Trevor England

D326 Performance Assessment

11/25/2024

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.*

In this project, I will be giving a Seasonal Trend Analysis report. One real world business question that would be helpful to know is “During the summer months of 2005, what were the top three genres of movies that customers rented the most?” Please note, that I have defined the term “summer months” as being June 1st – August 31st.

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 **summary** table:

**genre** VARCHAR(25)

**total\_rentals** INTEGER

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

**rental\_id** SERIAL PRIMARY KEY,

**rental\_date** TIMESTAMP

**customer\_id** INTEGER,

**customer\_name** VARCHAR(60),

**movie\_title** VARCHAR(255),

**genre** VARCHAR(25),

**film\_id** INTEGER,

**category\_id** INTEGER

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

The data types used for this report include VARCHAR for all string data and INTEGER for all number data.

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 **summary** table will be populated by joining rental, inventory, film, film\_category, and category together.

The **detailed** table will be populated by the joining rental, inventory, film, film\_category, category, and customer tables together.

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).*

The fields that I have created a user-defined function for are the “first\_name” and last\_name” fields. I chose to do a transformation on these fields for better readability of the data, as well as condensing the detailed table down by one column.   
  
The other user-defined function I created was on the name field in the category table. With the help of <https://www.geeksforgeeks.org/postgresql-array_agg-function/>, I was able to create this function that essentially parses the rental summary table and

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

The **summary** table provides a superficial insight as which movie genres are rented the most in the summer months. This information can be used to compare the data against previous years to determine if there is a trend of these genre types being popular during the summer. If there does seem to be a trend, it will help the store come up with a business plan for how many types of movies they will have available for rent for the upcoming summer season to boost their number of rentals.

The **detailed** table provides a granular look into the data from the summary table. Here, the business can see which movies within the top three genres were most frequently rented and try to determine correlations between the specific movies (such as movie duration, actors, etc.). This table also includes the name for every customer that rented a movie in the summer months that matches a top rented genre. This can be helpful as well to a business because they can use the data for personalized marketing to customers. For example, if you use this same table structure and modify the months for each season, you can then see which customers are most active during each season. This allows the business to use loyalty rewards program more efficiently by offering deals at calculated times to keep customers coming in.

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

Since this specific report is a seasonal trend analysis report, the report should be refreshed at the end of the season. For this specific report, the report would be refreshed on September 1st to ensure that all data from August is captured.   
  
Ideally there would be other types of reports that the business would use to determine sales strategies on a weekly basis. If for some reason there was not another report for stakeholders to view, this report would need to be refreshed on a weekly basis, perhaps at closing time on every Friday.

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

-- User defined function on first and last name columns

CREATE OR REPLACE FUNCTION get\_customer\_name(first\_name TEXT, last\_name TEXT)

RETURNS TEXT AS $$

BEGIN

IF first\_name IS NULL AND last\_name IS NULL THEN

RETURN 'N/A';

ELSEIF first\_name IS NULL THEN

RETURN TRIM(last\_name);

ELSEIF last\_name IS NULL THEN

RETURN TRIM(first\_name);

ELSE

RETURN TRIM(first\_name) || ' ' || TRIM(last\_name);

END IF;

END;

$$ LANGUAGE plpgsql

-- User Defined Function for dynamically populating detailed table with top three genres from the summary table

CREATE OR REPLACE FUNCTION get\_top\_three\_genres()

RETURNS VARCHAR[] AS $$

DECLARE

top\_three\_genres\_list VARCHAR[];

BEGIN

SELECT array\_agg(cat.name)

INTO top\_three\_genres\_list

FROM (

SELECT rs.genre, rs.total\_rentals

FROM rental\_summary AS rs

ORDER BY rs.total\_rentals DESC

LIMIT 3

) AS top\_three\_genres

INNER JOIN category AS cat ON top\_three\_genres.genre = cat.name;

RETURN top\_three\_genres\_list;

END;

$$ LANGUAGE plpgsql;

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

-- Create summary table

CREATE TABLE rental\_summary (

genre VARCHAR(25),

total\_rentals INTEGER

);  
  
-- Create detailed table

CREATE TABLE rental\_details(

rental\_id SERIAL PRIMARY KEY,

rental\_date TIMESTAMP,

customer\_id INTEGER,

customer\_name VARCHAR(60),

movie\_title VARCHAR(255),

genre VARCHAR(25),

film\_id INTEGER,

category\_id INTEGER,

FOREIGN KEY(customer\_id) REFERENCES customer(customer\_id),

FOREIGN KEY(film\_id) REFERENCES film(film\_id),

FOREIGN KEY(category\_id) REFERENCES category(category\_id)

);

-- Populate rental\_summary

INSERT INTO rental\_summary(genre, total\_rentals)

SELECT cat.name AS genre, COUNT(r.rental\_id) AS rental\_count

FROM rental AS r

INNER JOIN inventory AS i ON r.inventory\_id = i.inventory\_id

INNER JOIN film AS f ON i.film\_id = f.film\_id

INNER JOIN film\_category AS fc ON f.film\_id = fc.film\_id

INNER JOIN category AS cat ON fc.category\_id = cat.category\_id

WHERE r.rental\_date BETWEEN '2005-06-01' AND '2005-09-01'

GROUP BY cat.name

ORDER BY rental\_count DESC

LIMIT 3;

-- Populate rental\_details

INSERT INTO rental\_details

SELECT r.rental\_id, r.rental\_date, cust.customer\_id, get\_customer\_name(cust.first\_name, cust.last\_name) AS customer\_name, f.title AS movie\_title, cat.name AS genre, f.film\_id, cat.category\_id

FROM rental AS r

INNER JOIN inventory AS i ON r.inventory\_id = i.inventory\_id

INNER JOIN film AS f ON i.film\_id = f.film\_id

INNER JOIN film\_category AS fc ON f.film\_id = fc.film\_id

INNER JOIN category AS cat ON fc.category\_id = cat.category\_id

INNER JOIN customer as cust ON r.customer\_id = cust.customer\_id

WHERE r.rental\_date BETWEEN '2005-06-01' AND '2005-09-01'

AND cat.name IN (SELECT unnest(get\_top\_three\_genres()))

ORDER BY r.rental\_date DESC;

1. *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.*
2. *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.*
3. *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.*
4. *Identify a relevant job scheduling tool that can be used to automate the stored procedure.*
5. *Provide a Panopto video recording that includes the presenter and a vocalized demonstration of the functionality of the code used for the analysis.*
6. *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.*
7. *Demonstrate professional communication in the content and presentation of your submission.*