Steven Bennett

ID 003761827

D191 - Advanced Data Management

Performance Assessment

1. *Summarize****one****real-world written business report that can be created from the DVD Dataset from the “Labs on Demand Assessment Environment and DVD Database” attachment.*

The company would like to create a report that identifies 10% of customers that have the lowest amount of total transactions greater than 1. Using this report will allow promotions to be sent via email directly to this customer focus group to increase store visits and revenue.

1. *Identify the specific fields that will be included in the detailed table and the summary table of the report.*

Specific fields to be included in the detailed table: The fields customer\_id, store\_id, first\_name and last\_name as full\_name, email, and num\_rentals.

Specific fields to be included in the summary table: The fields first\_name and last\_name as full\_name, email, and num\_rentals.

1. *Describe the types of data fields used for the report*.

The types of data fields used for this report are a combination of integer, smallint, and varchar. The customer\_id and num\_rentals fields are integer type, store\_id is smallint type, and the full\_name and email fields are all var char type.

1. *Identify at least****two****specific tables from the given dataset that will provide the data necessary for the detailed table section and the summary table section of the report.*

The customers and rentals tables provide the necessary data for both tables to be constructed.

1. *Identify at least****one****field in the detailed table section that will require a custom transformation with a user-defined function and explain why it should be transformed (e.g., you might translate a field with a value of N to No and Y to Yes*).

In the detailed table, the first\_name and last\_name fields are concatenated (separated by a space) and transformed into full\_name. This will decrease the number of fields in the report and improve readability.

1. *Explain the different business uses of the detailed table section and the summary table section of the report*.

The summary table gives a quick view comprised of the full names of the 10% of the desired customer group, their email addresses, and number of rentals. This allows the Marketing department to use this information to send email promotions directly. This information will also be used as a scorecard for monitoring results of this targeted campaign as the reports will be updated monthly.

The detailed table provides the customer id, store id, customer full name, email address and number of DVD rentals. This report shows the desired group of customers’ customer id, store id, full name, email, and number of rentals. This information can be used to interpret which store has the greater/lesser share of least active customers, which could provide insights on store specific key performance metrics.

1. *Explain how frequently your report should be refreshed to remain relevant to stakeholders*.

The report should be refreshed monthly to allow the Marketing department to monitor progress and send monthly incentives.

B.  *Provide original code for function(s) in text format that perform the transformation(s) you identified in part A4*.

CREATE OR REPLACE FUNCTION Concat\_Names(first\_name varchar(45), last\_name varchar(45))

RETURNS VARCHAR(90)

LANGUAGE plpgsql

AS

$$

DECLARE full\_name VARCHAR(90);

BEGIN

SELECT first\_name || ' ' || last\_name INTO full\_name;

RETURN full\_name;

END;

$$

C.  *Provide original SQL code in a text format that creates the detailed and summary tables to hold your report table sections*.

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

-- CREATE DETAILED TABLE

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

DROP TABLE IF EXISTS Lowest\_10\_Detailed;

CREATE TABLE Lowest\_10\_Detailed(

customer\_id integer,

store\_id smallint,

full\_name varchar(90),

email varchar(50),

num\_rentals integer);

---------------------------------------------------------------------------------------------------------- CREATE SUMMARIZED TABLE

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

DROP TABLE IF EXISTS Lowest\_10\_Summary;

CREATE TABLE Lowest\_10\_Summary(

full\_name varchar(90),

email varchar(50),

num\_rentals integer);

D.  *Provide an original SQL query in a text format that will extract the raw data needed for the detailed section of your report from the source database*.

-- INSERT INTO DETAILED TABLE

INSERT INTO Lowest\_10\_Detailed

SELECT customer.customer\_id,

customer.store\_id,

Concat\_names(customer.first\_name, customer.last\_name) AS full\_name,

email,

COUNT(rental.customer\_id)

FROM customer

LEFT JOIN rental ON customer.customer\_id = rental.customer\_id

GROUP BY customer.customer\_id

ORDER BY COUNT(\*);

E.  *Provide original SQL code in a text format that creates a trigger on the detailed table of the report that will continually update the summary table as data is added to the detailed table*.

DROP TRIGGER IF EXISTS Update\_Summarized on Lowest\_10\_Detailed;

CREATE TRIGGER Update\_Summarized

AFTER INSERT ON Lowest\_10\_Detailed

FOR EACH ROW

EXECUTE PROCEDURE sum\_table\_update();

F.  *Provide an original stored procedure in a text format that can be used to refresh the data in both the detailed table and summary table. The procedure should clear the contents of the detailed table and summary table and perform the raw data extraction from part D*.

CREATE PROCEDURE refresh()

LANGUAGE SQL

AS $$

DELETE FROM Lowest\_10\_Detailed;

DELETE FROM Lowest\_10\_Summary;

INSERT INTO Lowest\_10\_Detailed

SELECT customer.customer\_id,

customer.store\_id,

CONCAT (customer.first\_name, ' ', customer.last\_name) AS full\_name,

customer.email,

COUNT(rental.customer\_id)

FROM customer

LEFT JOIN rental ON customer.customer\_id = rental.customer\_id

GROUP BY customer.customer\_id

ORDER BY COUNT(\*)

$$;

CALL refresh();

SELECT \* FROM Lowest\_10\_Summary

1. *Identify a relevant job scheduling tool that can be used to automate the stored procedure*.

PostgreSQL has no job scheduling capabilities, yet it is possible to automate this procedure. A job scheduling agent called pgAgent is available and allows stored procedures, SQL statements, and shell scripts to be executed and managed by PgAdmin 4. It is mentioned that pgAgent is not included in default installations but can be downloaded and installed by the user or admin.

G.  *Provide a Panopto video recording that includes the presenter and a vocalized demonstration of the functionality of the code used for the analysis.*

*Note: For instructions on how to access and use Panopto, use the "Panopto How-To Videos" web link provided below. To access Panopto's website, navigate to the web link titled "Panopto Access," and then choose to log in using the “WGU” option. If prompted, log in using your WGU student portal credentials, and then it will forward you to Panopto’s website.*

*To submit your recording, upload it to the Panopto drop box titled “Advanced Data Management D191 | D326 (Student Creators) [assignments].” Once the recording has been uploaded and processed in Panopto's system, retrieve the URL of the recording from Panopto and copy and paste it into the Links option. Upload the remaining task requirements using the Attachments option.*

Here is a link to a;dslkfja;dlkfja;sldkfja

H.  *Acknowledge all utilized sources, including any sources of third-party code, using in-text citations and references. If no sources are used, clearly declare that no sources were used to support your submission*.

No third-party code, in-text citations, or references were used in this submission.

I.  *Demonstrate professional communication in the content and presentation of your submission*.

Thank you for viewing my submission. If for any reason a resubmission is required, please give me as much feedback as possible. Have a great day.

***File Restrictions***

*File name may contain only letters, numbers, spaces, and these symbols: ! - \_ . \* ' ( )  
File size limit: 200 MB  
File types allowed: doc, docx, rtf, xls, xlsx, ppt, pptx, odt, pdf, txt, qt, mov, mpg, avi, mp3, wav, mp4, wma, flv, asf, mpeg, wmv, m4v, svg, tif, tiff, jpeg, jpg, gif, png, zip, rar, tar, 7z*