­

**Performance Assessment for D205: Data Acquisition**

Christopher Fischer

College of Information Technology, Western Governors University

February 17, 2024

Performance Assessment for D205: Data Acquisition

This document contains the tasks and outputs required for the “TGM3 TASK 1: DATA ACQUISITION” All work is original to the author.

# A - The Relevant Research Question

Management is disheartened by the number of customers cancelling their service. Clearly, work needs to be done to better understand the needs of our customers. However, there are insufficient resources (funding and staffing) to perform detailed studies for all customers. The data analytics team has been provided with both the existing database of customers, as well as a CSV file containing 10,000 customer survey results. An analysis will be performed to segment our customers by age (in decades) and then rank the segments from highest to lowest total average survey score. Doing this will allow the company to target its resources and efforts on the segments which are the most dissatisfied with their service and our company.

## A1 – How Question Will be Answered

All customers who have responded to the recent survey will be grouped by their age in decades. For example, all tweens and teens, i.e., those customers aged 10-19, will be grouped together, etc. Next, the average total survey score will be computed for each age segment. The results will then be ranked from highest to lowest.

## A2 – What Data Will Be Used

To answer this question, these data will be studied:

|  |  |  |
| --- | --- | --- |
| Table | Source | Purpose |
| Customer | Existing database | Provides the demographic data necessary for segmentation |
| Survey Responses | CSV file | Provides the quantitative data on customer satisfaction with various aspects of our company services |

# B Entity Relationship Diagram

The following diagram shows the entities involved in the study described in section A above as well as any relationships between those entities.

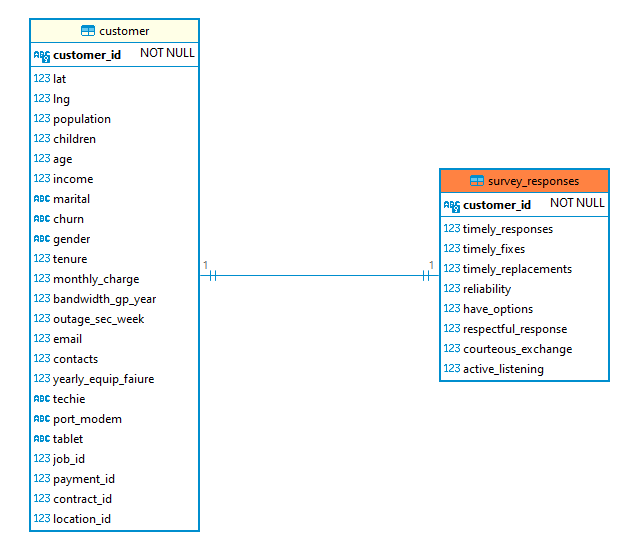


Figure 1 - Entity Relationship Diagram for Study

## B1 – Relationship Discussion

Notice that there is a one-to-one relationship (1:1) between the existing Customer table and the table of Survey Responses imported from the CSV file. This relationship was surfaced by examining the data in Customer and in Survey Responses and finding that each customer had one survey response. While this relationship matches the current data, in the future, it may be necessary to accommodate subsequent survey responses from customers which have previously responded. In that instance, the relationship will need to be changed to a one-to-many (1:M).

## B2 – SQL Statement to Create the Table

This section details the SQL statements used to create the Survey Responses table.

The first statement leverages PostgreSQL’s user-defined domain feature. The concept of a domain is a part of the relational model. A domain specifies the “set of acceptable values that a column is allowed to contain.” (Watt & Eng, 2014) Using domains facilities the reuse of domain-related business rules on multiple columns in multiple tables. Should those requirements evolve in the future, altering the domain will affect the change to all columns defined using the domain. While the domain concept has been a part of the ANSI SQL standard since 1992 (Melton & Simon, 1993), few major relational databases added support for domains until much later, if at all.[[1]](#footnote-2)

--------------------------------

-- Create a user-defined data type

--------------------------------

CREATE DOMAIN survey\_answer AS smallint

NOT NULL

CHECK (VALUE BETWEEN 1 AND 8);

The domain **survey\_answer** declares a datatype of small integer. In addition, it adds two value-related constraints which require the value to be non-null and to be an integer value between 1 and 8, inclusive.

The second statement uses the domain defined above to create the **survey\_responses** table. It contains the **customer\_id** of the customer which submitted each survey response, and one column for the answer to each survey question in a response. A primary key is created for **customer\_id** as there is only one survey response per customer. (See discussion on limitations to this arrangement in section B1 above.) Similarly, a foreign key from **customer\_id** to the column of the same name in the **customer** table is added to ensure the integrity of the submitted survey response.

--------------------------------

-- Create a table to hold the imported

-- survey response data

--------------------------------

CREATE TABLE survey\_responses (

customer\_id VARCHAR(128)

,timely\_responses survey\_answer

,timely\_fixes survey\_answer

,timely\_replacements survey\_answer

,reliability survey\_answer

,have\_options survey\_answer

,respectful\_response survey\_answer

,courteous\_exchange survey\_answer

,active\_listening survey\_answer

,CONSTRAINT pk\_survey\_responses

PRIMARY KEY (customer\_id)

,CONSTRAINT fk\_survey\_responses\_customer

FOREIGN KEY (customer\_id)

REFERENCES customer(customer\_id)

);

## B3 – SQL Statements to Import the CSV File and Verify the Contents

The following SQL statement is used to import the Survey Responses from the CSV file into the table created by the statements in B2 above:

--------------------------------

-- Import the survey response CSV file

-- into the table just created

--------------------------------

COPY survey\_responses (customer\_id

,timely\_responses

,timely\_fixes

,timely\_replacements

,reliability

,have\_options

,respectful\_response

,courteous\_exchange

,active\_listening)

FROM 'c:/LabFiles/SURVEY~1.CSV'

CSV HEADER QUOTE '"' ESCAPE '''';

Once the copy command has completed, the following two SQL statements are used to verify the number of rows imported and to visualize a sample of the imported data.

--------------------------------

-- Verify the number of rows imported

--------------------------------

SELECT COUNT(\*)

FROM survey\_responses;

--------------------------------

-- Examine a sample of the imported data

--------------------------------

SELECT \*

FROM survey\_responses

LIMIT 10;

# C – Answering the Research Question

The following SQL statement accesses the tables shown in section B above to answer the research question.

--------------------------------

-- Answer the research question

--------------------------------

*-- First, define a CTE to segment customers*

*-- by decade of age*

WITH **customer\_segments** AS (

SELECT customer\_id

,age - MOD(age,10) AS decade

FROM **customer**

),

*-- Next, define a CTE which computes the*

*-- average total score (rounded) for each segment*

**decade\_groups** AS (

SELECT c.decade AS decade

,COUNT(\*) AS customer\_count

,ROUND(AVG(timely\_responses +

timely\_fixes +

timely\_replacements +

reliability +

have\_options +

respectful\_response +

courteous\_exchange +

active\_listening)

,2) AS avg\_total\_score

FROM **customer\_segments** AS c

JOIN **survey\_responses** AS s

USING (customer\_id)

GROUP BY decade

)

*-- Last, rank the segments by total score,*

*-- highest best, and sort results in rank order*

SELECT RANK() OVER

(ORDER BY d.avg\_total\_score DESC) AS ranked

,d.decade

,d.avg\_total\_score

FROM **decade\_groups** AS d

ORDER BY ranked ASC;

## C1 – Results of Queries

The results of the query are shown here and attached for later reference. Notice that the customer segment which rates us most highly is customers between the ages of 70-79, while the segment which rates us the worst is customers between the ages of 20-29.

|  |  |  |
| --- | --- | --- |
| ranked | decade | avg\_total\_score |
| 1 | 70 | 28.12 |
| 2 | 80 | 28.08 |
| 3 | 50 | 27.99 |
| 4 | 30 | 27.98 |
| 5 | 40 | 27.97 |
| 6 | 10 | 27.94 |
| 7 | 60 | 27.89 |
| 8 | 20 | 27.82 |

Table 1 - Results of Research SQL



# D – Data Refresh Timeline

Since the data in the Survey Responses file comes from customer surveys, the file and related database table should be refreshed at least quarterly, or more often if a material number of new survey responses are received.

## D1 – Rationale for Refresh Recommendation

Since the purpose of this study is to identify the customer segment with which our company is most struggling to satisfy, it is imperative that we have a means to measure our progress towards improving the survey scores within that segment. Quarterly updates will give time for various initiatives to be instituted and begin to have an impact on the satisfaction our customers experience.

# E – Recorded Code Review

A recording of the code review presentation was uploaded with this submission. For quick reference, that video may be found here: <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=4d771c4b-e592-4e44-8c75-b11901345bcd>

# F –Works Cited

Melton, J., & Simon, A. (1993). *Understanding the New SQL: A Complete Guide.* San Mateo, CA: Morgan Kaufmann Publishers. Retrieved Feb 16, 2024, from <https://archive.org/details/understandingnew00melt/page/10/mode/2up>

Microsoft. (2023, May 23). *Transact-SQL statements*, SQL Server 2022. Retrieved Feb 16, 2024, from <https://learn.microsoft.com/en-us/sql/t-sql/statements/statements?view=sql-server-ver16>

Oracle Corporation. (2023, Dec). *SQL Language Reference*. Retrieved Feb 16, 2024, from [https://docs.oracle.com/en/database/oracle/oracle-database/23/sqlrf/create-domain.html#GUID-17D3A9C6-D993-4E94-BF6B-CACA56581F41](https://docs.oracle.com/en/database/oracle/oracle-database/23/sqlrf/create-domain.html%23GUID-17D3A9C6-D993-4E94-BF6B-CACA56581F41)

Watt, A., & Eng, N. (2014). *Database Design – 2nd Edition.* Victoria, B.C, Canada: BCcampus. Retrieved Feb 16, 2024, from <https://opentextbc.ca/dbdesign01/>

1. Oracle did not add the support for CREATE DOMAIN until version 23c. (Oracle Corporation, 2023) SQL Server still does not support the CREATE DOMAIN statement. (Microsoft, 2023) [↑](#footnote-ref-2)