Shawn Wheeler

000364855

D205 – Data Acquisition PA

Table of Contents

**Section A - Question**3

A1 : Question Justification3

A2 : Identifying Data3

**Section B – Entity Relationship Diagram4**

B1 : Relationship Discussion5

B2 : Statement for the ERD5

B3: Loading CSV Data………………………………………………………………………………………………………………….6

**Section C – SQL Query6**

C1 : CSV Output from SQL Statements6

**Section D – Add-On File Time Period7**

D1 : Explanation of Time Period7

**Section E – Panopto Video8**

**Section F - Sources8**

**Section A – Question**

Does a diagnosis of high blood pressure relate to a patient providing lower survey scores?

**A1 – Question Justification**

The research question to be considered in this project will attempt to see if there may or may not be a difference in the survey response questions of patients based on their diagnosis of high blood pressure. The hypothesis to be tested is that those with high blood pressure will rate their interactions in surveys lower than their counterparts. This would be useful information for a business to consider as it attempts to increase patient satisfaction and return visits. If there is shown to be a correlation, one may consider introducing additional training to interact better with those afflicted with high blood pressure to bring them more in line with other patients by addressing their specific needs.

**A2 – Identifying Data**

In order to perform this analysis, the patient data will be utilized to create a group containing those diagnosed with high blood pressure and one other group inclusive of only those with no such diagnosis. This data is included in the patient table of the original database and listed as the patient\_id and hignblood, presumably a typo but used as is for the purposes of this project. Afterwards, as the comparison requires information from the survey responses, the fields for Respectful Response, Courteous Exchange, and Evidence of Active Listening will be acquired from the add-on table survey\_responses along with the customer\_id field which mirrors the patient\_id in the original patient table. This survey\_responses table will be joined with the patient table in the original database in order to allow for greater comparison.

From patient table :

A screenshot of a computer

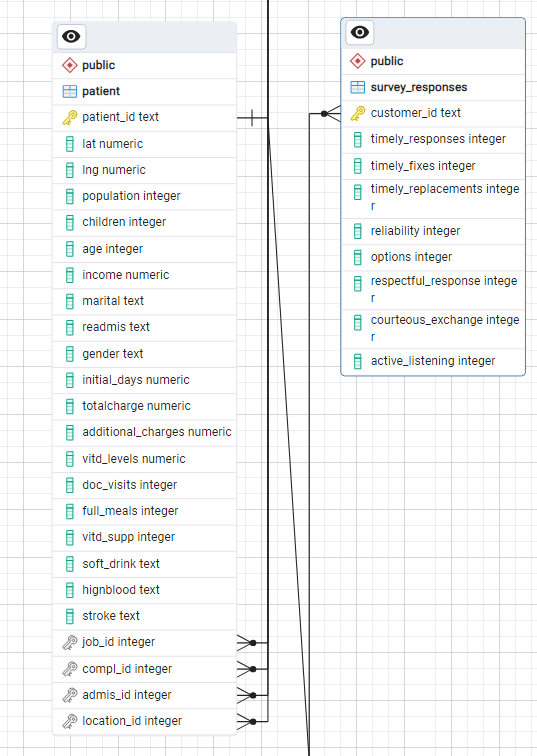
Description automatically generated

From survey\_responses table :

A screenshot of a computer

Description automatically generated

**Section B – Entity Relationship Diagram**

****

**B1 – Relationship Discussion**

After importing the table using the code below, the ERD produced above was created. In the imported table, survey\_responses, customer\_id was chosen to be the primary key. This was chosen due to the fact that it is the only column within survey\_responses that uniquely identifies each row. Updating this table with new information will not create a new record if one already exists but update the current one based on customer\_id.

A foreign key is essentially the same as a primary key but is within another table in a database. Customer\_id was also selected to be the foreign key in the survey\_responses table, as well, because it matches up with the primary key from the patient table containing the same unique values but simply with an alternate column title patient\_id. By doing this, a new customer can not be added to the survey\_responses table if there is not corresponding patient information in the patient table. This forcibly maintains integrity between the two columns and tables allows for usable data.

The ERD shows the relationship between the two tables as 1:m from the patient table to the survey\_responses table. This means that one and only one value from patient can be related to a survey\_responses record. Whereas many survey\_responses fields can be related to one record from the patient table.

**B2 – Statement for the ERD**

CREATE TABLE public.survey\_responses

(

customer\_id text NOT NULL,

timely\_responses integer,

timely\_fixes integer,

timely\_replacements integer,

reliability integer,

options integer,

respectful\_response integer,

courteous\_exchange integer,

active\_listening integer,

CONSTRAINT survey\_responses\_pkey PRIMARY KEY (customer\_id),

CONSTRAINT custom\_id FOREIGN KEY (customer\_id)

REFERENCES public.patient (patient\_id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.survey\_responses

OWNER to postgres;

Of note, this was provided by pgAdmin4 and would create issues with updating data in the future if the foreign key dependencies are not taken into consideration and other constraints due to being unable to duplicate primary keys, in this case the customer\_id field from survey\_responses.

**B3 – Loading CSV Data**

After the new table is created within the database containing the proper fields, the data from the .csv file needs to be populated. To do this, we will execute another SQL script shown below within the PSQL Tool of pgAdmin4. For this portion, PostgreSQL 16 Documentation was referenced (P.G.D.G.). The file path is dependent on local settings and must be altered accordingly.

COPY survey\_responses FROM 'C:\LabFiles\Survey\_Responses.csv' DELIMITER ',' CSV HEADER ;

**Section C – SQL Query**

The below query is used to join the newly added survey\_responses table with the patients table along with the relevant fields pertaining to this question. In this section, “PostgreSQL Primer for Busy People” was referenced (Neander).

SELECT pa.patient\_id, pa.hignblood, su.respectful\_response, su.courteous\_exchange, su.active\_listening

FROM patient AS pa

LEFT JOIN survey\_responses AS su

on pa.patient\_id = su.customer\_id

;

For the purposes of our question, we are going to use a SQL statement to provide our results as two fields: patients with high blood pressure and patients with a no diagnosis. Then, we are going to look at the count of surveys within those groups which responded with a 3, 4, or 5 in all three of respectful\_response, courteous\_exchange, and active\_listening.

**Section C1 – CSV Output from SQL Statements**

WITH hbpyes AS

(SELECT COUNT (pa.patient\_id) AS hbpyes

FROM patient AS pa

LEFT JOIN survey\_responses AS su

on pa.patient\_id = su.customer\_id

WHERE pa.hignblood = 'Yes'

AND su.respectful\_response >= 3

AND su.courteous\_exchange >= 3

AND su.active\_listening >= 3 )

,

hbpno AS

(SELECT COUNT (pa.patient\_id) AS hbpno

FROM patient AS pa

LEFT JOIN survey\_responses AS su

on pa.patient\_id = su.customer\_id

WHERE pa.hignblood = 'No'

AND su.respectful\_response >= 3

AND su.courteous\_exchange >= 3

AND su.active\_listening >= 3 )

SELECT \* FROM hbpyes, hbpno

Results:

A screenshot of a computer

Description automatically generated

The above screenshot shows the results from the previous SQL query as output into a .csv file which is also included separately with this project. 3151 patients with high blood pressure rated Respectful Response, Courteous Exchange, and Active Listening all with at least a 3 on the 1-5 scale versus 3384 without high blood pressure. These were out of 10,000 entries. Therefore, only 6,535 patients total rated those three survey responses at least a 3 which is something that a business may consider merits further analysis.

**Section D – Add-On File Time Period**

A recommendation of once per fiscal quarter seems logical as explained below.

**Section D1 – Explanation of Time Period**

Using just these simple commands, there seems to be some difference between the two groups analyzed. Whether this information is correlated or not would require greater depth and insight than required for this project. Using this data, however, if one was attempting to close the gap between clientele differences and satisfaction, it could make sense to update and re-analyze this data every fiscal quarter due to the fact that it requires a decently large dataset, time for potential training and implementation, along with myriad other factors. The survey\_responses table would be merely updated with data from new surveys for existing and / or new patients. With the current configuration, it would only allow for updating a singular survey for a singular patient due to potential conflicts with key constraints between tables. The patient table which is also used would theoretically be updated in real-time as clients are processed.

**Section E – Panopto**

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3e16026b-76ee-4915-860a-b140013e8047

**Section F – Sources**

Neander J. PostgreSQL Primer for Busy People · Zaiste Programming. zaiste.net. Accessed March 10, 2024. https://zaiste.net/posts/postgresql-primer-for-busy-people/

‌ The PostgreSQL Global Development Group. PostgreSQL Documentation. Published February 8, 2024. Accessed March 10, 2024. https://www.postgresql.org/docs/16/app-psql.html

‌