Jonathan Bartolo
D326 Performance Assessment
7/1/2024

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.*

One useful business report that can be created from the DVD Dataset is a report on which actor's movies bring in the most revenue.

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

The following fields will be included in the *detailed* table:

Actor_id, first_name, last_name, film_id, title, rental_date, return_date, payment_amount

The following fields will be included in the *summary* table:

First_name, last_name, total_revenue

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

The types of data fields that will be used for the detailed table are as follows: Actor_id as an integer which identifies each actor, first_name as variable length string containing the first name of an actor, last_name as a variable length string containing the last name of an actor, film_id as an integer which identifies a film, title as a variable length string containing the title of the film, rental_date as a date identifying when the rental occurred, return_date as a date identifying when the rental was returned, payment_amount as a decimal identifying how much was paid

The types of data fields that will be used for the summary table are as follows: First_name as a variable length string containing the first name of the actor, last_name as a variable length string containing the last name of the actor, total_revenue as a decimal containing the total revenue generated by the actor's movies

3. 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 detailed table will use the film table for film_id and title, film_actor table for actor_id, actor table for first_name and last_name, rental table for rental_date and return_date, and payment table for payment_amount.

The summary table will source from the detailed table.

4. 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).*

Rental_date and return_date will be transformed from timestamp to date to enhance readability.

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

The *detailed* table section provides insights on individual transactions and can be used to determine which movies are rented more frequently so more stock should be ordered.

The *summary* table section provides a higher level of insight on which actor's movies generate the most revenue. This information can be used for corporate planning and trend analysis so that when actors have new movies being released, corporate can have an idea on how popular a rental it will be.

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

My report should be refreshed on a monthly basis. This allows for strategic planning and trend analysis

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

```
CREATE OR REPLACE FUNCTION ConvertTimestampToDate(ts TIMESTAMP)
RETURNS DATE AS $$
BEGIN
        RETURN DATE(ts);
END;
$$ LANGUAGE plpgsql;
```

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

```
CREATE TABLE Detailed (
        Actor_id INT,
        First_name VARCHAR(45),
        Last_name VARCHAR(45),
        Film_id INT,
        Title VARCHAR(255),
        Rental_date DATE,
        Return_date DATE,
        Payment_amount DEC(5,2)
);

CREATE TABLE Summary (
        First_name VARCHAR(45),
        Last_name VARCHAR(45),
        Total_revenue DEC(8,2)
);
```

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
```

```
SELECT a.actor_id, a.first_name, a.last_name, i.film_id, f.title,
ConvertTimestampToDate(r.rental_date), ConvertTimestampToDate(r.return_date),
p.amount
FROM rental r
JOIN payment p ON r.rental_id = p.rental_id
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film f ON i.film_id = f.film_id
JOIN film_actor fa ON f.film_id = fa.film_id
JOIN actor a ON fa.actor_id = a.actor_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 OR REPLACE FUNCTION summary_trigger()
RETURNS TRIGGER AS $$
BEGIN
	DELETE FROM summary;
	INSERT INTO summary SELECT first_name, last_name, SUM(payment_amount)
	FROM detailed
	GROUP BY first_name, last_name;
	RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER new_summary
AFTER INSERT
ON detailed
FOR EACH STATEMENT
EXECUTE PROCEDURE summary_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_tables()
AS $$
BEGIN
	DELETE FROM detailed;
	DELETE FROM summary;
	INSERT INTO Detailed
	SELECT a.actor_id, a.first_name, a.last_name, i.film_id, f.title,
	ConvertTimestampToDate(r.rental_date), ConvertTimestampToDate(r.return_date),
	p.amount
	FROM rental r
```

```
        JOIN payment p ON r.rental_id = p.rental_id
        JOIN inventory i ON r.inventory_id = i.inventory_id
        JOIN film f ON i.film_id = f.film_id
        JOIN film_actor fa ON f.film_id = fa.film_id
        JOIN actor a ON fa.actor_id = a.actor_id;
        RETURN;
END;
$$ LANGUAGE plpgsql;

CALL refresh_tables();
SELECT * FROM detailed;
SELECT * FROM summary;
```

1. *Identify a relevant job scheduling tool that can be used to automate the stored procedure.*
   pgAgent is a job scheduling tool that can be installed in pgAdmin and works with Postgres databases.

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

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=59cbe63d-fe0d-4977-8d95-b1a80151d117

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 sources were used.