

TGM1 Task 1: Data Acquisition

A:RESEARCH QUESTION

The question I considered for this task is “Are customers who give high ratings for courteous exchanges and respectful responses less likely to churn compared to customers who give low ratings?” Customer responses are valuable for both the customer and the business. Customers may feel heard, and businesses may utilize this information to inform business decisions. This question specifically could be beneficial to the business when reviewing employee performances to ensure proper customer service. This question may also address a potential reason for churn rates, which could allow the business to make changes to possibly avoid such events.

A1:IDENTIFYING DATA

Considering that the research question at hand pertains to specific aspects to customer and employee communication, the data required can be found in the add-on Customer Responses CSV file within the following columns:

- Courteous Exchange
- Respectful Responses

From the original data set, it will be necessary to obtain the following data to process this question:

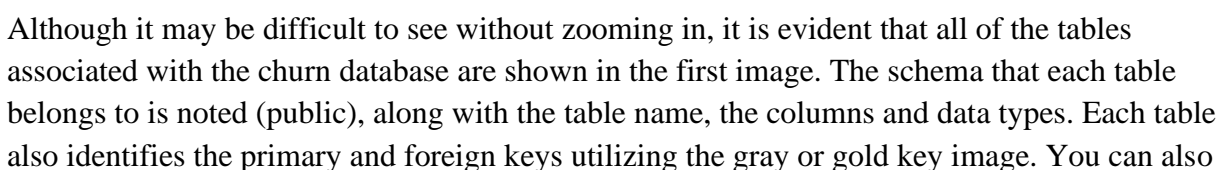
- Churn

There is one column that is available within the two tables, which is customer ID. This field will be used to join the original dataset to the add-on data. By joining these tables, we will be able to compare the customer responses in regards to courteous exchange and respectful responses, with churn rates for those same customers.

B:LOGICAL DATA MODEL

There are numerous columns found in the add-on CSV file, so I knew I needed to account for all the applicable data when importing the CSV file to the existing database. The data types found within the file were text, and the customer ID was the primary key. Data acquired needs to be applicable to the question, and as such an added constraint of “NOT NULL” was implemented for all columns.

There are also numerous tables and columns found in the churn database, and all of that applicable information along with the add-on CSV can be seen in the ERD below.



note how each table is connected. In the second image, you can see the table I created and how it is connected to the customer table within the database using Customer ID. You also see the primary key being customer ID as mentioned before, and all the applicable columns as well.

B1:CODE FOR THE PHYSICAL DATA MODEL

Using the GUI tool within pgAdmin4, I was able to create a table, which I named “responses_csv.” I was also able to add the appropriate columns, data type, and constraints. All of the data types were text, I confirmed that “NOT NULL” data would not be applicable, and added that the customer ID is the primary key. I also indicated that the Customer ID from the Customer table within the churn database was identified as the foreign key, to maintain relational integrity between the two tables. I executed the function and copied the script generated:

```
CREATE TABLE public.responses_csv
(
customer_id text COLLATE pg_catalog."default" NOT NULL,
timely_responses text COLLATE pg_catalog."default" NOT NULL,
timely_fixes text COLLATE pg_catalog."default" NOT NULL,
timely_replacements text COLLATE pg_catalog."default" NOT NULL,
reliability text COLLATE pg_catalog."default" NOT NULL,
options text COLLATE pg_catalog."default" NOT NULL,
respectful_response text COLLATE pg_catalog."default" NOT NULL,
courteous_exchange text COLLATE pg_catalog."default" NOT NULL,
evidence_of_active_listening text COLLATE pg_catalog."default" NOT NULL,
PRIMARY KEY (customer_id),
CONSTRAINT "customer_id_fkey" FOREIGN KEY (customer_id)
    REFERENCES public.customer (customer_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID
```

)

```
TABLESPACE pg_default;
```

```
ALTER TABLE public.responses_csv
```

```
OWNER to postgres;
```

B2:LOADING CSV DATA

Again using pgadmin4 processes, I was able to import the CSV file: Customer Responses and load the data from the file to the new table created by the SQL script above. The following script was copied after importing was complete:

```
--command " "\copy public.\"Responses_CSV\" (\"Customer_id\", \"TimelyResponses\",  
\"TimelyFixes\", \"TimelyReplacements\", \"Reliability\", \"Options\", \"RespectfulResponse\",  
\"CourteousExchange\", \"EvidenceofActiveListening\") FROM 'C:/LabFiles/SURVEY~1.CSV'  
DELIMITER ',' CSV HEADER QUOTE '\"' ESCAPE '\"';"
```

As you can see, the appropriate columns were identified, the location of the imported file was identified as well as delimiter and header columns accounted for. However, if I wanted to import the data myself directly from the desktop, I could have done so by executing the following:

```
COPY customer_responses_csv  
FROM 'C:\LabFiles\customer_responses.csv'  
DELIMITER ','  
CSV HEADER;
```

C:SQL QUERIES

A few queries were necessary to run in order to get the appropriate data needed to inform the question. First, I wanted to make sure that the data was imported correctly, so I did a simple select query to select the columns I would be using. This returned correctly, so I completed queries to alter the data type from text to integer in the courteous exchange column, and the respectful response column.

Altering the data type from text to integer allows me to later select specific results from those two columns. The specific results I was looking for were the 3 lowest ratings, and the 3 highest ratings.

I also wanted to join the customer table and the responses table I created, by utilizing the Customer ID column, as both tables would be using the same customer ID and this was a great way to join the tables together. Grouping and ordering the results in a specific order made the results more cohesive and organized, so this was a necessary step as well. The following SQL

statements are appropriate for the queries I explained above, which would provide information regarding the research question described earlier:

```
SELECT courteous_exchange, respectful_response  
FROM responses_csv  
GROUP BY courteous_exchange, respectful_response;
```

```
ALTER TABLE responses_csv  
ALTER COLUMN courteous_exchange TYPE integer USING courteous_exchange::integer;
```

```
ALTER TABLE responses_csv  
ALTER COLUMN respectful_response TYPE integer USING respectful_response::integer;
```

```
SELECT COUNT (c1.customer_id) AS total, c1.churn, cr.courteous_exchange,  
cr.respectful_response  
FROM customer AS c1  
INNER JOIN responses_csv AS cr ON c1.customer_id = cr.customer_id  
WHERE (courteous_exchange <= 3 OR courteous_exchange >= 6 )  
AND (respectful_response <= 3 OR respectful_response >= 6 )  
GROUP BY courteous_exchange, respectful_response, c1.churn  
ORDER BY courteous_exchange, respectful_response, c1.churn;
```

C1:CSV FILE(S)

The CSV file which provides the result from the query I created can be found attached appropriately to this assessment; I've also included a screenshot of the results below. Although this course is about acquiring the data, upon review of the CSV file - the different responses of customers and whether they have churned or not can be seen. Because the original add-on data had the highest result of 8, and lowest of 1 for customer responses in both fields, I accounted for responses <=3 and >=6. This was to get a range for the "highest" and "lowest" response. I was able to accomplish this by altering the data type from text to integer. Doing this rendered 33 rows, which would allow a complete review for the analyst to assess.

| total | churn | courteous_exchange | respectful_response |
|-------|-------|--------------------|---------------------|
| 15 | No | 1 | 1 |
| 6 | Yes | 1 | 1 |
| 71 | No | 1 | 2 |
| 19 | Yes | 1 | 2 |
| 51 | No | 1 | 3 |
| 23 | Yes | 1 | 3 |
| 1 | No | 1 | 6 |
| 62 | No | 2 | 1 |
| 27 | Yes | 2 | 1 |
| 287 | No | 2 | 2 |
| 89 | Yes | 2 | 2 |
| 371 | No | 2 | 3 |
| 132 | Yes | 2 | 3 |
| 3 | No | 2 | 6 |
| 1 | Yes | 2 | 6 |
| 41 | No | 3 | 1 |
| 20 | Yes | 3 | 1 |
| 447 | No | 3 | 2 |
| 116 | Yes | 3 | 2 |
| 1014 | No | 3 | 3 |
| 400 | Yes | 3 | 3 |
| 19 | No | 3 | 6 |
| 12 | Yes | 3 | 6 |
| 6 | No | 6 | 2 |
| 2 | Yes | 6 | 2 |
| 18 | No | 6 | 3 |
| 11 | Yes | 6 | 3 |
| 15 | No | 6 | 6 |
| 10 | Yes | 6 | 6 |
| 2 | Yes | 6 | 7 |
| 1 | No | 6 | 8 |
| 1 | No | 7 | 3 |
| 1 | Yes | 7 | 6 |

D:ADD-ON FILE

Many customers interact with multiple business associates every day, so it would be extremely repetitive to acquire and analyze this data on a daily basis, or even weekly basis. It may be best to keep the responses.csv file within the database and update monthly; but the review of this

question and associated data would better be completed on a quarterly basis. This could allow new-hire and tenured training sessions to reflect on the question at hand and alternative measures to ensure appropriate handling of customer service on a continuous basis.

E:SQL SCRIPT

As can be expected, new responses from customers will become available over time. As such the applicable data will need to be refreshed within the database. Because we previously altered the exchange and responses columns from data type text to integer, we will want to alter them back to ensure appropriate importing. Then delete the outdated data from the table, and import the CSV file upon update. The following SQL script would allow just that:

```
ALTER TABLE responses_csv  
ALTER COLUMN courteous_exchange TYPE text USING courteous_exchange::text;
```

```
ALTER TABLE responses_csv  
ALTER COLUMN respectful_response TYPE text USING respectful_response::text;
```

```
DELETE FROM responses_csv;
```

```
COPY customer_responses_csv  
FROM 'C : \LabFiles\customer_responses.csv'  
DELIMITER ','  
CSV HEADER;
```

F:PANOPTO VIDEO

When attempting to copy the URL for the Panopto video, I received the following message:

“Sessions in assignment folders can't be shared”

As such, the panopto video may be found under the “Master of Science, Data Analytics TGM1 | D205 (Student Creators) [assignments].” folder.

G:WEB SOURCES & H: Sources

I included both section G and E here, because no web or non-digital sources were used to implement or execute the results of this assessment.