John McGrann
XQ75295

Global Warming Time Series Data Exploration

Goal:

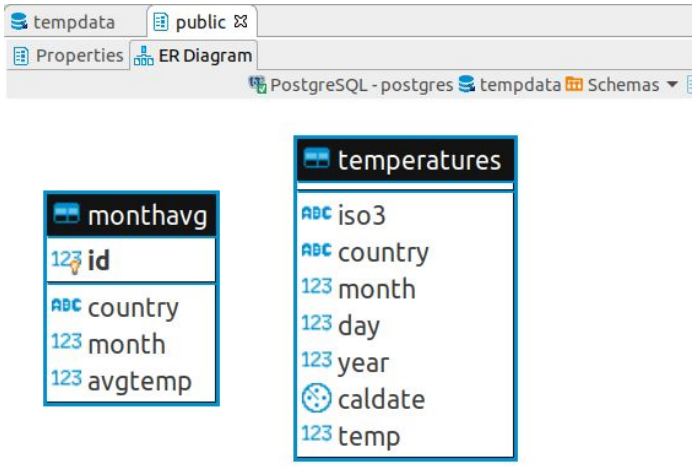
The overall goal of this project was to create a time series data exploration tool that allows any user to explore factors contributing to global warming data and be able to compare data between different countries. This project also served as my Data Visualization project for CMSC 436/636, and only one of the features fully utilized the skills obtained in this class, and will be focused on much more. The particular feature that I will be discussing is the visualization created regarding each countries average monthly temperatures. I had obtained temperature data for each country from 1901-2016 showing the average monthly temperature in that country at that year and month. The temperature data gathered was obtained from the Climate Change Knowledge Portal (CCKP) created by the World Bank Group. It was generated from thousands of weather stations worldwide which are collecting temperature and rainfall data [1]. From this data, I wanted to create a smaller table of the overall monthly average temperatures for each country over the entire timespan, so that way the user could compare any of the monthly temperatures for a particular year in a country to that countries overall monthly temperatures in order to track how the climate has changed over time.

Software and Technologies Used:

For all organization and analysis of my databases, I used an Ubuntu 18.04.03 VirtualBox Machine, with PostgreSQL version 12.0 running on it. All the visualizations were then created on Tableau Public using my Windows 10 laptop after the modified data sets were transferred back to my main machine. In order to perform the calculations to determine the overall monthly

averages for each country, I implemented a script in python utilizing psycopg2 in order to connect to my PostgreSQL database on my Ubuntu virtual machine.

ER Diagram:



Because of the way the data was organized in the initial database, I was unable to form a effective foreign key between the two tables after creation. Since each country appeared multiple times for each month on the monthavg table which had the overall monthly averages for each country, and since the months repeated for each country for every year the data was available in the temperatures table, it was not possible to form a unique key on either table, and therefore not possible to form a foreign key between the two.

Implementation:

I first uploaded the temperatures table from the original source data. I initially added a unique ID to each row, but later removed that value since it was of no use. In order to compute the overall average monthly temperatures for each country, I implemented a python script utilizing psycopg2 in order to connect to my PostgreSQL database. I then had it implement search queries in order to compute the overall average temperatures for each month in each

particular country utilizing multiple cursors. I then had the script create new entries in the monthavg table in order to store the resulting overall average temperatures per month for each country. The screenshots of the table being created and the applicable code are included below. I would include the resulting tables as well, but they would be far too large to fit in the documentation.

```
tempdata=# create table temperatures (
id serial NOT NULL,
iso3 character varying(50),
country character varying(50),
month int, day int, year int,
caldate date,
temp float(8),
CONSTRAINT temperatures_pkey PRIMARY KEY (id)
)
;
CREATE TABLE
tempdata=# COPY temperatures(iso3,country,month,day,year,caldate,temp) FROM '/home/jack/Documents
/DataVis/TempData.csv' DELIMITER ',' CSV HEADER;
COPY 272832
tempdata=# SELECT * FROM temperatures;
```

ComputeMonthAvg.py > ...

```python
import psycopg2

#Connects to db
con = psycopg2.connect(
            host = "localhost",
            database = "tempdata",
            user = "jack",
            password = "lozfiresword")

#Create Cursor
cur = con.cursor()
#Execute Query
cur.execute("select distinct(country) from temperatures;")
#Catch all results as countries
countries = cur.fetchall()

```

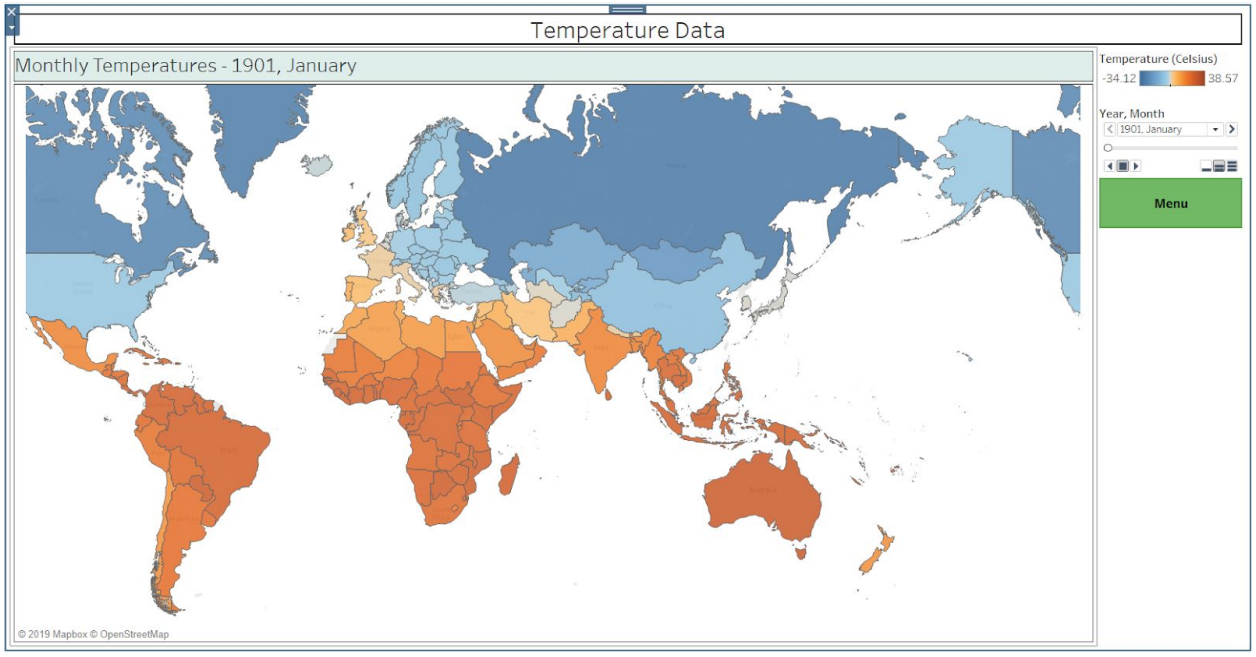
```
16
17  #Create second cursor for averages query
18  cur2 = con.cursor()
19  #Create third cursor for insertion into monthavg
20  cur3 = con.cursor()
21  avgmonth = 0.0
22  for i in countries:
23      for c in i:
24          for j in range(1,13):
25              cur2.execute("select avg(temp) from temperatures where country = %s and month = %s;",(c,j))
26              avgmonth = cur2.fetchall()
27              for avg in avgmonth:
28                  cur3.execute("insert into monthavg(country, month, avgtemp) values(%s,%s,%s);",(c,j,avg))
29
30  #Commit any changes
31  con.commit()
32
33  #Close Cursor
34  cur.close()
35  cur2.close()
36  cur3.close()
37  #Closes Connection
38  con.close()
```

After computing the overall average temperatures per month for each country, I then exported the database and added both tables to my visualization tool, Tableau Public, in order to create a visualization for the user to compare the average monthly temperatures per year in each country to their overall average monthly temperatures. I exported the monthavg table since I already had the original data file uploaded to Tableau with the original monthly averages.

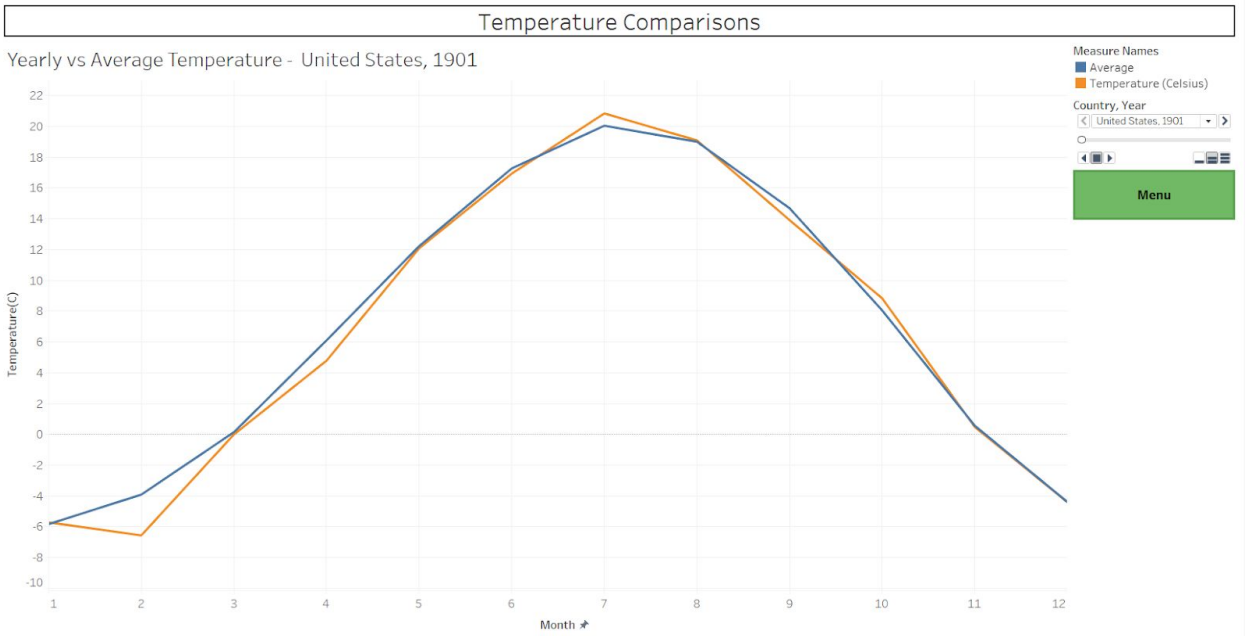
Exporting monthavg:

```
jack@jack-VirtualBox:~$ psql tempdata jack
psql (12.0 (Ubuntu 12.0-2.pgdg18.04+1), server 10.10 (Ubuntu 10.10-1.pgdg18.04+1))
Type "help" for help.

tempdata=# select * from monthavg;
 id | country | month | avgtemp
----+---------+-------+---------
(0 rows)

tempdata=# select * from monthavg;
tempdata=# \copy monthavg to 'TempMonthAvgs.csv' csv header;
COPY 2340
tempdata=#
```

Final Interface:



From this menu, users can select any year and month on the drop down menu to the side in order to directly compare the average temperature for that month and year for each country. Then, if the user wishes to compare any of the average monthly temperatures in a particular country for that year and month to the overall average temperature for that country during that month, the user need only click on the desired country with the appropriate year selected to take them to the next menu:



Once a country has been clicked by the user, the view will change to a direct comparison of the monthly averages in that country for that year vs the overall monthly averages for that country. From this menu, the user can change the year selected for the particular country, or select the menu option to return to the main menu and begin the exploration process over again. The user can mouse over any data point to display exact temperature values on each line to be compared.

Challenges Faced:

One of the challenges I faced while doing this project was the rapid progression and deadlines that I had to keep up with in my Data Visualizations class for this project, while also trying to apply the skills from Database Management Systems class. Many times I would have a certain requirement to meet for Data Visualizations and after creating a large portion of the program, I would learn new ways to make it more efficient in DBMS a couple weeks later. But in order to change the databases, I would have had to redo all of the work I had done for the

project in Data Visualizations at that point, and my partner was not fully on board with this. As a result I focused my application of DBMS skills to the visualization of the temperature data we had. Although I was still unsure of what was required from the DBMS project guidelines when I had to have a working Beta version ready to demo for Data Visualizations by October 22nd.

The only other challenges I had when implementing this project was not having enough dedicated time to work on this specific project. I took too many high level CMSC courses in one semester, including this course, 436/636 Data Vis, 441 Design and Analysis of Algorithms and 304 Social and Ethical Issues in Information Technology. While I only really struggled with the course material in 441, I ended up lacking any time at all during this semester with the sheer quantity of assignments due at a time, and the amount of hours each of them required. I also had to manage three different semester long projects at the same time, two of which were group projects that required managing schedules of more than one person, and as a result I ended up having to do most of the work on all of my group assignments because I did not have the time to wait for others to complete their portion of the work, or they simply did not help. That being said I am especially proud of the final product from this joint project given the time I had to work on it.

Future Improvements:

The main improvements I would add in order for the visualization to run more smoothly would be to create a larger database that encompasses all of the datasets used in the overall visualization, and perform and inner join based either on the name of the countries, or the ISO A3 which specify each country. Unfortunately, since the databases were drawn from so many different sources, many of the values in theses categories for the datasets varied slightly because

of different naming conventions, whitespace, and the different names and borders of particular countries over the years as well as the heavy time constraints faced, we were unable to perform such a join and as a result, some of the visualizations can take up to half a minute to load. Another improvement I would add upon a database which included all of our datasets would be an index on the country names or, more likely, the ISO A3 values. Since they are internationally standard they remain mostly consistent within the databases, and would improve the speed at which it is able to produce the categorical data for each nation.

Works Cited:

[1]Temperature Data:

Climate Change Knowledge Portal. (2018). *Metadata of the Climate Change Knowledge Portal*. *Metadata of the Climate Change Knowledge Portal*. World Bank Group.

World Bank Group. (2016). World Bank Climate Change Knowledge Portal. Retrieved October 3, 2019, from https://climateknowledgeportal.worldbank.org/.