D205 Data Acquisition

By: Kevin Sandoval

A.

What percentage of people aged 25-35 have an online security add-on?

A1.

In order to answer the question posed above, I will need to create a table that includes the data from the services.csv file. After that table is created I can then run SQL code to search both tables of data for the information necessary. I will also need the “customer\_id” columns of both tables as that will be the primary key I will use to join the tables and get the information I need

A2.

I’ll focus on the “onlinesecurity” column of the new table called “services”, and then a range of values from the “Age” column of the “customer” table. The “onlinesecurity” column is composed of strings of “Yes”/”No” responses to the question of if the customer has an online security add-on. The column “Age” consists of integer values that refers to the age of the given customer. The “customer\_id” column consists of a string that denotes a unique value for each customer.

B.

The generated ERD from the Churn database is shown below.

A screenshot of a computer

Description automatically generated

From the diagram, we can see that the data contained in the file is a 1:1 relationship. Each table has 1 key that is linked to the customer database and the rest of the data is not connected. Some of the relational constraints that are satisfied are the primary key connection, multi-column constraints, and the not null constraints. None of the columns have null values which is important to consider.

B1.

The existing table, “customer”, has a relationship with the new table, “services”, in that they both share the customer\_id column. Customer\_id is the primary key for both tables but it also acts as the foreign key for the services table, allowing the services table to connect with the customer table. Some issues in the worth noting that the saved image cropped out part of the line for the “payment” and “location” tables, but they do connect to the specified key in the “customer” table. One other thing to consider is that all values in the “services” table are some length of varchar per Dr. Sewell’s recommendation in video lecture “*D205 SQL Sunday Presentation.”* It’s worth noting as the rest of the data in the other tables are integer, numeric, and text, so it’s important to acknowledge potentially different data types. Lastly, each other table has a referenced column in the “customer” table. I considered creating another column in the “customer” table to include a “services\_id” to match the other tables, however since the services table already has customer\_id included, and the other values are all responses to questions, I decided it would be easiest to keep the primary key and foreign key the same and keep it as customer\_id. I thought it would make it easier to do it this way but it is worth considering that a “services\_id” key could be relevant.

B2.

The SQL code to create the new table for the add-on CSV is copied from pgAdmin and shown below. The drop table part of the code is for the Panopto video. It will drop the table and then recreate the table with no values stored in it. I can then import the data into the table and it can be easily shown in the Panopto video. I added every column that was in the services .csv file and then also designated that the customer\_id column will be the primary key. I also stated that it would act as the foreign key that can connect it to the customer table for the future. Code was helped by (W3Schools *SQL FOREIGN KEY Constraint).*

DROP TABLE services;

CREATE TABLE services (

      customer\_id varchar(10),

      internetservice varchar(20),

      phone varchar(3),

      multiple varchar(3),

      onlinesecurity varchar(3),

      onlinebackup varchar(3),

      deviceprotection varchar(3),

      techsupport varchar(3),

      PRIMARY KEY (customer\_id),

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

);

B3.

The SQL code that loads the data from one of the add-on CSV files into the newly created table is copied from the environment and shown below.

" "\\copy public.services (customer\_id, internetservice, phone, multiple, onlinesecurity, onlinebackup, deviceprotection, techsupport) FROM 'C:/LabFiles/Services.csv' DELIMITER ',' CSV QUOTE '\"' ESCAPE '''';""

C1.

The SQL code that answers my question in Part A is shown below. I have attached the CSV of the output for my question and I also decided to show a screenshot here of the output as well.

SELECT onlinesecurity, count(onlinesecurity) FROM services INNER JOIN customer ON services.customer\_id = customer.customer\_id WHERE customer.age > 24 AND customer.age < 36 GROUP BY onlinesecurity;

Output:



From the output of the code we can see that of the 1,586 people aged 25-35 in the data set, 1,020 peopled replied “No” to the onlinesecurity add-on question, and 566 people replied “Yes.” That shows 64.31% “No” response and 35.69% of “Yes” responses which answers the question in part A.

D1.

I believe the time period for how often the add-on file should be acquired is once per year. Since the database is looking at churned data, the responses that people give are important to track and model, but their responses would likely not change much from month-month. After a year people’s responses will undoubtedly change in some aspect, and thus we would want updated information that we can use to keep track of certain trends or patterns we discover.

E.

Panopto video provided in submission.

F.

Dr. Sewell, W (n.d). *D205 SQL Sunday Presentation.* Retrieved August 13th, 2024,From D205 Course Tips

Khan, M. Faisal. (n.d). *D205 Data Acquisition.* Joining Data in SQL. Retrieved August 5th, 2024 From <https://app.datacamp.com/learn/custom-tracks/custom-d205-data-acquisition>

W3Schools. (n.d). *SQL FOREIGN KEY Constraint.* Retrieved August 13th, 2024 From <https://www.w3schools.com/sql/sql_foreignkey.asp>