ANDY BAYARDO  
D205 DATA ACQUISITION

PERFORMANCE ASSESSMENT  
13 APRIL 2023

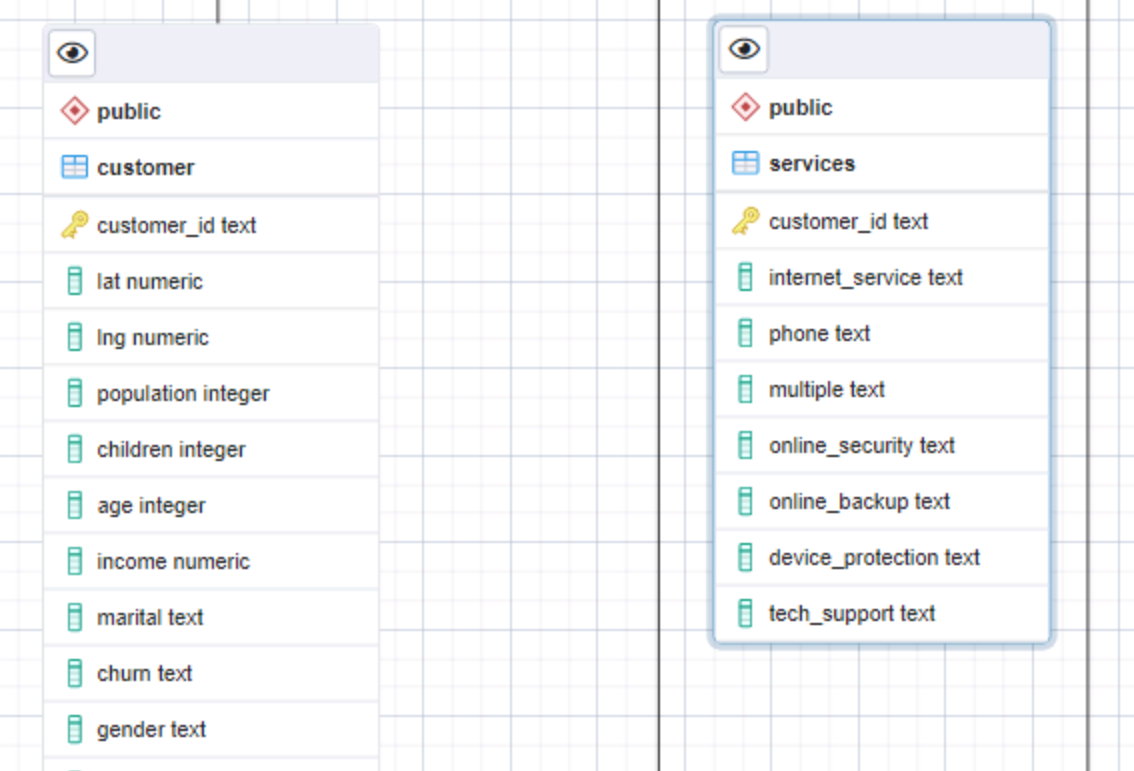
**A.**

The database for customer churn has tables which contain data on characteristics of each customer. The additional data in CSV form contains data in relation to the customer id number that shows which services the customers have, which include fiber optic, DSL, and none. My question aims to identify which cities have the highest number of customer churn for the past month; Of the total churn number per city, what is the percentage of fiber, DSL, and none in these areas? This query is to identify large numbers of churn and the service composition of those that left the service.

**1.**

The tables within the churn database titled customer and location from the original data have the unique customer ID and locational data to retrieve relevant results from the question. The add-on CSV data has the services data. This contains the type of service each customer had at the time of their active service the month prior.

**B.**



The first column identifying the customer ID will have a key constraint as a public key to uniquely identify each data row and to use for joining with other tables. All the columns will have an attribute constraint of a text data type. The “services” table does not have a direct relationship with the customer table. It is an extension of the customer table which would have a 1:1 relationship.

**1.**

CREATE TABLE services(

customer\_id text PRIMARY KEY,

internet\_service text,

phone text,

multiple text,

online\_security text,

online\_backup text,

deive\_protection text,

tech\_support text

);

**2.**

COPY services(

customer\_id,

internet\_service,

phone,

multiple,

online\_security,

online\_backup,

device\_backup,

tech\_support

)

FROM 'C:\LabFiles\Services.csv'

DELIMITER ','

CSV HEADER;

Reference 1 – Code compiled using information from reference.

**C.**

SELECT l.county AS county, l.state, COUNT(DISTINCT cm.customer\_id) as total\_customer\_last\_month,

COUNT(CASE WHEN cm.churn = 'Yes' THEN cm.customer\_id END ) as churn\_count\_last\_month,

ROUND(AVG(CASE WHEN s.internet\_service = 'Fiber Optic' AND cm.churn = 'Yes' THEN 1

WHEN s.internet\_service != 'Fiber Optic' AND cm.churn = 'Yes' THEN 0 END), 2)

as churn\_percent\_fiber,

ROUND(AVG(CASE WHEN s.internet\_service = 'DSL' AND cm.churn = 'Yes' THEN 1

WHEN s.internet\_service != 'DSL' AND cm.churn = 'Yes' THEN 0 END), 2)

as churn\_percent\_DSL,

ROUND(AVG(CASE WHEN s.internet\_service = 'None' AND cm.churn = 'Yes' THEN 1

WHEN s.internet\_service != 'None' AND cm.churn = 'Yes' THEN 0 END), 2)

as churn\_percent\_only\_phone

FROM customer as cm

INNER JOIN location as l

ON cm.location\_id = l.location\_id

INNER JOIN services as s

ON cm.customer\_id = s.customer\_id

GROUP BY county, l.state

ORDER by churn\_count\_last\_month DESC

**1.**   
  
**CSV file attached in submission.**

**D.**   
**From the results of the query on the available data that shows how many customers left the service provider the previous month the add-on data should be refreshed weekly. Each week new customers will be added to the service or existing customers may change their service type. To keep accurate results, it would be best to have this data updated weekly to ensure timely access for monthly or semi-monthly analysis which would be summarized in quarterly reports.**

**E.**

CREATE TABLE services(

customer\_id text PRIMARY KEY,

internet\_service text,

phone text,

multiple text,

online\_security text,

online\_backup text,

device\_protection text,

tech\_support text

);

COPY services(

customer\_id,

internet\_service,

phone,

multiple,

online\_security,

online\_backup,

device\_backup,

tech\_support

)

FROM 'C:\LabFiles\Services.csv'

DELIMITER ','

CSV HEADER;

**F.**

**Panopto Video Link Copied into Submission Page**

**https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=a44be8a4-09cf-4431-b9f3-afe300585390**

**G.**

**REFERENCES**

*Reference 1.*

*Import CSV File into PostgresSQL Table*. PostgresTutorial.com. Retrieved April 12, 23, from https://www.postgresqltutorial.com/postgresql-tutorial/import-csv-file-into-posgresql-table/