Jeremiah Lairson

ID: 009965674

ADVANCED DATA MANAGEMENT — D191

VDM2 — VDM2 TASK 1: DATA ANALYSIS

A) 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 DVD dataset can be used to track rental customers by their rental instances to better monitor and track inventory, share information with customers about video availability, assess rental and return dates for assigning fees, and assessing customer rental frequency to aid in targeted advertising and promotional offers.

A1. Identify the specific fields that will be included in the detailed table and the summary table of the report.

The detailed table should include the dates of rental and return along with the customer’s ID, full name, and email.

The summary table could include the customer ID, customer name, and the total number of rental instances.

A2. Describe the types of data fields used for the report.

The rental date data will use timestamps and the customer ID will be an integer. The customer name and its components will be written in variable length character strings, as will the email address.

A3. 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.

I will use data from the customer table and from the rental table to create a table more specifically geared toward the goals of the business.

A4. 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).

I will combine the customer first and last name in the detailed table just to eliminate a column and offer a more concise row.

A5. Explain the different business uses of the detailed table section and the summary table section of the report.

The detailed table can be used to monitor inventory, to extrapolate and share information about when videos will be back in, to calculate and assign late fees.

The summary table could be used to assess customer loyalty.

A6. Explain how frequently your report should be refreshed to remain relevant to stakeholders.

There is probably no need to refresh the report any more frequently than every 24 hours. Movie rentals are typically due by the end of a specific day but not at any specific time otherwise.

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

CREATE FUNCTION condense\_name(first\_name VARCHAR(45), last\_name VARCHAR(45))

RETURNS VARCHAR(90)

LANGUAGE plpgsql

AS $$

DECLARE full\_name VARCHAR(90);

BEGIN

SELECT CONCAT (last\_name, ', ', first\_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 TABLE customer\_rental (

customer\_id INT,

customer\_name VARCHAR(90),

customer\_email VARCHAR(50),

rental\_date TIMESTAMP,

return\_date TIMESTAMP

);

CREATE TABLE customer\_rental\_summary (

customer\_id INT,

customer\_name VARCHAR(90),

num\_rentals INT

);

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 customer\_rental (

customer\_id, customer\_name, customer\_email, rental\_date, return\_date

)

SELECT customer.customer\_id, condense\_name(customer.first\_name, customer.last\_name), customer.email AS customer\_email, rental.rental\_date, rental.return\_date

FROM customer

JOIN rental

ON customer.customer\_id = rental.customer\_id

ORDER BY customer\_id

LIMIT 1000;

INSERT INTO customer\_rental\_summary (

customer\_id, customer\_name, num\_rentals)

SELECT customer\_rental.customer\_id, customer\_rental.customer\_name,

COUNT (customer\_rental.customer\_id)

FROM customer\_rental

GROUP BY customer\_id, customer\_name

ORDER BY customer\_id;

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.

CREATE FUNCTION refresh\_summary\_on\_trigger()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

DECLARE full\_name VARCHAR(90);

BEGIN

DELETE FROM customer\_rental\_summary;

INSERT INTO customer\_rental\_summary (

customer\_id, customer\_name, num\_rentals)

SELECT customer\_rental.customer\_id, customer\_rental.customer\_name,

COUNT (customer\_rental.customer\_id)

FROM customer\_rental

GROUP BY customer\_id, customer\_name

ORDER BY customer\_id;

END;

$$;

CREATE TRIGGER refresh\_cr\_summary

AFTER INSERT ON customer\_rental

FOR EACH STATEMENT

EXECUTE PROCEDURE refresh\_summary\_on\_trigger();

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 OR REPLACE PROCEDURE refresh\_cr\_all

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM customer\_rental;

DELETE FROM customer\_rental\_summary;

INSERT INTO customer\_rental (

customer\_id, customer\_name, customer\_email, rental\_date, return\_date

)

SELECT customer.customer\_id, condense\_name(customer.first\_name, customer.last\_name), customer.email AS customer\_email, rental.rental\_date, rental.return\_date

FROM customer

JOIN rental

ON customer.customer\_id = rental.customer\_id

ORDER BY customer\_id

LIMIT 1000;

INSERT INTO customer\_rental\_summary (

customer\_id, customer\_name, num\_rentals)

SELECT customer\_rental.customer\_id, customer\_rental.customer\_name,

COUNT (customer\_rental.customer\_id)

FROM customer\_rental

GROUP BY customer\_id, customer\_name

ORDER BY customer\_id;

END;

$$;

F1. Identify a relevant job scheduling tool that can be used to automate the stored procedure.

I can use pgAgent to schedule the job on regular intervals.

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

[Panopto Hosted Video](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e95eba80-5db2-4020-ae3d-afe40132b660)

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 external sources were utilized.