Part 2 Script

Review Research Question

* Show Customer Density (Loyal Customers) by Region in the United States per 1,000 total population
  + Break down regions by states
  + Calculate density based on total state or region population
  + Show density as “0.043/1000”, meaning 0.043 customers per 1,000 total population.
  + Allow dashboard user to filter map and details by Region

**Describe Technical Environment**

* PostgreSQL
* pgAdmin 4
* Tableau 2021.4 (student)

**Demonstrate Functionality of Dashboards**

Interactive Maps and Summaries: The dashboard features an interactive map that visualizes customer density by state and region. Users can click on different regions in the “density summary” to filter the map and details accordingly.

Dynamic Filtering: By selecting a region in the “density summary,” the map and detailed information about loyal customers in that region are displayed. This action filter enhances the user experience by providing focused insights.

Detailed Tooltips and Labels: Hovering over any data point on the map or summary provides detailed tooltips, including customer density, state name, population, and customer count. These labels are dynamically updated based on user interaction.

**Explain SQL used to support Dashboards**

* SQL scripts were used to create the customer churn tables
* SQL scripts were occasionally used to query data during the initial design of the dashboards

**Explain How Data Streams Prepared**

* The customer churn data contained location data for city, but the state used was just a 2-letter state code
* Needed to find external data of states where the states were broken down by major regions AND where the states were referenced by name and by the 2-letter state code

**Calculated Fields**: We created calculated fields in Tableau for customer density and other metrics to derive insights directly within the visualization tool. Describe Data Alignment

**Describe how data were aligned with other data points.**

Churn data

* The customer table is inner joined (or left joined) by a 1:many relationship with the location table.
* The relationship uses an attribute named “**location\_id**” which is in both tables (see next slide)

States

* The states table is inner joined (or left joined) by a 1:1 relationship with the population table.
* The relationship uses attribute “**State**” in the states table and attribute “**Name**” in the population table (see next slide)

**Explain How Databases Were Created**

* Using pgAdmin
* Create database
* Run SQL scripts to create tables
* Import data into tables
* Referential Integrity

**Explain Referential Integrity**

**To ensure referential integrity:**

Foreign Keys: We used foreign keys to enforce relationships between tables:

sql

Copy code

ALTER TABLE d211\_churn

ADD CONSTRAINT fk\_state

FOREIGN KEY (state) REFERENCES states(state);

Consistent Data Entry: We enforced consistent data entry practices through constraints and normalization to prevent orphaned records and ensure data accuracy.