**ETL Project Final Report**

**Maz King & Vic Caterina**

DATA SOURCES:

<https://github.com/planetsig/ufo-reports> (originally found via Kaggle)

<http://spacelaunchreport.com/>

BACKGROUND

We began our project by asking a question--this question would help us determine how we needed to clean and query on our data. Were intrigued by a Google Trends UFO sightings plot and several significant spikes in sighting searches. This was interesting and started to consider what would be responsible for the uptick in searches and reports of UFO sightings. We speculated that maybe orbital launches were responsible.

We sought to answer: is there any connection between date and/or location of orbital launches and UFO sightings? From there we found a csv on GitHub (via Kaggle) which was the result of web scraping of the National UFO Reporting Center (found [here](http://www.nuforc.org/)), and we copy and pasted data from the Space Launch Report (found [here](http://spacelaunchreport.com/)) since the webpage’s coding wasn’t suitable for scraping. This was the beginning of our Space Database.

**EXTRACT**

Our data came from a csv and a webpage. In order to utilize the data on the webpage, we copy and pasted directly from the webpage into Numbers and created a csv of the launch data. We created a separate csv based on location of the launch sites, which would result in a third table for our database from which we could easily join tables and identify countries in plain English instead of coded site name.

**TRANSFORM**

The two most important things to focus on were date and location. Because of this, our UFO data required significant transformation. This was thanks to the “datetime” column of the csv, which combined both date and time into one column. The date was formatted as month/date/year and the time was in a format of twenty-four hours and minutes. By importing datetime into Jupyter Notebook, we were able to isolate the data we needed via lists and loops until we had a single column containing the date. We then converted the list to a data frame with fewer columns and renamed the columns to match our SQL schema. Along the way we encountered additional hiccups such as a space after a column name in the csv, which created an issue when attempting to rename the column. Notable finding requiring significant time to identify and correct--the data contained 24:00 as a time which is not allowed. The datetime package does not recognize 24:00 date/time so we had to locate the records containing 24:00 time in the dataset and transform by setting the time to 0:00 and incrementing the date by 1 day.

Our launch and location data required less cleaning. We primarily focused on renaming columns and plucking only the columns we needed from the larger data set. Additionally, we truncated the site column so that the site codes in both our locations and launch data sets would match.

**LOAD**

Our final database, the SpaceDatabase, included three tables: ufo\_sightings, launches, and locations. The ufo\_sightings table contained a condensed version of our UFO data to include date, city, state, country, and latitude/longitude data. This dataset did not contain a unique identifier, thus the index was used as the primary key. For launches, although our data came with an “id” column the values were not unique and thus we similarly used the index as the primary key, we pulled vehicle for additional qualifying data, and the date and site of the launch. The site we then were able to join to our third table, locations, which was a list of the site ids and laymans terms for the sites (for instance, CC, the code for Cape Canaveral, has the country USA associated with it).

We chose to use Postgres for this project because we envisioned a database of connected space-related data--if we were to continue this project, we could include additional data such as return entry of rockets into Earth’s atmosphere.