TASK1: AUTOMATING DATA INTEGRATION

1. *Summarize****one****real-world business report that can be created from the attached Data Sets and Associated Dictionaries.*

A positive trend in sales of Movies-To-Go DVD Rentals has stakeholders pleased. However, they would like to see more growth. The marketing team has developed a loyalty program for customers to acquire rewards via a point system. The point rewards through a loyalty program will convince current customers to rent again and maintain relevance with both existing and future clientele. This will build the most revenue by utilizing a much cheaper marketing strategy. To kick off this promotion, we will reward our top ten renters each with 10,000 points multiplied by the number of their total rentals. The points earned can be used towards multiple free rentals. The customers will receive a confirmation of their rewarded points by email.

1. *Describe the data used for the report.*

This data uses customer first name, last name, email, number of rentals, and amount spent. The customer's information, such as name and email, individually identifies each customer and how to contact them. The number of rentals determines the number of DVDs rented from each customer to target the top ten renters. The amount spent calculates the revenue from rentals per customer.

1. *Identify****two****or more specific tables from the given dataset that will provide the data necessary for the detailed and the summary sections of the report.*

CUSTOMER table for the customer id, first name, last name, and email

RENTAL table for the rental id

PAYMENT table for payment id and amount

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

The detailed section includes eight fields: customer id, first name, last name, email, payment id, amount, rental id.

The summary section includes four fields: customer name, email, rentals, rental amount.

1. *Identify****one****field in the detailed section that will require a custom transformation and explain why it should be transformed. For example, you might translate a field with a value of ‘N’ to ‘No’ and ‘Y’ to ‘Yes’.*

I concatenated each customer's first and last name into a new attribute called ‘customer\_name’, creating the format of the first and last name with a space in between. Condensing the number of fields will improve readability.

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

The detailed section of the report provides a more detailed look at all customers and Movies-To-Go. This stored data helps to understand the various business trends of sales and implement marketing strategies, such as the points reward via the loyalty program, to increase company growth. This section provides the necessary data to construct the summary section.

The summary section of the report shows the top ten renters in an organized and formatted view. The query will only have to run once when refreshing the information monthly. Improving time management and more energy can be focused on the loyalty program.

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

The report should be refreshed monthly to stay current with both future and existing clientele and to ensure the customer rewarded is receiving the correct number of points.

*F1. Explain how the stored procedure can be run on a schedule to ensure data freshness.*

This procedure should be run at least monthly before awarding points and business meetings to ensure that the information is current and accurate for both the detailed and summary report. Data freshness correctly identifies the top ten renters for that month. A failure to refresh outdated information would mean a loss in customers and revenue for Movies-To-Go.

**Hello Nyssa Robinson, I am going to go through task 1 for the Advance Data Application PA.**

1. *Write a SQL code that creates the tables to hold your report sections.*

**So here I am going to create a new table named detailed**

CREATE TABLE detailed (

customer\_id INTEGER,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(90),

payment\_id INTEGER,

amount NUMERIC(5,2),

rental\_id INTEGER

);

**Now I can verify that the table is created and empty**

I verified the empty detailed table by the following code:

SELECT \* FROM detailed;

**Here I am going to create a new table named summary**

CREATE TABLE summary (

customer\_name VARCHAR(100),

email VARCHAR(90),

rentals SMALLINT,

rental\_amount NUMERIC(5,2)

);

**Now I can verify that the table is created and empty**

I verified the empty summary table by the following code:

SELECT \* FROM summary;

1. *Write a SQL query that will extract the raw data needed for the Detailed section of your report from the source database and verify the data’s accuracy.*

**For part C is going to fill the detailed table with all the information we need.**

**I inner join the customer, payment, and rental tables to extract information needed for the detailed table.**

INSERT INTO detailed (

payment\_id, amount, rental\_id, customer\_id, first\_name, last\_name, email)

SELECT p.payment\_id, p.amount, p.rental\_id,

c.customer\_id, c.first\_name, c.last\_name, c.email

FROM payment AS p

INNER JOIN customer AS c ON c.customer\_id = p.customer\_id

INNER JOIN rental AS r ON r.rental\_id = p.rental\_id

**I can now verify that the detailed table has been filled**

I verified the detailed table accuracy by the following code:

SELECT \* FROM detailed;

**I inner join the customer, payment, and rental tables to extract information needed for the detailed table.**

1. *Write code for function(s) that perform the transformation(s) you identified in part A4.*

**For part D is a function that will each customer's first and last name into a new attribute called ‘customer\_name’, creating the format of the first and last name with a space in between. Condensing the number of fields will improve readability**

DROP FUNCTION transform\_data();

CREATE OR REPLACE FUNCTION transform\_data()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

BEGIN

SELECT

customer\_id,

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

email,

COUNT (payment\_id) AS rentals,

SUM (amount) AS rental\_amount

FROM detailed;

END;

$$;

The function will concatenated each customer's first and last name into a new attribute called ‘customer\_name’, creating the format of the first and last name with a space in between. Condensing the number of fields will improve readability

**What we need from this Sequel is the top ten renters so that they can receive their point rewards. We also want to be able to see their email, number of rentals, rental amount So this will create a view of a list of the ten top renters.**

INSERT INTO summary (

SELECT

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

email,

COUNT (payment\_id) AS rentals,

SUM (amount) AS rental\_amount

FROM detailed

GROUP BY customer\_name, email

ORDER BY rentals DESC

LIMIT 10

);

I verified the summary table by the following code:

SELECT \* FROM summary;

**The summary table will now show the top ten renters data extracted from the detailed table.**

1. *Write a SQL code 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.*

**Part E is going to update the summary table whenever there is something new**

CREATE OR REPLACE FUNCTION update\_summary()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM summary;

INSERT INTO summary (

customer\_name, email, rentals, rental\_amount)

SELECT

customer\_id,

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

email,

COUNT (payment\_id) AS rentals,

SUM (amount) AS rental\_amount

FROM detailed

GROUP BY customer\_name, email

ORDER BY rentals DESC

LIMIT 10;

RETURN NEW;

END;

$$;

This function will update the summary table with data from the detailed table.

CREATE OR REPLACE TRIGGER update\_summary

AFTER INSERT ON detailed

FOR EACH STATEMENT

EXECUTE PROCEDURE transform\_data();

**The trigger above will allow that to happen. It’ll update the summary table with any information entered into the updated detailed table.**

1. *Create a stored procedure that can be used to refresh the data in both your detailed and summary tables. The procedure should clear the contents of the detailed and summary tables and perform the ETL load process from part C and include comments that identify how often the stored procedure should be executed.*

**This procedure should be run at least monthly before**

**awarding points and business meetings to ensure that the information is**

**current and accurate for both the detailed and summary report.**

**Data freshness correctly identifies the top ten renters for that month.**

CREATE OR REPLACE PROCEDURE refresh\_data()

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM summary; --this will clear the contents of the summary table.

DELETE FROM detailed; --this will clear the clear the contents of the detailed table.

INSERT INTO detailed (payment\_id, amount, rental\_id, customer\_id, first\_name, last\_name, email)

SELECT

p.payment\_id, p.amount, p.rental\_id,

c.customer\_id, c.first\_name, c.last\_name, c.email

FROM payment AS p

INNER JOIN customer AS c ON c.customer\_id = p.customer\_id

INNER JOIN rental AS r ON r.rental\_id = p.rental\_id;

--Here we are reentering the new data into the detailed table

END;

$$;

**AND THAT IS IT! THANK YOU**

1. *Provide a Panopto video recording that includes a demonstration of the functionality of the code used for the analysis and a summary of the programming environment.*

I did not use any external web sources to acquire data or segments of third-party code.