CS 440 Midterm Progress Report

Project Title: International Satellite Database

Group Members

Alexander Hull (hullale@oregonstate.edu)
Ryan Crane (cranery@oregonstate.edu)
John Magenheimer (magenhej@oregonstate.edu)
Eli Sandine (sandinee@oregonstate.edu)
Cooper Hutchens (hutchico@oregonstate.edu)

Problem to be Solved

This project will facilitate monitoring large quantities of satellites. Users will be able to easily search for a particular satellite and find information about it. This information may include the location, launch date, altitude, period, and id of a given satellite. The total number of satellites launched by a country or agency should be easy to retrieve and use for making observations across international space stations.

Tools

We have chosen Python as our programming language, which will be used to pull data from various sources and interfaces with our database management system (DBMS). We chose python because it is easy to use, and is well-documented for working with DBMS. For our DBMS, we will use mySQL, because our group has the most combined experience working with it--making it easier for us to implement our database and manipulate more data. We will expand our toolset if we encounter any limitations in the programming language and DBMS that we were accustomed to in CS340.

Data Sources

Our data source is the international satellite database n2yo. This database organizes all satellites into tables by country/organization as well as by satellite category. The satellites are identified by their unique NORAD ID, and their owner is identified by a unique country/organization code. Satellite records include the aforementioned NORAD ID, name, international code, launch date, period, and magnitude. With the category and country/organization tables separated on the website, we will need to create a relational schema that will make joining them efficient.

Progress

Using python's json and urllib libraries, a python program was written to pull satellite tle and position data from n2yo in the json format generated through the n2yo API. Using SQLizer, the json files are converted into SQL databases to be imported into MySQL. This API has been helpful for gathering our starter data.

Using excel's ImportHtml command was useful in extracting the satellite Country/Organization and Categories tables from n2yo. These datasets will help us collect the rest of our data; the category names and country/organization names will be used to enumerate the satellite records that need to be retrieved from each table within the Country/Organization and Categories tables.

Challenges

Using the n2yo API requires creating an account and receiving an api key. The site limits the number of transactions a single key can make per hour to 1,000 transactions; our python program has to monitor and pause transactions when this limit is reached in order to avoid abusing the API and being blocked. This is manageable, but also time-consuming (there are more than 43,000 satellites in the database and retrieving data for a single satellite costs one transaction). A given query for all satellites in the database would take over 43 hours--and there is no API call that will give us *all* of the attributes that we need. We will be writing programs that scrape the tables directly from the n2yo web pages rather than continue using the API.

While the ImportHtml command in excel is useful for retrieving a single parent table, it is of no use for retrieving the records within each category/child table of the overarching tables. Our programs will need to iterate through each sub-table in the Country/Organization and Categories tables in order to retrieve the satellite records. Luckily, the n2yo website uses url parameters that will make it easier for us to iterate through these subtables.

Going Forward

We will work to find an efficient way of gathering and updating database entries from the source website. We will also investigate ways of combining data from different tables. This will expand the satellite information that we have available to us, and will allow for cross-referencing data. Once the data acquisition has been refined, some useful queries will have to be constructed. Searching for a particular subset of satellite data will be one component, but also including the ability to update our database will be crucial.