1. Which age group is the most highly scored with the customer service referencing the average of respectful responses and active listening and when the customers had single or multiple services.
   1. I will group the customers by ages and whether they received a single or multiple services, then calculate the average of customer representative’s respectful responses and active listening.
   2. First, I created 2 tables which are called services and survey\_responses as following.

|  |  |
| --- | --- |
| **services** | **survey\_responses** |
| customer\_id (text) | customer\_id (text) |
| internet\_service (text) | timely\_responses (int) |
| phone (text) | timelyl\_fixes (int) |
| multiple (text) | timely\_replacements (int) |
| online\_security (text) | reliability (int) |
| online\_backup (text) | options (int) |
| device\_protection (text) | respectful\_responses (int) |
| tech\_support (text) | courteous\_exchange (int) |
|  | evidence\_of\_active\_listening (int) |

I imported the data for both tables from C:\LabFiles folder. I set the customer\_id in these tables as not null, primary keys, and foreign key to customer\_id column in the customer table .

Following table is the tables and column names I used for this assessment.

|  |  |  |
| --- | --- | --- |
| **customer** | **services** | **survey\_responses** |
| customer\_id | customer\_id | customer\_id |
| age | multiple | timely\_responses |
|  |  | evidence\_of\_active\_listening |

A screenshot of a computer

Description automatically generated

* 1. It is both 1-to-1 relationship between the customer and services tables and the customer and survey\_relationship tables. The services table has 1 record per customer for their type of service and the survey\_relationship table also has 1 record per customer survey. There can’t be more than 1 row in each table for each customer, otherwise, the customer\_id is duplicated and customer\_id column can’t be primary key anymore.

In the screenshot above, I was not able to add FK in the diagram. The PostgreSQL in the LOD did not have a FK feature. However, I designated the customer\_id column from both services and survey\_responses tables as FK using REFERENCES clause with the customer\_id column in the customer table.

* 1. **CREATE TABLE** public.services (  
     customer\_id text **NOT NULL REFERENCES** customer(customer\_id)**,**internet\_service text,  
     phone text,  
     multiple text ,  
     online\_security text ,  
     online\_backup text ,  
     device\_protection text ,  
     tech\_support text ,  
     **PRIMARY KEY** (customer\_id)  
     );  
       
     **ALTER TABLE** public.services  
      **OWNER** to postgres;  
       
     **CREATE TABLE** public.survey\_responses (  
     customer\_id text **NOT NULL REFERENCES** customer(customer\_id),  
     timely\_responses int,  
     timely\_fixes int,  
     timely\_replacements int,  
     reliablity int,  
     options int,  
     respectful\_responses int,  
     courteous\_exchange int,  
     evidence\_of\_active\_listening int,  
     **PRIMARY KEY** (customer\_id)  
     );  
       
     **ALTER TABLE** public.survey\_responses  
      **OWNER** to postgres;
  2. **COPY** survey\_responses  
     **FROM** ‘C:\LabFiles\Survey\_Responses.csv’ DELIMITER ‘,’ CSV HEADER;

1. **SELECT  
    CASE WHEN** c.age **BETWEEN** 18 **AND** 19 **THEN** ’18 – 19’

**WHEN** c.age **BETWEEN** 20 **AND** 29 **THEN** ’20 – 29’  
  **WHEN** c.age **BETWEEN** 30 **AND** 39 **THEN** ’30 – 39’  
  **WHEN** c.age **BETWEEN** 40 **AND** 49 **THEN** ’40 – 49’  
  **WHEN** c.age **BETWEEN** 50 **AND** 59 **THEN** ’50 – 59’  
  **WHEN** c.age **BETWEEN** 60 **AND** 69 **THEN** ’60 – 69’  
  **WHEN** c.age **BETWEEN** 70 **AND** 79 **THEN** ’70 – 79’  
 **ELSE** ’80 or Older’  
 **END AS** age\_bracket,  
 s.multiple **AS** multiple,  
 ROUND(**AVG**(sr.respectful\_responses + sr.evidence\_of\_active\_listening)/2, 2)   
 **AS** avg\_response\_listening   
**FROM** customer **AS** c  
**LEFT JOIN** services **AS** s  
**ON** c.customer\_id = s.customer\_id  
**LEFT JOIN** survey\_responses **AS** sr  
**ON** c.customer\_id = sr.customer\_id  
**GROUP BY** age\_bracket, multiple  
**ORDER BY** age\_bracket, multiple;

* 1. Submitted with ‘Saemi Ramirez (011926418) PA for D205 v3.csv’

1. The database will be updated once a day and the research query can be conducted every month.
   1. The database requires daily updates to ensure the availability of the latest consumer information, service details, and survey data, always facilitating comprehensive research with up-to-date information accessible. Conducting the research query monthly seems adequate to assess the customer satisfaction levels. This monthly report can help monitor the trend by tracking customer service satisfaction over time and identifying the trends and patterns in their feedback. From the customer’s feedback, the company can identify the areas for improvement. Also, the monthly reports allow the leadership to assess the impact of changes over time. The report can be compared with industry benchmarks or competitor’s performance to provide the context and insights into the organization’s standing within the market too.
2. <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=c27859f2-c461-425f-84ef-b14e0179b51c>
3. Acknowledge 2 web sources for any code used to support the application
   1. [11 Import CSV File To PostgreSQL (youtube.com)](https://www.youtube.com/watch?v=yDtgk_OLHUc)
   2. [PostgreSQL CASE (postgresqltutorial.com)](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-case/)