

Aidan Soares

012042436

D205 – Data Acquisition

Western Governors University

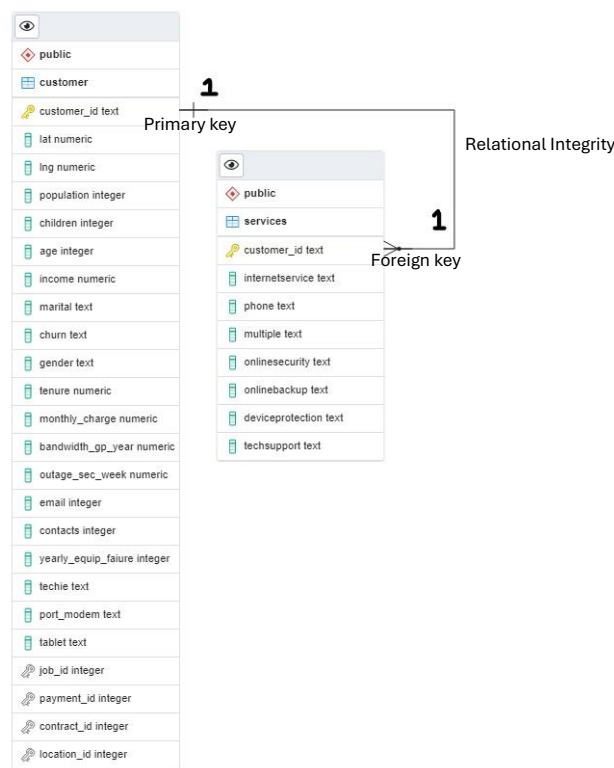
March 4, 2024

A. Question: How many individuals over the age of 50 do not have tech support with their service providers?

A1 Question justification: Utilizing pgAdmin4, I will be using SQL queries to count the total number of individuals who have not purchased the tech support package with their service providers, grouped by ‘yes’ or ‘no’. To isolate my desired dataset, I will use a clause in my query that selects only individuals who are over the age of 50 to achieve the answer to my question.

A2 Identifying data: The ‘customer’ table provided within the churn database is the table I will be using to isolate my target age range. These values are stored as integers in the age column and will be used to count the total number of individuals over the age of 50. Additionally, the primary key for this table, the customer_id column, is stored as text and will be necessary for joining the table created for the data stored in the ‘services’ csv. From the add-on csv file mentioned, I will be utilizing the ‘techsupport’ column. The information stored in this column is stored as text in the form of a ‘yes’ or ‘no’ format and will be used to sort the total count for my question. All the columns specified above will be used to solve my question, no additional tables will be required.

B. The Entity Relationship Diagram:



The ERD chart was created through PgAdmin4’s Generate ERD tool (PgAdmin Development Team, 2024).

B1 Relationships and constraints: From the ERD chart created above, the table ‘customer’ and the table ‘services’ have a 1-to-1 relationship. Their relationship is 1-to-1 as each customer within the ‘services’ table can only have one customer_id which corresponds to exactly one customer_id within the ‘customer’ table.

The customer_id serves as the primary key for the ‘customer’ table and serves as the foreign key for the ‘services’ table since no two individuals can share the same customer_id. This however creates a relational integrity constraint between the pair as the ‘services’ table refers to data in the ‘customer’ table through it’s foreign key. A row from the ‘customer’ table cannot be deleted, as the ‘services’ table would no longer be able to refer to the customer_id that was deleted, causing an error. By default, this database server will not allow for the violation of this constraint and will provide an error message unless the table is built to allow for the data to cascade; allowing for deletion of data in one table that will then delete from the other linked table (IBM, 2022).

B2 Creating add-on table:

```
CREATE TABLE public.services
(
    customer_id text NOT NULL,
    internetservice text NOT NULL,
    phone text NOT NULL,
    multiple text NOT NULL,
    onlinesecurity text NOT NULL,
    onlinebackup text NOT NULL,
    deviceprotection text NOT NULL,
    techsupport text NOT NULL,
    PRIMARY KEY (customer_id),
    CONSTRAINT customer_id_fk FOREIGN KEY (customer_id)
        REFERENCES public.customer (customer_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
);

ALTER TABLE public.services
    OWNER to postgres;
```

For the above statement, I re-used the titles for the columns found in the ‘services.csv’ file and matched the datatype structure to those found in the ‘customer’ table’s columns. I then specified the foreign key constraint on the customer_id column after designating it as a primary key and utilized the additional query structure to alter the table’s owner as per the other tables within the churn dataset.

B3 Importing the add-on table:

```
COPY public.services (  
    customer_id, internetsecurity, phone, multiple, onlinesecurity,  
    onlinebackup, deviceprotection, techsupport)  
    FROM 'C:/LabFiles/Services.csv'  
    DELIMITER ','  
    CSV HEADER ENCODING 'UTF8'  
    QUOTE '"'  
    ESCAPE '\\';  
FROM 'C:/LabFiles/Services.csv' DELIMITER ',' CSV HEADER QUOTE '\"' ESCAPE  
''';
```

C SQL Query:

```
SELECT services.techsupport, COUNT(customer.age) AS people_over_50  
    FROM customer INNER JOIN services  
    ON services.customer_id = customer.customer_id  
    WHERE customer.age >= 50  
    GROUP BY (services.techsupport);
```

The above is the SQL query used to identify the total number of individuals over the age of 50 who have not purchased the tech support package from their service provider.

C1 Query Results:

	techsupport text	people_over_50 bigint
1	No	3367
2	Yes	2095

This is the result from my SQL query conducted above. Within, the answer to my question is shown: 3367 individuals over the age of 50 have not purchased the tech support package from their service provider. I have included the exported csv file alongside my submission.

D Time period: For the purposes of general business activities, the information within the add-on csv file should be updated daily, maintaining up-to-date information on the services being provided to customers and for marketing purposes with the question posed above.

D1 Time period explanation: The information provided within the 'services.csv' add-on file is quite flexible and demonstrates potential to change at a moment's notice. If a customer were interested in changing their current service package, they could do it simply through a phone call. If this information were not updated on a daily basis, changes made by consumers may not be recognized until after bills are sent out, in which case the customers would be charged an inappropriate rate

for their current service package. Similarly, the question posed above seeks relevant data to target focus groups for marketing campaigns. Should an individual over 50 call to add tech support to their current service package, it would be important that said information be updated as soon as possible to prevent excessive advertising from being delivered, potentially leading to dissatisfied customers.

E Panopto: Panopto video recorded and provided through the webpage hosted by WGU. Link can be found in submission.

F Sources:

PgAdmin Development Team. (2024, March 3). *ERD Tool*. ERD Tool – pgAdmin 4 7.4 documentation. https://www.pgadmin.org/docs/pgadmin4/development/erd_tool.html

Referential integrity. IBM. (2022, October 21). <https://www.ibm.com/docs/en/informix-servers/14.10?topic=integrity-referential>