**ETL Project Report**

National Parks: Wildlife, Trails, Campgrounds, and Fees

**Team Members**

Melanie Follett

Jonathan Wu

Jingchen Liu

Jenie Nam

**Objective**

We collected data about the wildlife, hiking trails, campgrounds, and entrance fees for various national parks and loaded it into a relational database.

**Extract**

**Parks**

The data containing the park names and locations was downloaded from <https://www.kaggle.com/nationalparkservice/park-biodiversity/data> as a CSV file.

**Wildlife**

For the data regarding the types of wildlife found in each of the parks, Melanie downloaded the most recent species data from <https://irma.nps.gov/NPSpecies>. This data was last updated on 10-07-2019. The data was downloaded as 56 separate excel files, which were then converted into CSVs.

**Hiking Trails**

For extracting the different types of trails in the parks, two csv files were used: one with the trail data (<https://www.kaggle.com/planejane/national-park-trails>), as well as the parks data mentioned previously (<https://www.kaggle.com/nationalparkservice/park-biodiversity/data>). The trail csv file included the trail name, park name, as well as other information such as the longitude, latitude, and state name. For the parks csv file, park codes, park names and other additional information were included.

**Campgrounds**

We needed to extract the total amount of campgrounds for the parks in within our data set. We were able to access the National Parks API ([www.NPS.gov/api/v1](http://www.nps.gov/api/v1)) and peruse the documentation to figure out which dictionary contained the information needed for extraction. Luckily, the total number was readily available and we were able to pull the required information out by referencing the key.  We then iterated the information through the list of Park codes obtained from the initial data-set .

**Entrance Fees**

For the park entrance fee table, we extracted information from the National Parks API ([www.NPS.gov/api/v1](http://www.nps.gov/api/v1)). With a list of 56 identified national park codes, we were able to loop through the query urls (<https://developer.nps.gov/api/v1/parks?parkCode=park&fields=entranceFees&api_key=>) and gather entrance fee data for each park. One point worth noting is that each park charges differently for different entry options such as per vehicle, per motorcycle, and per individual. Since the entrance fee scenarios vary from park to park, we looped though the entrance fee dictionary for each park and pulled out all entrance fee conditions. Inside the loop, we assigned each piece of information, along with its park code, into a dictionary, and appended all of them into an empty park entrance fee list.

**Transform**

**Parks**

The data for the park names and locations needed no transforming. It was a very simple table and only included the information we needed, most importantly the park code that would be the primary key and link to all the other tables.

**Wildlife**

Using pandas, we merged all 56 CSVs files into a single dataframe. The columns of interest were selected out, and any columns with information we didn’t need were excluded. Many of the columns had NaN values. These were replaced with None, 0, or Unknown depending on the column. All the species that had NaN for Occurrence were removed from the dataframe since their presence in the parks was unknown. The index was reset to start at 1 and converted to a column so it could be used as the primary key for the species table.

**Hiking Trails**

We first imported the two csv files into pandas and created data frames using them. We cleaned them by eliminating, renaming columns, and extracting the specific columns needed such as trail name and park code. Since the hiking trails data did not have the park code, we merged the two dataframes together. We also got rid of any N/A or irrelevant data entries that were present in the files.

**Campgrounds**

The list of dictionaries collected from the API calls was converted into a dataframe via Pandas and then converted it into a CSV file to export out.  The data was very simple and did not need to be transformed.

**Entrance Fees**

To clean the data, we converted the result of the API calls to a dataframe. Inside the dataframe, we changed all the park codes to uppercase so that they are uniform among all our tables and can be used as primary/ foreign keys. We also converted the “cost” column type from string to float. The entrance fee list was then exported to csv and SQL for future uses.

**Load**

**Wildlife**

The data for this project was loaded into a relational database using SQL. We collected data from 56 different national parks. All the data tables related back to the parks table via the park code (which was the primary key). Tables were created in PGAdmin for each of the dataframes created. The parks table has a primary key (park\_code). The species table has a foreign key (park\_code) that references the parks table (park\_code). We used pandas to connect to PGAdmin and loaded the data into the tables.

**Parks & Hiking Trail Data**

We created tables in SQL that corresponded to the two data sets created: parks and trails. We made a connection between the SQL database and the pandas data frames through a connection string and input the parks and trails data into the tables. Through SQL, we combined specific columns in the two tables to create a new table park\_trails. We then double-checked that all the tables were created on pandas, and converted the file into a new csv.

**Campgrounds & Entrance Fees**

This CSV file would be joined with the other CSV files that were transformed from the extracted information from datasets and the National Parks API into an SQL database. We decided to utilize SQL as opposed to MongoDB due to the data's relational nature.  We had a natural Primary Key, the Park Code, to employ and could use that key to reference information in various tables for effective access.