**ETL Project**

**Team Effort**

Due to the short timeline, teamwork will be crucial to the success of this project! Work closely with your team through all phases of the project to ensure that there are no surprises at the end of the week. Working in a group enables you to tackle more difficult problems than you'd be able to working alone. In other words, working in a group allows you to **work smart** and **dream big**. Take advantage of it!

Team Members: Kale Ford, Tim Netherton, Rod Skoglund, and Karishma Borse

**Tim**: Created the Summarized report, Created GitHub Repository

**Rod**: SQL queries, Pandas, SQL Alchemy, and finding data via GoogleBigQuery dataset.

**Kale**: Data Cleaning, Jupyter Notebook, SQL Alchemy

**Karishma**: SQL Queries and inserting into pandas.

**Project Proposal**

Before you start writing any code, remember that you only have one week to complete this project. View this project as a typical assignment from work. Imagine a bunch of data came in and you and your team are tasked with migrating it to a production data base. Take advantage of your Instructor and TA support during office hours and class project work time. They are a valuable resource and can help you stay on track.

Providing a collaborative report based upon Social Security data to find out how many births each state had for each year. This data is based upon information for everyone who registered for a social security number from 1910 to 2017.

**Finding Data**

Your project must use 2 or more sources of data.

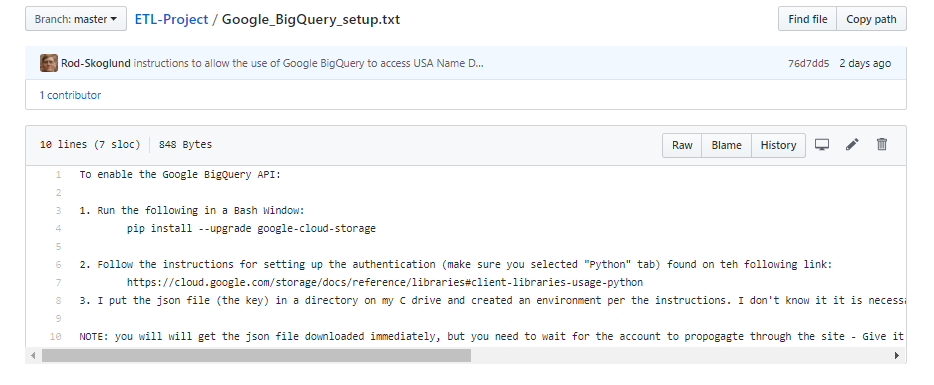
We used Kaggle to and Google BigQuery Dataset to gather our Data.

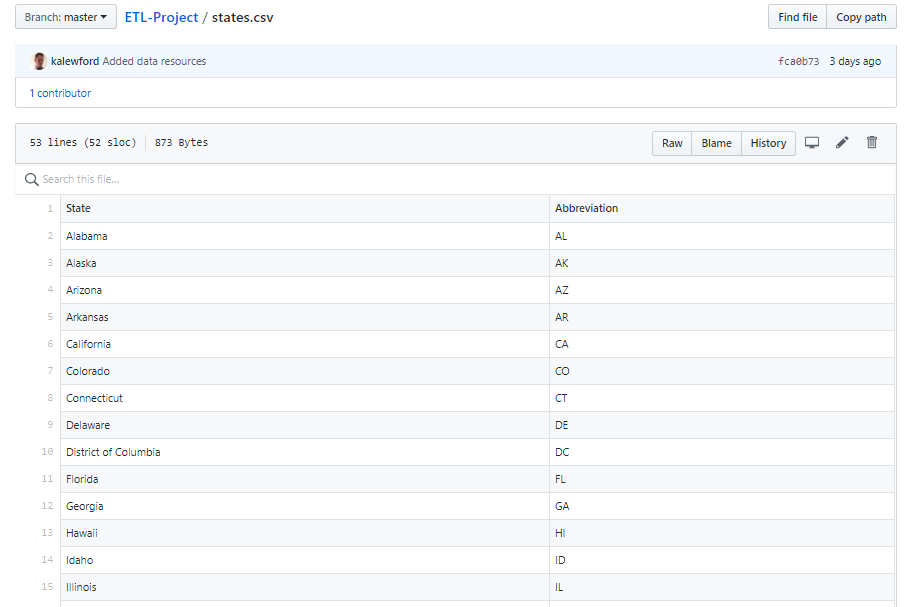
1. <https://www.kaggle.com/stansilas/us-state-county-name-codes#states.csv>

<https://www.kaggle.com/datagov/usa-names>

States. CSV

usa\_1910\_2013

1. USA Name Data (BigQuery DataSet) using the Google Cloud BigQuery. Instructions included in Repo. (See Below)

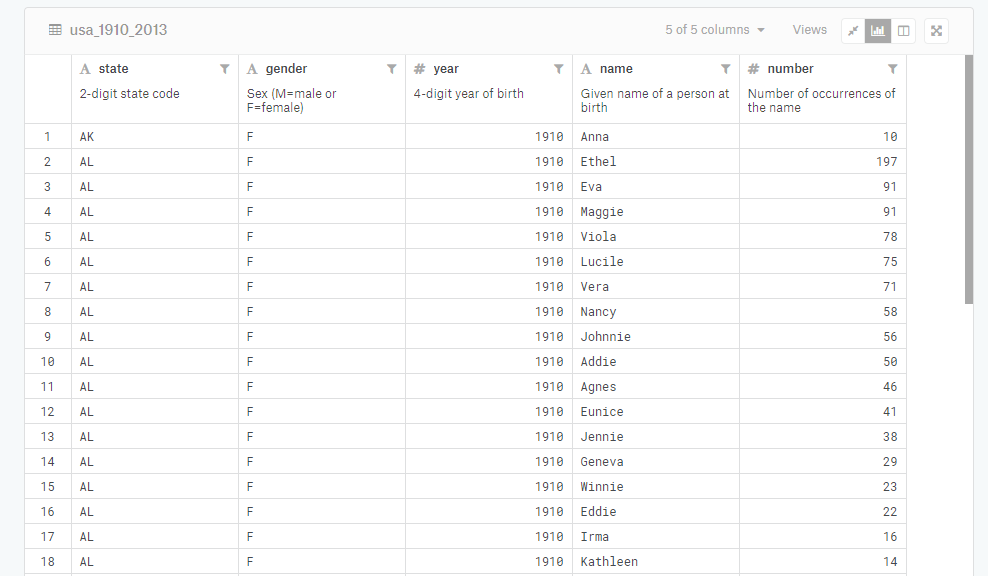


**Data Cleanup & Analysis**

Once you have identified your datasets, perform ETL on the data. Make sure to plan and document the following:

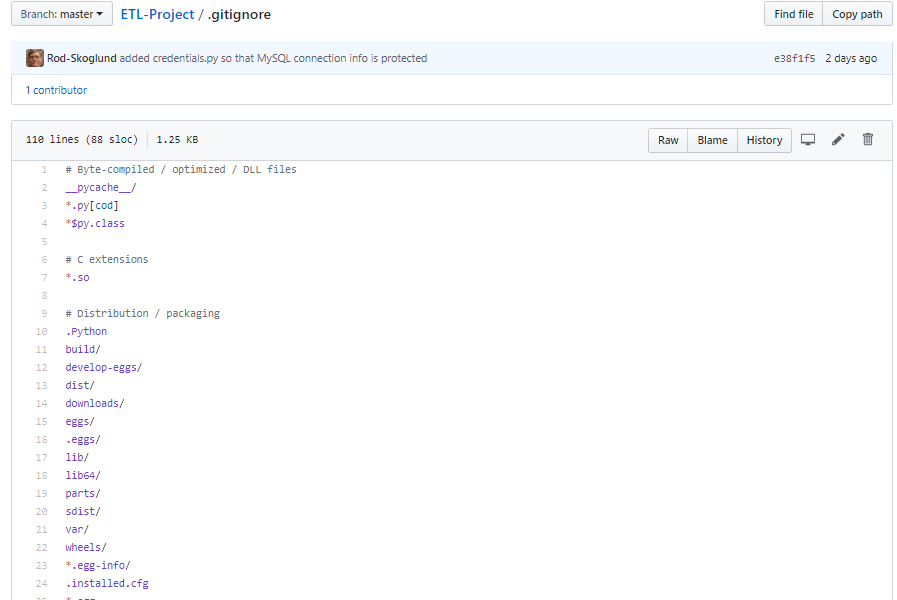
* The sources of data that you will extract from.
* The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).
* The type of final production database to load the data into (relational or non-relational).
* The final tables or collections that will be used in the production database.

You will be required to submit a final technical report with the above information and steps required to reproduce your ETL process.

* Source came from “usa\_1910\_2013” via <https://www.kaggle.com/datagov/usa-names>

We summarized by eliminating the name and the gender columns.

* Took a sum of all births for each state in each year. Eliminating individual name counts. We put in safeguards in place (Git ignore files) to make sure our SQL usernames and passwords were not included in the final report.



Rod used MySQL to run queries comparing State Name to total number of births for each state from 1910 to 2017.

* Below you’ll see the final data tables we created. We imported pandas and SQL Alchemy via Jupyter Notebook to show this file. These were put into MySQL which is a Relational Database.





**My SQL Queries:**

1. SELECT State\_Data.Full\_Name, SUM(Birth\_Data.Num\_Births) AS "Total\_Births" FROM Birth\_Data

Join State\_Data ON State\_Data.State\_Abbr = Birth\_Data.State\_Abbr GROUP BY State\_Data.Full\_Name ORDER BY Total\_Births DESC;

2. SELECT State\_Data.Full\_Name, Birth\_Data.Birth\_Year, Birth\_Data.Num\_Births FROM Birth\_Data

Join State\_Data ON State\_Data.State\_Abbr = Birth\_Data.State\_Abbr GROUP BY State\_Data.Full\_Name, Birth\_Data.Birth\_Year

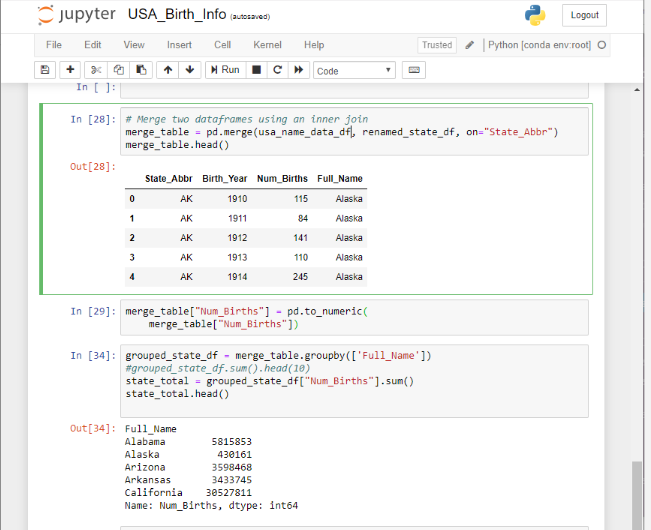
ORDER BY State\_Data.Full\_Name ASC;

3. SELECT Birth\_Year, SUM(Num\_Births) AS "Total\_Births" FROM Birth\_Data GROUP BY Birth\_Year ORDER BY Birth\_Year ASC;

**Q) Why this was chosen?** Ultimately, we wanted to analyze births in the United States. We were able to find a dataset to where it wasn’t just using the state abbreviation but the full name of the state. It was much easier to read as a result.

Below you’ll find a broad overview of the 3 Queries we ran. Refer to repo for more thorough analysis.

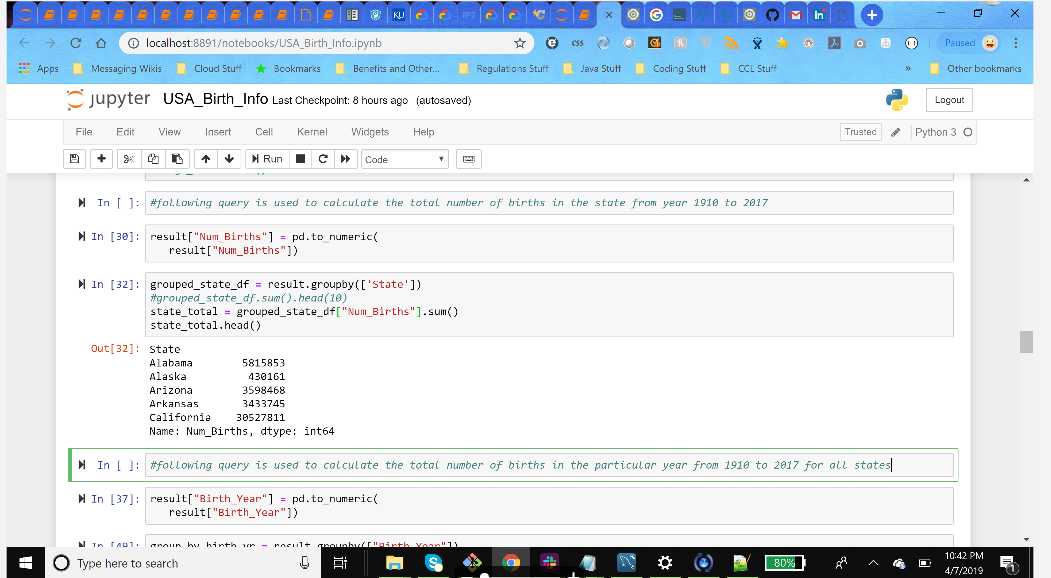
Needed to Merge two dataframes. (below)

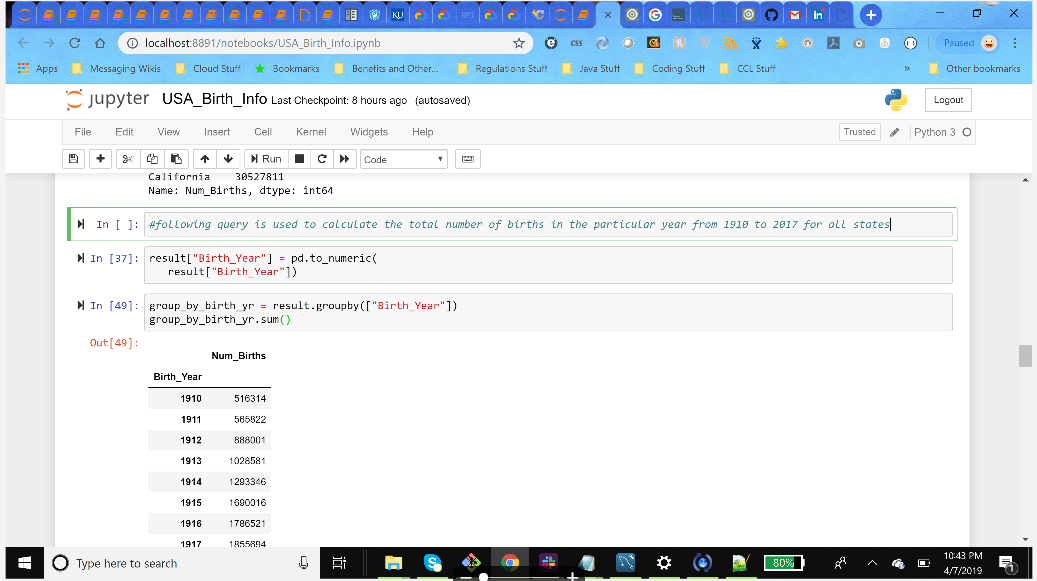
****

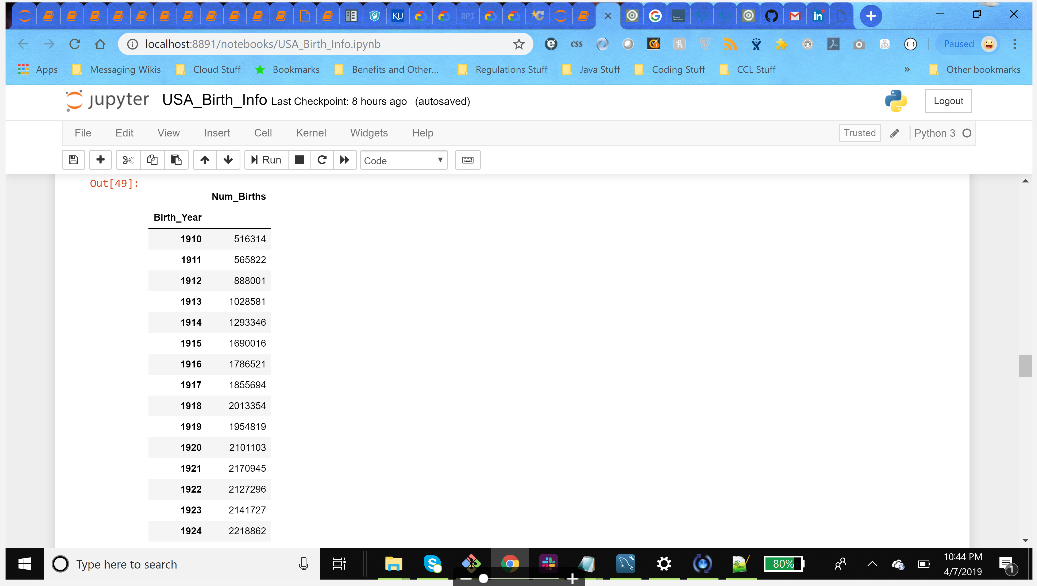
**1. Wanted to see compare the State Name, Year, and number of births associated with that state and year.**

**2. Wanted to compare the state and the total number of births from this timeframe**

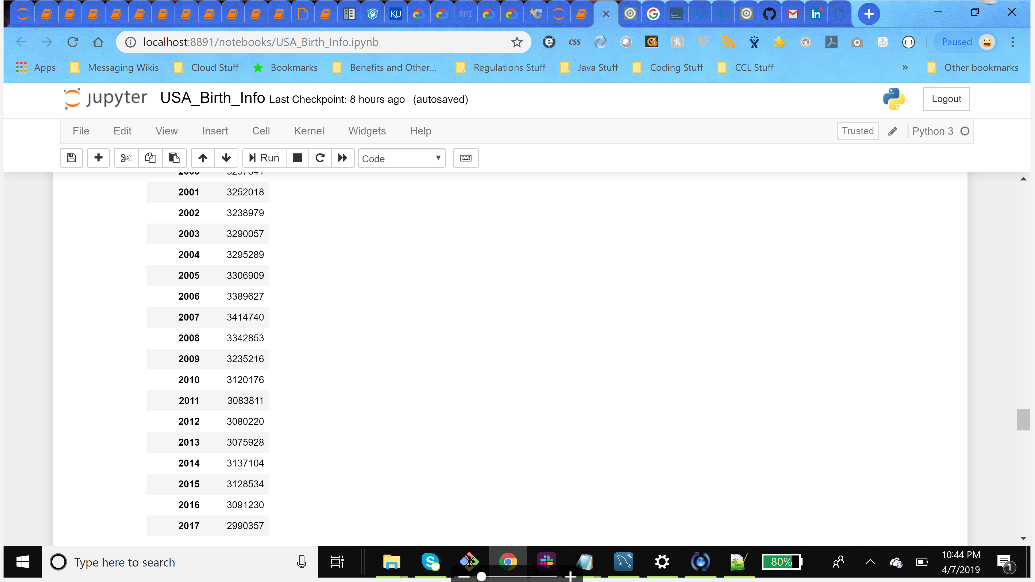
**3.Wanted to see the total number of births in each state given depending on the year.**

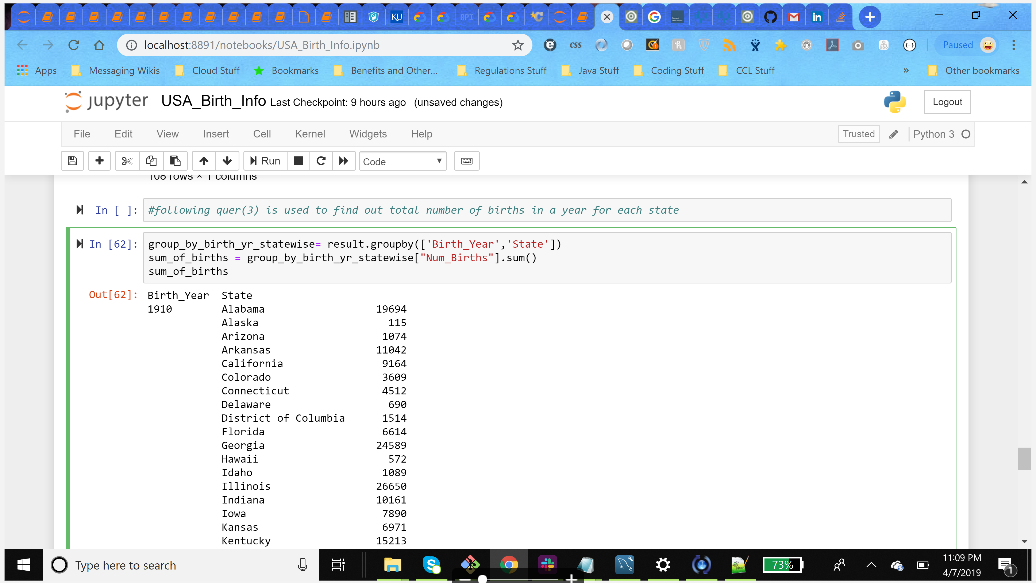
****



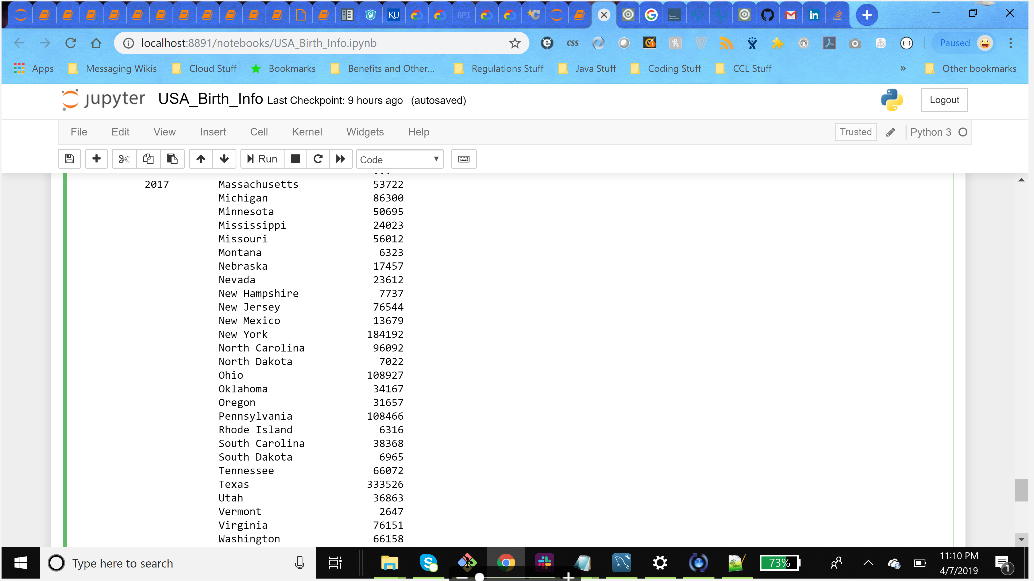


Output for Query 2 (above)





query 3 total births in a particular year for each state (above)



output of query (3) for year 2017 (above)