springs in Minnesota | Point vector layer with the headwater | Location points | Location | Mn GeoSpatial Commons | |
| 3 | University of Minnesota | String with the bounding box coordinates of the university | Coordinates | | Google Places | Convert coordinates from strings to points and then to polygons. Project to WGS 1984 UTM Zone 15N |
| 4 | The Huntington Bank Stadium | String with the bounding box coordinates of the stadium | Coordinates | | Google Places | |
| 5 | Meteorological data | Weekly measurement of temperature and solar radiation | Coordinates | Temperature (F) and Solar Radiation (Lys) | NDAWN Center | Convert csv files to vector layers and then, Project to WGS 1984 UTM Zone 15N |

## Input Data

The two datasets downloaded from the MGC are created by the Minnesota Department of Natural Resources (DNR). One dataset is a polygon shapefile (shp) with the ecological land type associations of Minnesota and the other one is a point shp with the springs in Minnesota. The information obtained from Google Places through a personal API key corresponds to raw data in code containing the coordinates of the campus of the University of Minnesota in Minneapolis and The Huntington Bank Stadium. The data retrieved from the NDAWN Center are csv files with weekly measures of temperature and solar radiation beginning on September 23 of 2022 in the stations Ada and Adams. These files have the location of the stations.

*Table 2. Input data*

| # | Title | Purpose in Analysis | Link to Source |
|---|---|---|---|
| 1 | Ecological Land Type Associations of Minnesota | Raw input dataset for spatial analysis to determine the ecological land type associations where the springs of Minnesota are located | https://gisdata.mn.gov/dataset/geos-land-type-associations |
| 2 | Springs in Minnesota | | https://gisdata.mn.gov/dataset/env-mn-springs-inventory |

| | | | |
|---|---|---|---|
| 3 | University of Minnesota, Minneapolis | Raw input strings containing coordinates to locate The Huntington Bank Stadium within the University of Minnesota in Minneapolis | https://maps.googleapis.com/maps/api/place/findplacefromtext/json?fields=formatted_address%2Cname%2Crating%2Copening_hours%2Cgeometry&input=University%20of%20Minnesota%20Minneapolis&inputtype=textquery&key=YOUR_API_KEY |
| 4 | The Huntington Bank Stadium | | https://maps.googleapis.com/maps/api/place/findplacefromtext/json?fields=formatted_address%2Cname%2Crating%2Copening_hours%2Cgeometry&input=The%20Huntington%20Bank%20Stadium%20Minneapolis&inputtype=textquery&key=YOUR_API_KEY |
| 5 | NDAWN weekly average temperature for the week beginning September 23, 2022 | Raw input datasets in csv files to analyze the weekly average temperature and solar radiation recorded in the stations Ada and Adams | https://ndawn.ndsu.nodak.edu/get-table.html?station=78&station=111&variable=wdavt&ttype=weekly&quick_pick=&begin_date=2022-09-23&count=1 |
| 6 | NDAWN weekly average solar radiation for the week beginning September 23, 2022 | | https://ndawn.ndsu.nodak.edu/get-table.html?station=78&station=111&variable=wdsr&ttype=weekly&quick_pick=&begin_date=2022-09-23&count=1 |

## Methods

First, through reverse engineering, the conceptual models of the APIs were figured out. Google Places is a get-request model whose response is not a spatial data file, (such as a vector file) but a JSON dictionary. This API requires the user to input the place to search by three options: text search, find place, or nearby search. If the user wants to narrow down their results, they can add place details which are organized into three categories depending on how robust they are. Likewise, the user can retrieve photos of the place(s) and has the option to set the maximum photo height and width in pixels. Place autocomplete request is another feature available that returns place predictions (Google, 2022). The conceptual model is shown in Figure 2.

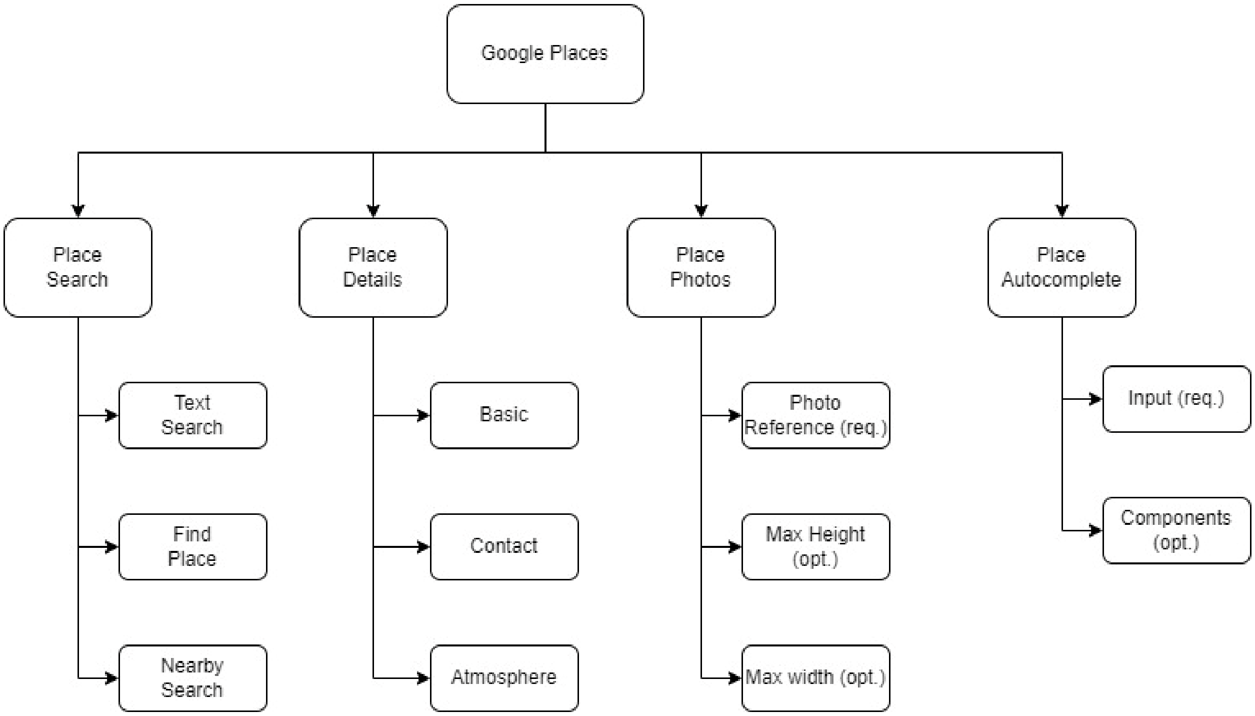


*Figure 2. Google Places' API conceptual model*

On the other hand, the NDAWN Center's API is less complex. It consists of a linear conceptual model that requires the user to input the frequency of the weather measurements (e.g., daily, weekly, yearly, etc.), the station(s), the weather variables (such as average air temperature), and the time period to retrieve a csv dataset as shown in Figure 3 (North Dakota State University, n.d.). The data can be retrieved through get-requests.

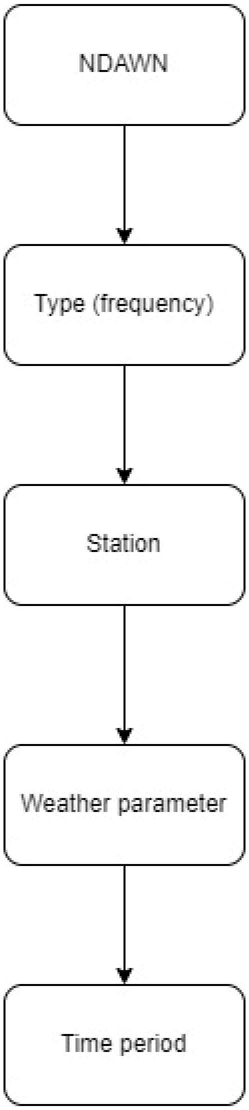


*Figure 3. NDAWN's API conceptual model*

In contrast to the above two web APIs, MGC is a CKAN post-request model where a JSON dictionary with the request is posted to the API URL. That dictionary already contains the type of data to download, and the agency and area of said agency that created the dataset. Likewise, the response to the request is a JSON dictionary as well, which then leads to downloading the data in the format requested (MGC, n.d.; Open Knowledge Foundation, n.d.). Figure 4 illustrates this conceptual model.

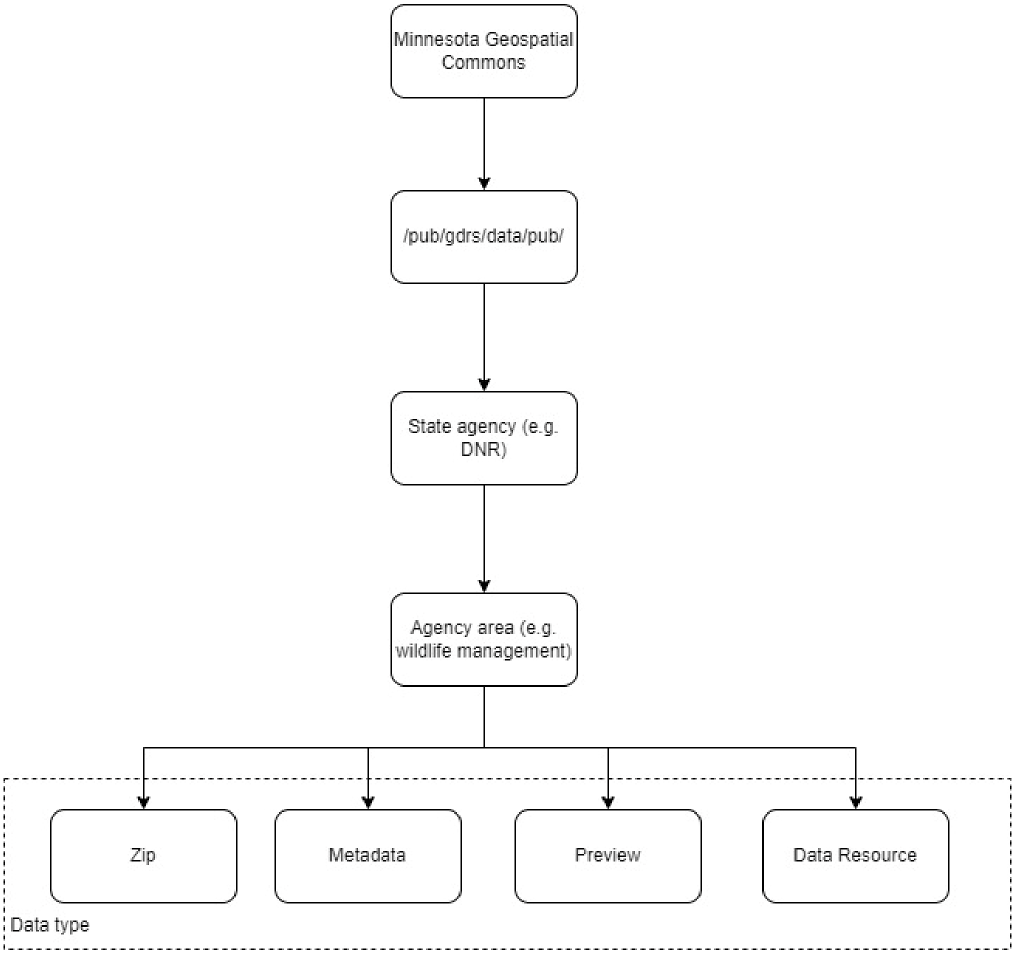


*Figure 4. MGC's API conceptual model*

Additionally, the differences among the 3 APIs can be seen when comparing their URLs. Table 3 shows that Google Places URL is made of a REST API (orange), specific service and function (purple), a response type (green), and function inputs for search (blue). NDAWN has a similar structure except for the specific service and function. Conversely, the MGC's structure is different as it does not require function inputs for search or specific service and function feature.

*Table 3. URLs from APIs*

| API | URL |
|---|---|
| Google Places | https://maps.googleapis.com/maps/api/place/findplacefromtext/json?fields=formatted_address%2Cname%2Crating%2Copening_hours%2Cgeometry&input=University%20of%20Minnesota%20Minneapolis&inputtype=textquery&key=YOUR_API_KEY |
| NDAWN | https://ndawn.ndsu.nodak.edu/table.csv?station=78&variable=wdmxt&ttype=weekly&quick_pick=&begin_date=2022-09-27&count=1 |

| | |
|---|---|
| MGC | https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_metrogis/trans_road_centerlines_log/shp_trans_road_centerlines_log.zip |

Regarding the ETL pipelines, Jupyter Notebooks on ArcPro and ArcOnline were used to develop the script. Since ArcOnline does not have the option to overwrite or remove files, try and except clauses were utilized to deal with “Dataset already exists. Failed to execute” errors.

First, the two MGC datasets were obtained through the API and downloaded as zip files, which eventually were unzipped later. Then, the coordinate system of the shapefiles was projected from NAD 1983 UTM Zone 15N to WGS 1984 UTM Zone 15N by using the WGS_1984_(ITRF00)_To_NAD_1983 geographic transformation. Later on, the spatial join was applied to integrate both datasets.

A more complex process was carried out with Google Places data. Initially, the data retrieved from the API was stored in JSON dictionaries from which the coordinates were extracted. The dictionaries only contained the northeast and southwest coordinates. Therefore, a function was created to generate the northwest and southeast coordinates and store the four pairs of XY coordinates in lists of points. Those lists were used to create point shapefiles with the WGS 1984 coordinate system. A new field was added to the attribute table to indicate one shapefile represents the University of Minnesota in Minneapolis and the other one the Huntington Bank Stadium. Later, polygon shapefiles were generated by creating bounding boxes with the point shapefiles. Both polygon shapefiles were then projected to the WGS 1984 UTM Zone 15N coordinate system (no geographic transformation needed) and integrated through a spatial join.

The NDAWN data was extracted as csv files. These files were converted to data frames removing rows 1, 2, 3, and 5 since they had a different number of columns (just 1) from the rest of the data table. The coordinates from the data frame were used to create the point shapefiles with the WGS 1984 coordinate system which was then projected to WGS 1984 UTM Zone 15N. Afterward, the spatial join was run to integrate the datasets.

At the end of the process, a geodatabase was created, and all 3 integrated datasets were included by transforming the shapefiles to geodatabase feature classes. Figure 5 illustrates the ETL pipeline workflow diagram.

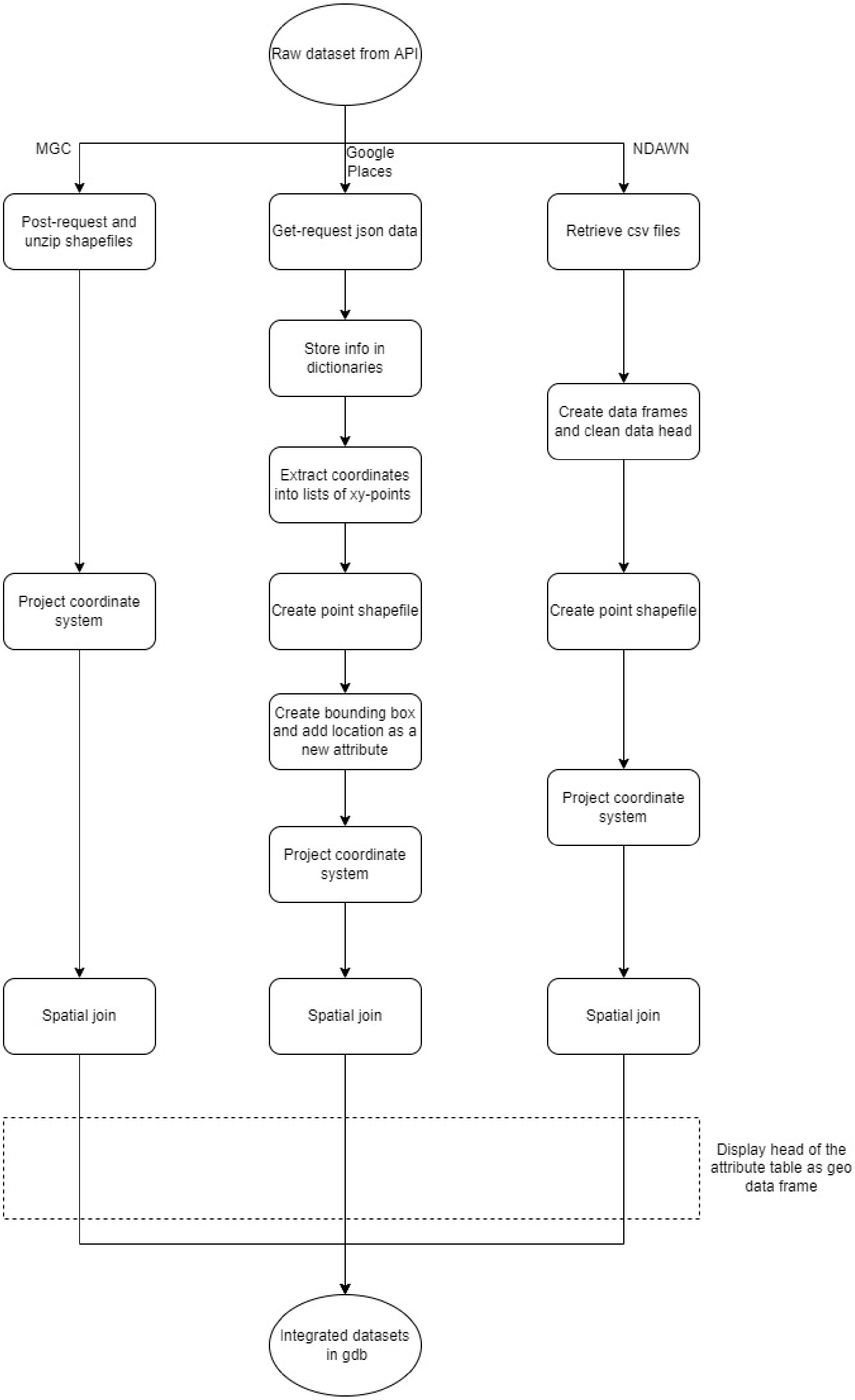
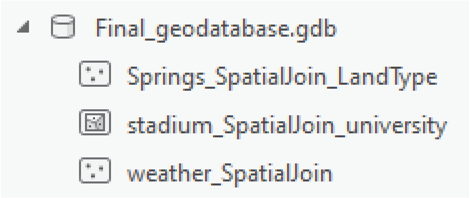

Raw dataset from API
MGC
Google Places
NDAWN
Post-request and unzip shapefiles
Get-request json data
Retrieve csv files
Store info in dictionaries
Create data frames and clean data head
Extract coordinates into lists of xy-points
Project coordinate system
Create point shapefile
Create point shapefile
Create bounding box and add location as a new attribute
Project coordinate system
Project coordinate system
Spatial join
Spatial join
Spatial join
Display head of the attribute table as geo data frame
Integrated datasets in gdb

*Figure 5. ETL pipeline conceptual model*

## Results

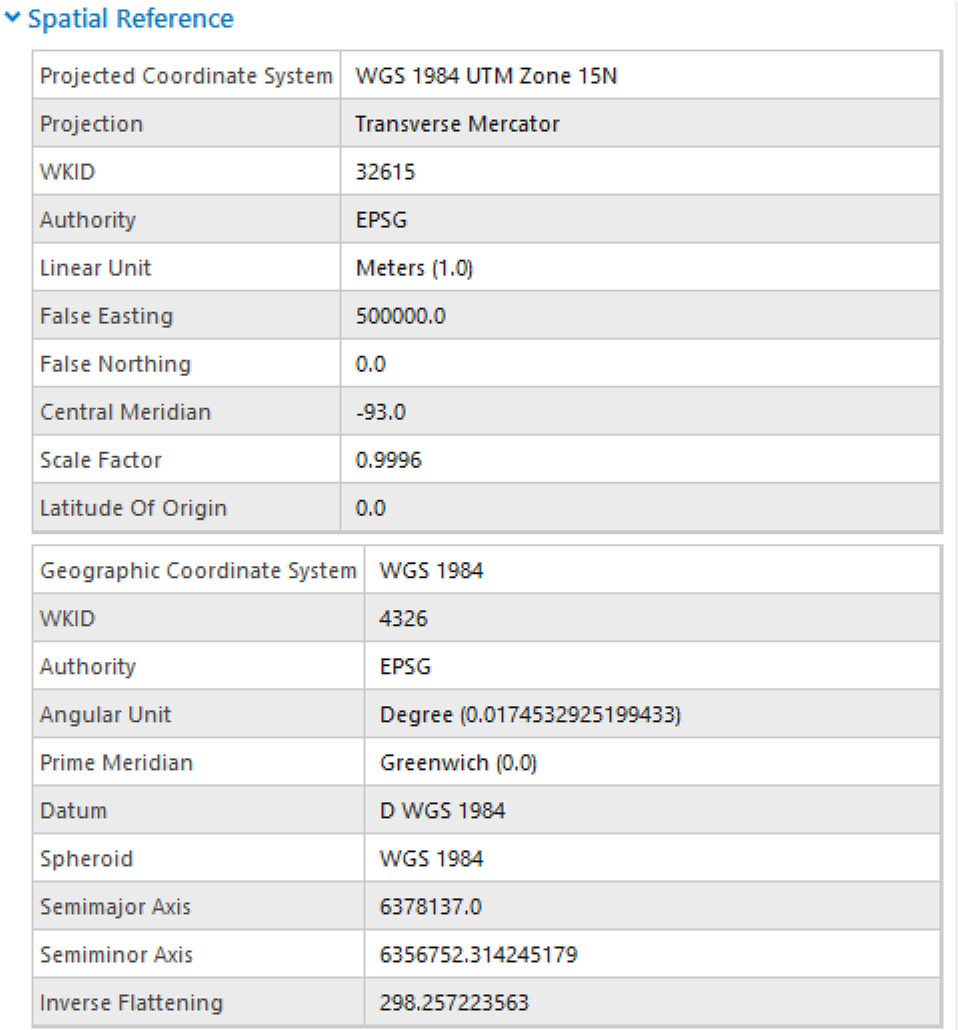
All the spatial joins were successfully executed and incorporated into a database as shown in Figure 6.



Final_geodatabase.gdb
Springs_SpatialJoin_LandType
stadium_SpatialJoin_university
weather_SpatialJoin

*Figure 6. Final dataset with the integrated datasets*

Figure 7 illustrates the final spatial reference of the three datasets utilizing the Transverse Mercator UTM Zone 15N projection with datum WGS 1984.



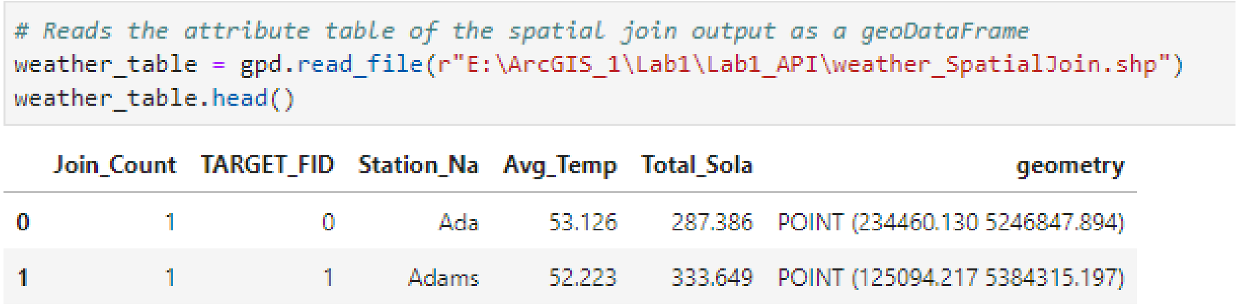
Spatial Reference

| Projected Coordinate System | WGS 1984 UTM Zone 15N |
|---|---|
| Projection | Transverse Mercator |
| WKID | 32615 |
| Authority | EPSG |
| Linear Unit | Meters (1.0) |
| False Easting | 500000.0 |
| False Northing | 0.0 |
| Central Meridian | -93.0 |
| Scale Factor | 0.9996 |
| Latitude Of Origin | 0.0 |

| Geographic Coordinate System | WGS 1984 |
|---|---|
| WKID | 4326 |
| Authority | EPSG |
| Angular Unit | Degree (0.0174532925199433) |
| Prime Meridian | Greenwich (0.0) |
| Datum | D WGS 1984 |
| Spheroid | WGS 1984 |
| Semimajor Axis | 6378137.0 |
| Semiminor Axis | 6356752.314245179 |
| Inverse Flattening | 298.257223563 |

*Figure 7. Projected and geographic coordinate systems of the integrated datasets*

Similarly, Figure 8 exposes the head of the attribute table of the spatial join output from the NDAWN Center’s datasets. This example shows the average temperature and total solar radiation were effectively merged into just one dataset and assigned to their respective stations.



```
# Reads the attribute table of the spatial join output as a geoDataFrame
weather_table = gpd.read_file(r"E:\ArcGIS_1\Lab1\Lab1_API\weather_SpatialJoin.shp")
weather_table.head()
```

| | Join_Count | TARGET_FID | Station_Na | Avg_Temp | Total_Sola | geometry |
|---|---|---|---|---|---|---|
| **0** | 1 | 0 | Ada | 53.126 | 287.386 | POINT (234460.130 5246847.894) |
| **1** | 1 | 1 | Adams | 52.223 | 333.649 | POINT (125094.217 5384315.197) |

*Figure 8. Attribute table of the integrated dataset from NDAWN Center*

## Results Verification

The results were verified by comparing the dataset coordinates to ArcPro. For example, the northeast coordinates of the Huntington Bank Stadium are 44.97999864999999 latitude and -93.21995195 longitude which are pretty much the same values displayed by ArcPro when hovering the mouse over that point as shown at the bottom of Figure 9.

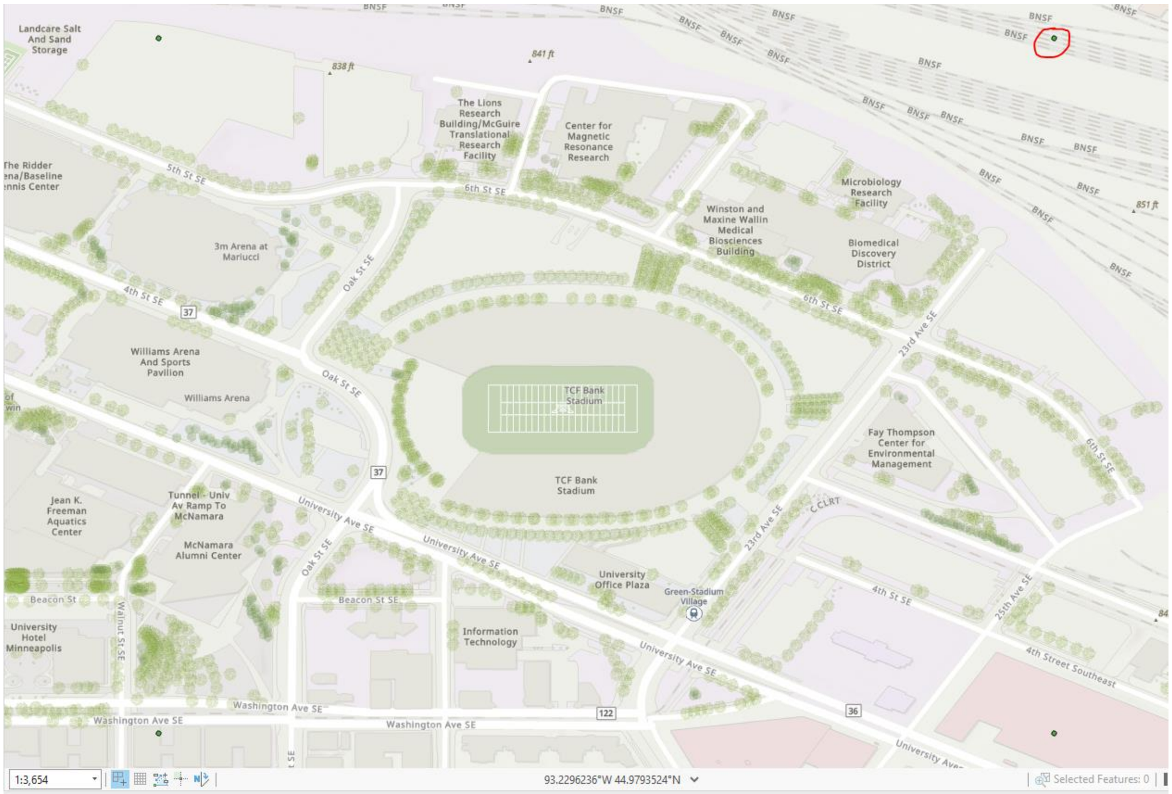
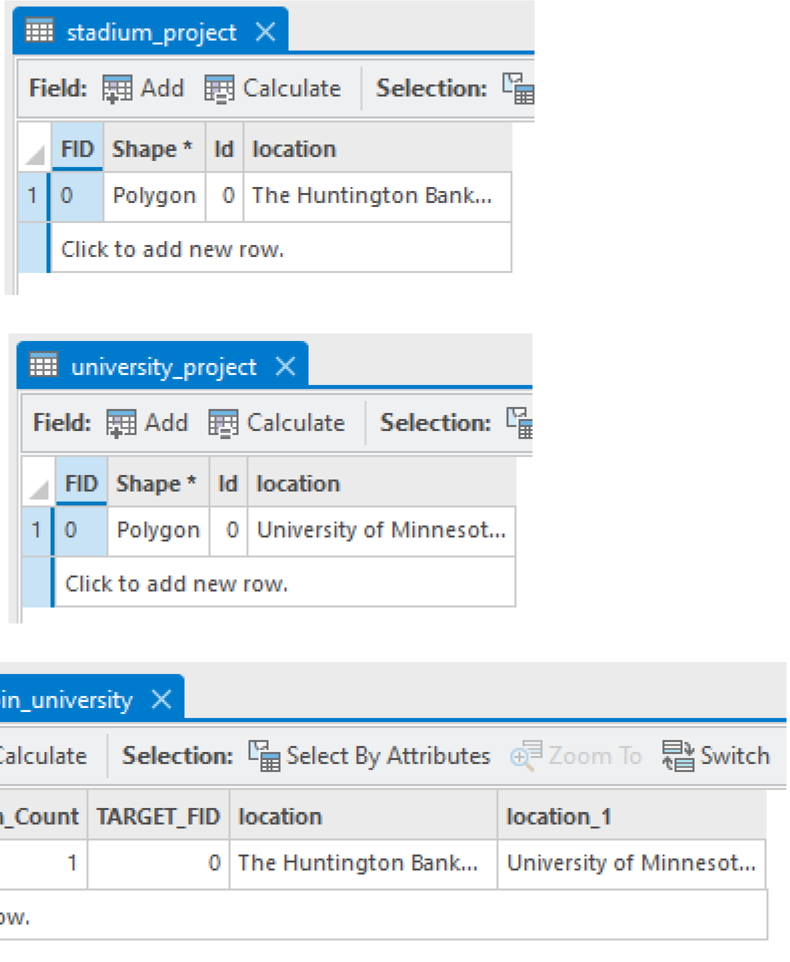

Landcare Salt And Sand Storage
The Lions Research Building/McGuire Translational Research Facility
Center for Magnetic Resonance Research
Microbiology Research Facility
Winston and Maxine Wallin Medical Biosciences Building
Biomedical Discovery District
3m Arena at Mariucci
Williams Arena And Sports Pavilion
Williams Arena
TCF Bank Stadium
TCF Bank Stadium
Fay Thompson Center for Environmental Management
Jean K. Freeman Aquatics Center
Tunnel - Univ Av Ramp To McNamara
McNamara Alumni Center
University Office Plaza
Green-Stadium Village
Information Technology
University Hotel Minneapolis
BNSF
5th St SE
6th St SE
4th St SE
Oak St SE
University Ave SE
23rd Ave SE
25th Ave SE
4th Street Southeast
Beacon St SE
Walnut St SE
Washington Ave SE
838 ft
841 ft
851 ft
1:3,654
93.2296236°W 44.9793524°N
Selected Features: 0

*Figure 9. The Huntington Bank Stadium point shapefile*

Likewise, the spatial joins were verified by comparing the attribute table of the inputs with that of the output. For example, Figure 10 shows that the two datasets from Google Places were satisfactorily joined as its attribute table contains all the information from the inputs.



| | FID | Shape * | Id | location |
|---|---|---|---|---|
| 1 | 0 | Polygon | 0 | The Huntington Bank... |

| | FID | Shape * | Id | location |
|---|---|---|---|---|
| 1 | 0 | Polygon | 0 | University of Minnesot... |

stadium_SpatialJoin_university
Field: Add Calculate
Selection: Select By Attributes Zoom To Switch

| | FID | Shape * | Join_Count | TARGET_FID | location | location_1 |
|---|---|---|---|---|---|---|
| 1 | 0 | Polygon | 1 | 0 | The Huntington Bank... | University of Minnesot... |

*Figure 10. Top to bottom, attribute tables of the Huntington Bank Stadium, University of Minnesota, Spatial Join*

All in all, the script developed in this lab yields high-quality results.

## Discussion and Conclusion

To sum up, the most different API out of the 3 is MGC due to the lack of function inputs for search and specific service and function feature which, at the same time, make it the most straightforward. Also, this API is the only one that retrieves spatial layers. On the other hand, Google Places' API is the most complex one and only retrieves JSON dictionaries that need to be then transformed into GIS layers. Photos can also be obtained here, but only when requested. Regarding the script, the ETL pipeline was built satisfactorily yielding three integrated, high-quality datasets from two inputs from each API into a geodatabase.

Additionally, this lab was very challenging since API was a topic totally new to me. First, I spent several hours on Google figuring out how each API works and then, translating that into code to develop the ETL pipeline. The process of transforming the datasets was also very complex and time-consuming.

## References

Google. (2022, September 29). *Places API*. Retrieved from Google Maps Platform: https://developers.google.com/maps/documentation/places/web-service/overview

MGC. (n.d.). *API Developer Resources*. Retrieved from https://gisdata.mn.gov/content/?q=help/api

North Dakota State University. (n.d.). *NDAWN Center*. Retrieved from https://ndawn.ndsu.nodak.edu//

Open Knowledge Foundation. (n.d.). *The CKAN API*. Retrieved from CKAN Documentation v2.1.5: https://docs.ckan.org/en/ckan-2.1.5/api.html

## Self-score

| Category | Description | Points Possible | Score |
|---|---|---|---|
| **Structural Elements** | All elements of a lab report are included **(2 points each)**:<br>Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score | 28 | 28 |
| **Clarity of Content** | Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level **(12 points)**. There is a clear connection from data to results to discussion and conclusion **(12 points)**. | 24 | 24 |
| **Reproducibility** | Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified. | 28 | 28 |
| **Verification** | Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated **(10 points)**, the method of comparison is clearly stated **(5 points)**, and the result of verification is clearly stated **(5 points)**. | 20 | 20 |
| | | 100 | 100 |