**Performance Assessment: D211**

**A. Data Dashboards**

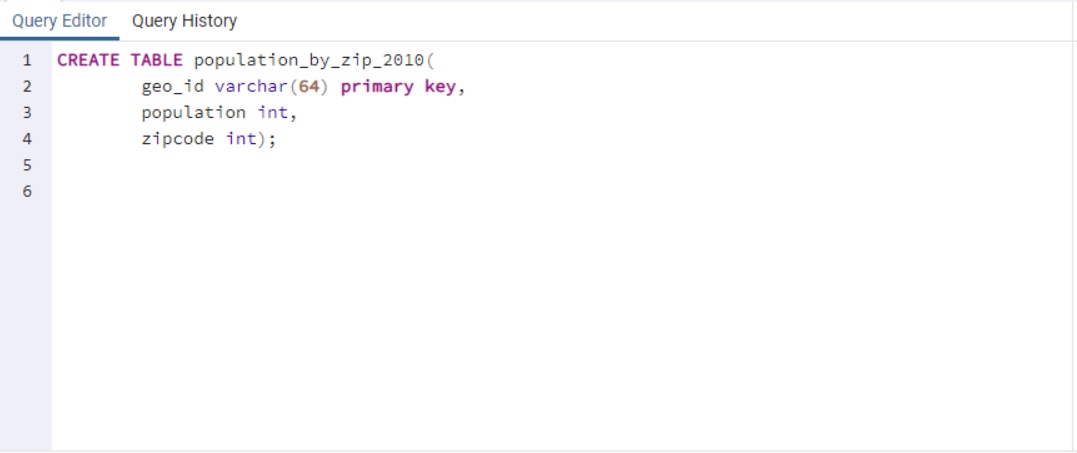
**1.** Both data sets used to create the dashboards have been attached alongside this assessment. Since one of the data sets was obtained through the use of PGAdmin4 within a virtual environment, an ERD of that data set has been used as a replacement file to showcase that data set, as seen below. This is the ERD of the medical dataset in PGAdmin4. The other data set has been submitted as a normal attachment.

A picture containing table

Description automatically generated

**2.** Listed below is a step-by-step guide as to how the dashboard was created in Tableau using the virtual environment.

* First, I created a table in PGAdmin4 called population\_by\_zip\_2010 and filled it with the data from the additional data set. Note: I didn’t use every variable from the data set, only the necessary ones. The SQL code to create the table and importation step are shown below:



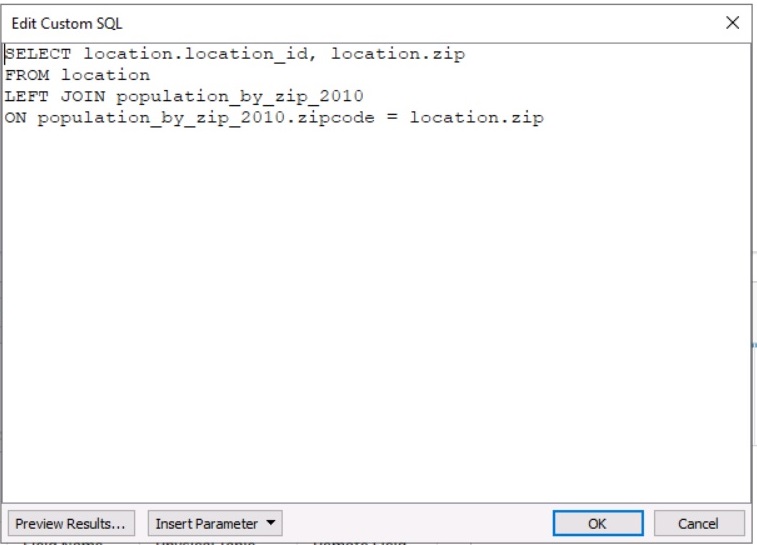
Graphical user interface, application, Teams

Description automatically generated

* When opening Tableau, I connected to the Postgres data server using the following information:
  + Server: localhost
  + Port: 5432
  + Database: medical\_data
  + Username: postgres
  + Password: Passw0rd!

This information was provided to us within the virtual environment, and I merely entered it as so to connect Tableau to the data source in PGAdmin4.

* Next, I created the following Custom SQL Query on the data source page where it was then joined to a second Custom SQL Query from the second image. The relationship was then set as “location\_id = location\_id (Custom SQL Query)”. The same inner join of the patient table to the location table could also have been done inside the first custom query by opening the query and then joining it to the patient table inside of Tableau directly but I chose this method to showcase that I knew how to include multiple custom SQL queries and to link them together in one data source.



Graphical user interface, text, application, email

Description automatically generated

* Renamed the “zip” field to “Zip Code” for readability.
* Then, I created each of these individual sheets:
  + ReAdmis by Zip: set the count of ReAdmis as the rows and the zip code as the columns.
  + ReAdmis by Zip with Avg Income at Poverty: repeated the same steps as above but changed the filter to AVG(Income) where I set the condition to at most $12,880 to match with the federal poverty level of 1 individual in 2021 (ASPE, n.d.).
  + Avg TotalCharge by Zip: set the rows as the AVG(TotalCharge) and set the columns as the zip code.
  + Avg Income by Zip: set the rows as the Avg(Income) and set the columns as the zip code.
* Afterwards, I created the following dashboards:
  + ReAdmis vs Poverty Line: added the ReAdmis by Zip sheet to the top and the ReAdmis by Zip with Avg Income at Poverty sheet to the bottom. This created a metric allowing the user to track the count of readmissions at all zip codes vs the zip codes where the average income is at or below the federal poverty level of $12,880 for 1 person (ASPE, n.d.).
  + Avg TotalCharge vs Avg Income: added the Avg TotalCharge by Zip sheet to the top of the dashboard and the Avg Income by Zip sheet to the bottom of the dashboard. This metric allows the user to track the average total charge for a patient in a zip code and then compare it to the average income in that zip code.

**3.** Once the dashboard has been downloaded to the user’s PC and then opened, first hit the **F7** key on your keyboard. This will open the dashboard in full-screen mode so the user can then use the arrow keys to scroll through the different dashboard. There are only two dashboards in this presentation. Both provide a visual comparison, with the first dashboard comparing the readmission count by zip code vs the readmission count by zip code where the average income is at or below the federal poverty line; and the second dashboard comparing the average total charge by zip code with the average income by zip code. The user can hover their mouse over each of the dashboards to directly view that individual metric. Note that there are no interactive features present in this dashboard, merely a simple and basic dashboard designed to show custom SQL queries being used to create a report.

**4.** All of the SQL code used to make these two dashboards, both in PGAdmin4 as well as directly inside of Tableau itself, have been shown through images that were attached during the listing of all of the steps followed.

**B. Report**

**1.** This dashboard was created to showcase how many people within a particular zip code have been readmitted to the hospital within one month of release. A metric was created using this information as well as readmissions by zip code where the average income is at or less than the 2021 federal poverty level, in order to determine if poorer patients are at an increased likelihood of readmittance. Secondly, a second dashboard was created to show the average total charge by zip code and compare it with the average income by zip code to see if poorer zip codes could potentially be getting overcharged and vice versa.

**2.** The business intelligence tool that I used was Tableau, which allows you to create custom SQL queries in order to create reports. As Tableau is the only BI software that I have experience using, it was the logical choice for this assessment since my experience using it both academically and professional made this assessment a lot easier to complete.

**3.** The first step that I used to clean the data was to drop a handful of columns from the additional data set as those entries would not be needed to do any joins or on any of the dashboards. The second and final step was to remove any duplicates from the geo\_id column as primary keys need to be unique, but population and zip code can be the same. This step was completed in R as shown below.

Graphical user interface, text, application, email

Description automatically generated

**4.** All of the steps used to create the dashboards were detailed in part **A2.**

**5.** The first dashboard showcases the readmission count by the patient’s zip code. It was then compared to the readmission count within zip codes where the average income is at or below the poverty level of $12,880 for 1 individual (ASPE, n.d.). This dashboard does not account for overall household income, nor does it use the poverty level of 2015 (when the data was recorded). This was to get a more accurate assessment of the readmissions based on poverty levels using today’s information. By using this information, executive leaders would be able to determine if there is a trend based on income and the likelihood of a patient being readmitted to the hospital within a month of release. On the dashboard you can clearly see the count of readmissions peak at a max of 6 with a handful of peaks at 5 whereas the count using the poverty line peaks at 4 with most of them being 1 or 2. This does not indicate a trend in the readmission count being attributed to poorer zip codes.

On the second dashboard, the average total charge by zip is uniformally distributed with no significant peaks. The average income by zip is also uniformally distributed. This would indicate that the average income of a zip code does not play a role in the average total charge.

**6.** Since the additional data set was simply the populations by zip code from the year 2010 it does not provide accurate population data since it is based on data over a decade old so some of the population data would obviously have changed. It is also fairly limited in that I can only analyze data tables where a location or zip code data point is relevant to a report or field.

**C. Sources**

“2021 Poverty Guidelines.” ASPE, https://aspe.hhs.gov/2021-poverty-guidelines.

Bureau, US Census. “US Population by ZIP Code.” Kaggle, 27 June 2017, https://www.kaggle.com/datasets/census/us-population-by-zip-code?select=population\_by\_zip\_2010.csv.