**D205: Data acquisition**

August 3, 2024

katherine marte

Student id: (redacted)

western GOVERNORS’ university

# **SECTION A: RESEARCH QUESTION**

My research question is: **What are the top three internet services used by senior citizens?**

# **SECTION A1: QUESTION JUSTIFICATION**

My question will be answered by querying columns within the provided “customer” table, as well as a newly created table that will be comprised of columns from our “services.csv” file. By using both the “customer” table and my newly created “marte\_table”, I will then be able to use the “Select”, “Join”, “Insert Into”, “Sort By”, “Group By”, “Order By”, and “Limit” SQL commands that will extract the most used internet service types. Once I have a list of the most used internet services, I will then be able to sort the ones that appear most frequently by age, focusing on our senior citizen population (defined as ages 64 or older). Once I have aggregated that information, I will be able to limit my final list to the top three internet services that have been used the most frequently by the surveyed senior citizen population.

# **SECTION A2: IDENTIFYING DATA**

To obtain the response to my question, I used data from both the original "customer" dataset and our add-on “services.csv” file, which I imported using PGAdmin4 (Elder, 2019). To find out the top three internet services used by senior citizens, I examined data from both sources. From the original dataset, I utilized the “customer” table, which included each customer’s unique ID (customer\_id) and age, helping me identify senior citizens (those aged 64 and older). I also used the “marte\_table” and the newly created “services\_new” table from the add-on CSV file. These tables contained the “customer\_id” that linked back to the “customer” table, the type of internet service (“internetservice”), and whether the customer used multiple services (“multiple”). By combining these data points through a join on “customer\_id” and filtering for customers aged 64 and older, I determined the most popular internet services among senior citizens. The data types for these columns were “VARCHAR(3) or VARCHAR(20)” for “customer\_id” and “internetservice”, “INT/INTEGER” for “age”, and “VARCHAR(3)” for “multiple”.

# **SECTION B: ENTITY RELATIONSHIP DIAGRAM (ERD) – PHYSICAL DATA MODEL**

A screenshot of a computer

Description automatically generated

# **SECTION B1: RELATIONSHIP DISCUSSION**

The “customer” table contains some of our most important details on the customers, such as the “customer\_id” (which is used as the primary key), “age”, and which internet service they chose from the options provided (“internetservice”). In creating my “marte\_table”, I essentially made a copy of my “services.csv” file, which helps to hold information about the various services, but this also helps create a duplicate within the schema which ultimately could confuse a user who might be unfamiliar with the dataset. The “customer” table has one-to-many relationships with both the “marte\_table” and the “top\_services” table, which are connected via “customer\_id”. The “top\_services” table includes columns for “customer\_id”, the type of internet service chosen by the customer (“internetservice”), and whether they use multiple services (“multiple”), as well as their age (“age”). However, due to the “marte\_table” being essentially a duplicate, the ERD is unable to clearly show these relationships, which can make it hard to see the foreign key constraints. To allow for better clarity and consistency with the data and across multiple tables, both the “top\_services” table and “marte\_table” should have a foreign key relationship with the “customer” table via my “customer\_id” column. In doing so, it would help link all the related data properly, maintain consistency across the tables, and ultimately ensure that the data is easy to understand for both casual and non-casual users.

# **SECTION B2: STATEMENT FOR THE ENTITY RELATIONSHIP DIAGRAM (ERD)**

When loading my services.csv to PGAdmin4, my running command was as follows: **Running command: --command" "\\copy public.marte\_table FROM “C:/LabFiles/Services.csv” DELIMITER"; CSV HEADER QUOTE “\" ESCAPE "",".** To create a table for the add-on Services CSV file based on my ERD (pgAdmin, n.d.), I could create a new table called “services\_new” which would be based on the schema of my previously created “marte\_table”. Below, is the SQL statement of the creation of the aforementioned “services\_new” table, in text format. The create table statement constructs our “services\_new” table and defines “customer\_id” as the primary key. This SQL statement also helps establish a foreign key constraint and helps link “services\_new” to the “customer” table.

**CREATE TABLE services\_new (**

**customer\_id VARCHAR(20) NOT NULL,**

**internetservice VARCHAR(20),**

**multiple VARCHAR(3),**

**phone VARCHAR(20),**

**onlinebackup VARCHAR(3),**

**onlinesecurity VARCHAR(3),**

**deviceprotection VARCHAR(3),**

**techsupport VARCHAR(3),**

**PRIMARY KEY (customer\_id),**

**FOREIGN KEY (customer\_id) REFERENCES customer(customer\_id)**

**);**

# **SECTION B3: LOADING CSV DATA**

When loading my services.csv to PGAdmin4, my running command was as follows: **Running command: --command" "\\copy public.marte\_table FROM “C:/LabFiles/Services.csv” DELIMITER"; CSV HEADER QUOTE “\" ESCAPE "",".** After researching how to import CSV files into PostgreSQL (PostgreSQL Tutorial Website, 2011), the following SQL statement in text format is what I would be using, assuming that I was loading my “services.csv” file into my newly created “services\_new” table.

**COPY services\_new (customer\_id, internetservice, multiple, phone, onlinebackup, onlinesecurity, deviceprotection, techsupport)**

**FROM “/path/to/your/services.csv”**

**DELIMITER “,”**

**CSV HEADER;**

# **SECTION C: SQL QUERY**

This section will be comprised of the SQL statements, in text format, that were used to answer the following question as previously mentioned in Section A: **What are the top three internet services used by senior citizens?**

* First, below is the SQL statement that helped create the “marte\_table”, which then had my services.csv imported into it. I was able to obtain this command by using the import tool because the import tool runs PSQL.exe using the copy command.

**CREATE TABLE marte\_table (**

**customer\_id VARCHAR,**

**internetservice VARCHAR,**

**multiple VARCHAR,**

**phone VARCHAR,**

**onlinebackup VARCHAR,**

**onlinesecurity VARCHAR,**

**deviceprotection VARCHAR,**

**techsupport VARCHAR**

**);**

**INSERT INTO marte\_table (customer\_id, internetservice, multiple)**

**SELECT c.customer\_id, s.internetservice, s.multiple**

**FROM customer c**

**JOIN marte\_table s ON c.customer\_id = s.customer\_id**

**WHERE c.age >= 64**

**GROUP BY c.customer\_id, s.internetservice, s.multiple**

**ORDER BY COUNT(s.internetservice) DESC**

**LIMIT 3;**

* Next, below is the SQL statement that allowed for the “customer” table to be joined with the “marte\_table” on “customer\_id”. The statement then filtered for senior citizens (defined as ages 64 and older) and grouped the rows by “customer\_id”, “internetservice”, and “multiple”. Lastly, I was able to order my results by the count of “internetservice” in descending order and limit my output results to the top three internet services.

**INSERT INTO marte\_table (customer\_id, internetservice, multiple)**

**SELECT c.customer\_id, s.internetservice, s.multiple**

**FROM customer c**

**JOIN marte\_table s ON c.customer\_id = s.customer\_id**

**WHERE c.age >= 64**

**GROUP BY c.customer\_id, s.internetservice, s.multiple**

**ORDER BY COUNT(s.internetservice) DESC**

**LIMIT 3;**

* Lastly, to ensure there was no confusion, as the “marte\_table” could be seen as containing duplicate information, I created the “top\_services” table and repeated the previous steps to clearly show my results and limit clutter.

**CREATE TABLE topservices AS**

**SELECT c.customer\_id, s.internetservice, s.multiple**

**FROM customer c**

**JOIN marte\_table s ON c.customer\_id = s.customer\_id**

**WHERE c.age >= 64**

**GROUP BY c.customer\_id, s.internetservice, s.multiple**

**ORDER BY COUNT(s.internetservice) DESC**

**LIMIT 3;**

# **SECTION C1: CSV FILES**

Below is a screenshot of my exported data file (top\_services.csv) that helped capture the results of my queries.

**A screenshot of a computer

Description automatically generated**

# **SECTION D: ADD-ON FILE TIME PERIOD**

I believe that for the data to remain relevant to the business activities, a quarterly update of the database would be the best course of action.

# **SECTION D1: EXPLANATION OF TIME PERIOD**

Choosing to update our add-on file (“marte\_table”) every quarter would make the most sense from a business perspective, as it would allow the refreshed data to remain relevant. This regular update would be crucial for the sales team and executives because it would allow them to understand how often our target customers are buying, keeping, and using the available internet services. I was able to do this by using the import tool, which runs PSQL.exe with the copy command.

# **SECTION E: PANOPTO VIDEO**

The following link is for my Panopto video (length 16:34): <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=61d7fb3a-efc9-4ce7-a027-b1c2001e682b>

# **SECTION F: WEB SOURCES**

1. Elder, J. (2019, May 27). \*Intro to PostgreSQL databases with PgAdmin for beginners - Full course\* [Video]. YouTube. <https://www.youtube.com/watch?v=Dd2ej-QKrWY>
2. pgAdmin. (n.d.). \*ERD tool\* [Documentation]. <https://www.pgadmin.org/docs/pgadmin4/development/erd_tool.html>
3. PostgreSQL Tutorial Website. (2011). \*Import CSV file into PostgreSQL table\*. <https://www.postgresqltutorial.com/postgresql-tutorial/import-csv-file-into-posgresql-table>

# **SECTION G: PROFESSIONAL COMMUNICATION**