D205 Data Acquisition Performance Assessment

Professor Dr. William Sewell

Author: Eric Colwell

February 20, 2022

This report deals with the rubric for Western Governors University course D205 Data Acquisition and answers all the items in rubric order.

A) The question posed is how many customers have all the services offered filtered by gender and age groups? This question requires data from the original dataset and the add-on CSV file ‘services’. Answering this question will provide data to the company about customers that purchase all the available services offered. This data will be separated into age groups and gender. This will allow the company to make more informed marketing decisions.

a1) To answer the proposed research question, two tables will need to be utilized. The ‘customer’ and ‘services’ tables from the ‘churn’ database. From the ‘customer’ table the columns ‘age’ and ‘gender’ are required. From the ‘services’ table the columns ‘internetservice’, ‘phone’, ‘multiple’, ‘onlinesecurity’, ‘onlinebackup’, ‘deviceprotection’, and ‘techsupport’ are required.

B) Using Lucidchart I created this model. The model shows all the original database tables and the table that has been created. Each box in the diagram illustrates a separate table showing the table name, all the columns and their data types contained within it. The lines connecting the boxes have symbols on the ends signifying the relationship between the entities. The ‘customer’ table has a one-to-many relationship with all the original tables. The relationship with the added ‘services’ table is one-to-one. This is because both the ‘customer’ and ‘services’ table have the same number of rows that match exactly with the ‘customer\_id’. This diagram shows the primary and foreign key designations (Created in Lucidchart, n.d.).

Logical Data Model below:

Diagram

Description automatically generated

b1) The following SQL code will create an empty table with columns that match the columns in the add-on CSV file.

CREATE TABLE public.services

(

customer\_id VARCHAR(30)

internetservice VARCHAR(25)

phone VARCHAR(3)

multiple VARCHAR(3)

onlinesecurity VARCHAR(3)

onlinebackup VARCHAR(3)

deviceprotection VARCHAR(3)

techsupport VARCHAR(3)

CONSTRAINT services\_pkey PRIMARY KEY (customer\_id));

The next part of the SQL code alters the ‘customers’ table by adding a constraint for a foreign key that references the newly created ‘services’ table.

ALTER TABLE customer

ADD CONSTRAINT customer\_id\_fkey FOREIGN KEY(customer\_id)

REFERENCES services(customer\_id);

b2) The following code is used to load data from the add-on file into the ‘services’ table that we created earlier (Postgresql Tutorial, n.d.).

--command “ “\\copy public.services (customer\_id, internetservice, phone, multiple, onlinesecurity, onlinebackup, deviceprotection, techsupport)

FROM ‘C:/Users/LabUser/DOWNLO~1/services.csv’ DELIMITER ‘,’ CSV HEADER QUOTE ‘\” ESCAPE “;”

C) The following SQL query will generate the information needed to inform the research question. Using the ‘CASE WHEN’ statements I have separated the results into age groups. Using multiple ‘AND’ statements I get the information for only the customers that utilize all services offered. Next, if all the criteria were met, I used ‘THEN 1 ELSE 0’ to assign a value that would be added up using ‘SUM’ and supply the totals. To use information from both tables I used an ‘INNER JOIN’. At the bottom of the code, I used ‘GROUP BY’ to group the information by gender (Allison & Berkowitz, 2008).

SELECT c.gender AS “Gender”,

SUM(CASE WHEN c.age BETWEEN 18 AND 30

AND s.internetservice != ‘None’

AND s.phone = ‘Yes’

AND s.multiple = ‘Yes’

AND s.onlinesecurity = ‘Yes’

AND s.onlinebackup = ‘Yes’

AND s.deviceprotection = ‘Yes’

AND s.techsupport = ‘Yes’

THEN 1 ELSE 0 END) AS “18 to 30”,

SUM(CASE WHEN c.age BETWEEN 31 AND 50

AND s.internetservice != ‘None’

AND s.phone = ‘Yes’

AND s.multiple = ‘Yes’

AND s.onlinesecurity = ‘Yes’

AND s.onlinebackup = ‘Yes’

AND s.deviceprotection = ‘Yes’

AND s.techsupport = ‘Yes’

THEN 1 ELSE 0 END) AS “31 to 50”,

SUM(CASE WHEN c.age BETWEEN 51 AND 70

AND s.internetservice != ‘None’

AND s.phone = ‘Yes’

AND s.multiple = ‘Yes’

AND s.onlinesecurity = ‘Yes’

AND s.onlinebackup = ‘Yes’

AND s.deviceprotection = ‘Yes’

AND s.techsupport = ‘Yes’

THEN 1 ELSE 0 END) AS “51 to 70”,

SUM(CASE WHEN c.age > 70

AND s.internetservice != ‘None’

AND s.phone = ‘Yes’

AND s.multiple = ‘Yes’

AND s.onlinesecurity = ‘Yes’

AND s.onlinebackup = ‘Yes’

AND s.deviceprotection = ‘Yes’

AND s.techsupport = ‘Yes’

THEN 1 ELSE 0 END) AS “Over 70”

FROM customer AS c

INNER JOIN services AS s

ON c.customer\_id = s.customer\_id

GROUP BY c.gender;

C1) The required CSV file that is returned from this query will be uploaded separately from this file using the Attachments option.

D) Frequency of updating the add-on file could be weekly, monthly, or yearly depending on what the company is using it for. This could be determined by knowing what the information is being used for. If it is being used for a yearly marketing campaign, then I would recommend it be updated yearly to keep it useful and relevant.

E) The code below is the same as part B2 and is generated in the PGAdmin4 software. There is an import/export tool that is used to load the add-on data into the ‘services’ table. After successfully loading the data using the tool, the following script is generated.

--command “ “\\copy public.services (customer\_id, internetservice, phone, multiple, onlinesecurity, onlinebackup, deviceprotection, techsupport)

FROM ‘C:/Users/LabUser/DOWNLO~1/services.csv’ DELIMITER ‘,’ CSV HEADER QUOTE ‘\” ESCAPE “;”

F) The Panopto video has been uploaded to the Panopto drop box. The URL of the video is copied and pasted into the Links option.

Reference Page

Allison, C.E. & Berkowitz, N.A. (2008). *Sql for microsoft access*. *Creating table joins and*

*unions* (2nd ed., pp. 145-172). Wordware Publishing.

Created in Lucidchart. (n.d.). <https://www.lucidchart.com>

Postgresql Tutorial. (n.d.). *Import csv file into postgresql table.*

<https://www.postgresqltutorial.com/import-csv-file-into-table>