Peter Green

D191 Advanced Data Management

Performance Assessment

A. The attached data sets provided can be used to generate a plethora of different business reports. One real world business report that can be created from the attached data sets is a summary of which employees get customers to rent the greatest number of DVDs. Knowing which employees exceed expectations and outperform their peers is a vital piece of information for a business. Similarly, understanding which employees could use more training to improve their sales is also crucial. This report can be used to identify and reward the staff that is doing well, as well as recognize which employees could increase their number of rentals to customers. This will ultimately help the business to grow by guaranteeing the best employees are recognized and rewarded for their hard work, therefore ensuring the retention of the business's best employees.

A1: The data that would be used for this report would be the DVD rental information as well as the staff's information. The DVD rental information would be used to know what was rented, and which employee's id was responsible for it. The staff information would be used to identify the employee's full name and matching id that was responsible for a specific rental.

A2: The two tables I would need to create the detailed and summary portions of the report is the rental table and the staff table. Combining both of those tables will provide all the information needed to know which employees rent out the most DVDs.

A3: The detailed section of the report would contain the employee's name and relevant information, including the rental id, staff id, first name, last name, email, store id, and rental date. The summary section of the report would contain the employee's staff id, the employee's full name, and the total number of rentals.

A4: One field in the detailed section that would need a custom transformation are the first name and last name fields. When first extracted, the first and last names are split into two separate columns. To improve readability for the business and stakeholders, a transformation of these fields would need to be performed. Concatenating the first and last names together into one full name field would make this part of the report easier for stakeholders to know the full name of the employee they're looking at.

A5: There are many uses that the detailed and summary sections of the report can provide. For example, the detailed section of the report can be used to view which DVDs each employee rents out the most, what dates they rented those movies, and the id of the store where it happened. The summary section is great for a quick look at which employees rent out the most amount of DVD rentals. This is especially useful for employee performance reviews from management.

A6: This report should be refreshed every month so it can remain relevant to stakeholders. Monthly performance goals and results for employees is valuable information to management and stakeholders. This can be refreshed by a trained manager, or database admin that can call the refresh function to ensure the data is fresh on the schedule of every month at minimum or whenever the business needs the information. There can also be a function written that automatically refreshes the data every month without human intervention if needed.

B:

Creates the Detailed Table:

```
CREATE TABLE detailed(
    rental_id INT PRIMARY KEY,
    staff_id INT,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    store_id INT,
    rental_date TIMESTAMP
);
```

Creates the Summary Table:

```
CREATE TABLE summary(
    staff_id INT PRIMARY KEY,
    full_name VARCHAR(100),
    total_rentals INT
);
```

C:

The data's accuracy can be verified by doing a sanity check by asking questions to see if the information makes sense. For example, good questions to ask are if the variables in the output are the right type and quantity expected. Questioning if the data is presented in the right order and free of null values also helps to verify the accuracy of the data.

Inserts data into Detailed Table:

```
INSERT INTO detailed(
        rental_id,
        staff_id,
        first_name,
        last_name,
        email,
        store_id,
        rental_date
)
SELECT
        r.rental_id, s.staff_id, s.first_name, s.last_name,
        s.email, s.store_id, r.rental_date
FROM rental AS r
INNER JOIN staff AS s ON s.staff_id = r.staff_id;
```

Inserts data into Summary Table:

```
INSERT INTO summary(
SELECT staff_id,
        CONCAT(first_name, ' ', last_name) AS full_name,
        COUNT(staff_id)
```

```
FROM detailed
GROUP BY staff_id, full_name
HAVING COUNT(staff_id) > 1
ORDER BY COUNT(staff_id) DESC
LIMIT 100
);
```

D:

Code for functions that perform transformations:

```
CREATE FUNCTION summary_refresh_function()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN

DELETE FROM summary;

INSERT INTO summary(
SELECT staff_id,
        CONCAT(first_name, ' ', last_name) AS full_name,
        COUNT(staff_id)
FROM detailed
GROUP BY staff_id, full_name
HAVING COUNT(staff_id) > 1
ORDER BY COUNT(staff_id) DESC
LIMIT 100
);
```

```
RETURN NEW;
END; $$ ;
```

E:

Trigger on the Detailed Table

```
CREATE TRIGGER refresh_summary
AFTER INSERT ON detailed
FOR EACH STATEMENT
EXECUTE PROCEDURE summary_refresh_function();
```

F:

This procedure should be run every month at minimum or whenever the business needs the information. This is because monthly results regarding employee rental sales numbers can be used to identify excellent or struggling employees frequently, which helps to increase rental sales once managerial action is taken. This can be accomplished by a trained manager or database admin that can call the refresh function on schedule to ensure the data is fresh. They can use job scheduling tools such as Agent pgAgent or Extension pg_cron to accomplish this. There can also be code written that automatically refreshes the data on schedule every month to guarantee data freshness.

```
CREATE PROCEDURE refresh()
LANGUAGE plpgsql
AS $$
BEGIN

DELETE FROM detailed;

INSERT INTO detailed(
        rental_id,
```

```
        staff_id,
        first_name,
        last_name,
        email,
        store_id,
        rental_date
)
SELECT
        r.rental_id, s.staff_id, s.first_name, s.last_name, s.email, s.store_id, r.rental_date
FROM rental AS r
INNER JOIN staff AS s ON s.staff_id = r.staff_id;

DELETE FROM summary;

INSERT INTO summary(
SELECT staff_id,
        CONCAT(first_name, ' ', last_name) AS full_name,
        COUNT(staff_id)
FROM detailed
GROUP BY staff_id, full_name
HAVING COUNT(staff_id) > 1
ORDER BY COUNT(staff_id) DESC
LIMIT 100
);

END; $$ ;
```

G:

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=d0c1b0b9-fd83-4c0d-b309-af0a0024ade3

H:

For this assignment, I did not use any web sources besides the provided example database from my course instructor.

I:

There were no outside sources used for this assignment. Therefore, there are no in text citations or references listed.