Silver Alcid
#011933973
08/28/2024
D326 – Task 1
Attempt 1

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.

A useful real-world report for this dataset is one that compares month-over-month data. Given that the dataset includes approximately four full months of rental activity, I opted to generate a month-over-month rental analysis for May and June 2005, categorized by staff members. The summary report focuses on the total number of rentals by month and by staff, while the detailed report breaks down rentals per film, also categorized by month and staff.

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:
staff_id (INT primary/foreign key)
month (VARCHAR(9) primary key)
total_rentals (INT)

The following fields will be included in the detailed table:
film_id (INT primary/foreign key)
month (VARCHAR(9) primary/foreign key)
staff_id (INT primary/foreign key)
film_title (VARCHAR(255))
count_rentals (INT)

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

Data types to be used include INT, and VARCHAR.

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 detail table will utilize data found in: film, rental, staff, and inventory tables within the dvdrental database.

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).**

   The field that must be transformed is the payment_date timestamp field. This will be transformed to a VARCHAR(9) field in order to increase understanding and clarity for stakeholders.

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

   The summary table serves as a quick reference to monitor the volume of rentals each staff member is handling, offering a snapshot of their productivity. This information can be crucial for ensuring that all staff members are adhering to their service level agreements (SLAs), if such agreements are in place. Additionally, it can provide an early indication of performance levels, although it's important to consider other factors. For instance, a shelf stocker may naturally process fewer rentals than a cashier, so performance assessments should be context-specific.

   The detailed table offers a deeper dive into rental patterns by identifying the most frequently rented DVDs. This data can be cross-referenced with inventory levels to verify that there are sufficient copies available to meet demand. Furthermore, this analysis can be extended over a longer period, such as a year, to identify DVDs that are underperforming in terms of rentals. Such insights could inform decisions to either remove these titles from inventory or reduce the number of copies in stock, thereby optimizing inventory management.

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

   These reports should be updated on a monthly basis, ideally at the beginning of each month, to reflect data from the two preceding months. This regular refresh ensures that the information remains relevant and allows stakeholders to make timely decisions based on the most recent trends. By consistently updating the reports, the business can maintain an accurate understanding of performance metrics, allowing for ongoing adjustments to strategies and operations. Regular updates also ensure that any corrective actions can be implemented promptly, minimizing the risk of prolonged inefficiencies or missed opportunities.

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

```
CREATE OR REPLACE FUNCTION format_month(payment_date TIMESTAMP)
RETURNS VARCHAR(9)
LANGUAGE plpgsql
AS
$$
DECLARE formatted_month VARCHAR(9);
BEGIN
formatted_month := to_char(payment_date, 'Month');
RETURN formatted_month;
END;
$$;
```

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

Summary Table:

```
CREATE TABLE rental_summary (
staff_id INT,
staff_name VARCHAR(45),
formatted_month VARCHAR(9),
total_rentals INT,
PRIMARY KEY(staff_id, formatted_month),
FOREIGN KEY(staff_id) REFERENCES staff (staff_id)
);
```

Detail Table:

```
CREATE TABLE rental_details (
film_id INT,
month VARCHAR(9),
staff_id INT,
staff_name VARCHAR(45),
film_title VARCHAR(255),
count_rentals INT,
PRIMARY KEY (film_id, month, staff_id),
FOREIGN KEY (film_id)
REFERENCES film (film_id),
FOREIGN KEY (staff_id)
REFERENCES staff (staff_id)
);
```

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 rental_details
SELECT
  i.film_id,
  format_month(r.rental_date),
  r.staff_id,
  s.last_name,
  f.title,
  COUNT(r.inventory_id) AS rental_count
FROM
  rental AS r
LEFT 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
  staff AS s ON r.staff_id = s.staff_id
WHERE
  r.rental_date BETWEEN '2005-05-01 00:00:00' AND '2005-06-30 23:59:59'
GROUP BY
  format_month(r.rental_date),
  r.staff_id,
  s.last_name,
  i.film_id,
  f.title
ORDER BY
  format_month(r.rental_date),
  r.staff_id,
  COUNT(r.inventory_id) DESC;
```

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.

```
-- Function to update the rental_summary table whenever new data is added to the
    rental_details table
CREATE OR REPLACE FUNCTION update_summary_trigger_function()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
  -- Clear the existing data in the summary table
  DELETE FROM rental_summary;

  -- Insert aggregated data into the summary table
  INSERT INTO rental_summary
  SELECT
    staff_id,
    staff_name,
    formatted_month,
    SUM(rental_count)
  FROM
    rental_details
  GROUP BY
    formatted_month,
    staff_id,
    staff_name
```

```
   ORDER BY
      formatted_month,
      staff_id;

   RETURN NEW;
END;
$$;

-- Trigger to automatically update the summary table after data is inserted into the detailed
      table
CREATE TRIGGER refresh_summary
AFTER INSERT
ON rental_details
FOR EACH STATEMENT
EXECUTE PROCEDURE update_summary_trigger_function();
```

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.

```
-- Stored procedure to refresh data in both the rental_details and rental_summary tables
CREATE OR REPLACE PROCEDURE refresh_report_tables()
LANGUAGE plpgsql
AS $$
BEGIN
   -- Clear the contents of the detailed table
   DELETE FROM rental_details;

   -- Populate the detailed table with fresh data
   INSERT INTO rental_details
   SELECT
      i.film_id,
      format_month(r.rental_date),
      r.staff_id,
      s.last_name,
        f.title,
        COUNT(r.inventory_id) AS rental_count
    FROM
        rental AS r
    LEFT 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
        staff AS s ON r.staff_id = s.staff_id
    WHERE
        r.rental_date BETWEEN '2005-05-01 00:00:00' AND '2005-06-30 23:59:59'
    GROUP BY
        format_month(r.rental_date),
        r.staff_id,
        s.last_name,
        i.film_id,
        f.title
    ORDER BY
        format_month(r.rental_date),
        r.staff_id,
        COUNT(r.inventory_id) DESC;

    -- The rental_summary table will be automatically cleared and refreshed by the trigger
    RETURN;
END;
$$;

-- Call the procedure to refresh both tables
CALL refresh_report_tables();

-- Verify the refreshed data
SELECT * FROM rental_summary;
SELECT * FROM rental_details;
```

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

We’re automating our data refresh process using **pgAgent**, a tool tailored for PostgreSQL. This job will run automatically on the first day of each month, executing the refresh_report_tables() procedure to update our detailed and summary tables with the latest data. This ensures our reports are always accurate and up-to-date without manual intervention, saving time and minimizing errors. We’ll monitor the job and make adjustments as needed to keep everything running smoothly.

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=919c836a-dbfd-4691-8c32-b1db010b7ef9

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.

Additional sources and references were not used in this project.