1.  Describe the technical environment used to create the dashboards.

1. Part of the data is sourced from the Postgres churn database available in the Lab.

2.  Demonstrate the functionality of the dashboards.

1. Click through the Dashboard describing the visuals and the filters.

3.  Explain the SQL scripts used to support the creation of the dashboards.

1. The churn database’s customer table was joined with the location table.

4.  Explain how the data streams were prepared to support the analysis.

A. The ‘customer+’ table was created by connecting to the Postgres churn database. Then I joined the customer table with the location table using the location\_id from both tables. After that, I created the ‘CaseOrder’ column in the customer table to create a custom measure for analysis. Then, I imported the ‘Telco-Customer-Churn’ from the Kaggle website and joined it with the churn table using the ‘churn’ columns from both data sources.

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

A. The unified churn filter created in the last step was used to align the data points. It ensured that only the selected data points from both sources were shown when enabled in the executive dashboard. Additionally, index columns with unique constraints were created and then tables were joined using it.

6.  Demonstrate how the databases were created.

A. Churn was already available in the labs; however, I downloaded the telco-customer-churn from Kaggle. Later, the churn data were joined using the ‘churn’ column.

7.  Explain how referential integrity was enforced in the database.

A. Referential integrity was enforced between the customer and location table by ensuring the location.location\_id equals customer.location\_id, and there are no null values after the join. Additionally, the index columns were used from customer and telcocustomerchurn to ensure unique values from both columns were used.