Task 1

Christopher McNey

Western Governors University

D326 - Advanced Data Management

10/19/2024

A: My business question is, "How many transactions were made per store, employee, and month?" The written business report I will create will show the number of films rented out per store, employee, and month. The first rental took place on February 14, 2007, and the most recent rental took place on May 14, 2007, so I will be looking at all the rentals within this period. The summary report will only contain information on how many transactions were completed per store per month. The detailed report will contain information on the stores, the months, and the employees. This report will benefit this DVD rental business by showing which stores are more profitable, which months are the most active, and which employees are performing the best. This will allow the business to make necessary changes in things such as marketing, training staff, and rewarding the stores and employees who perform the best.

A1: My **summary table** will include:

* store\_id (INT) PRIMARY KEY
* month (VARCHAR(9))
* total\_transactions (INT)

My **detailed table** will include:

* store\_id (INT) PRIMARY KEY, FOREIGN KEY
* staff\_id (INT) PRIMARY KEY
* month (VARCHAR(9))
* total\_transactions (INT)

A2: Data types used will include:

* INTEGER (a number; used for identifiable number like store\_id and staff\_id; used to show how many films were rented out)
* VARCHAR(9) (an alphanumerical string up to 9 characters; used to write out the months – September is the longest name at 9 letters)

A3: The detailed report table will pull from the ‘staff’ table, the ‘store’ table, and the ‘rental’ table.

The summary report table will pull from the ‘rental\_details’ table.

A4: The column 'rental\_date' in the 'rental' table will be transformed. The timestamp without timezone data will be transformed into VARCHAR(9) to replace the yyyy-mm-dd hh:mm:ss format into February, March, April, and May, respectively. This will be done for better readability (ex. "250 transactions in March" vs. "250 transactions in 03".) Only the months will be kept as all transactions take place within the same year, so there is no need to specify the year. Specific days are outside of the scope of the detailed report. This transformation will occur when the data is pulled from the 'rental' table and inserted into the 'detailed' table.

A5: The summary table can be used to show at a glance what the number of rentals is each month and at each store, which stores are performing better, and if rentals are increasing or decreasing or if there is no consistent trend over the four-month period. If there is a decrease in rentals over the four months, it could indicate that business strategies need to be altered (i.e., different marketing strategies, better store management, etc.). In contrast, if rentals are increasing, that can indicate that current business strategies and practices are working effectively.

The detailed table can provide more insight than the summary table because it will also show the number of transactions performed per employee, which can provide additional context. This can indicate which employees are performing acceptably and which may require adjustment. If a trend shows that all the employees perform better at a specific location on average, it can also indicate an issue with management.

A6: This report should be updated on a month-to-month basis to monitor monthly trends. This can be scheduled for the first of every month. Eventually, it should also be refreshed with year-to-year reporting for a broader view of trends.

B: CREATE OR REPLACE FUNCTION rental\_timestamp\_to\_month(rts TIMESTAMP)

RETURNS TEXT

LANGUAGE plpgsql AS $$

BEGIN

RETURN TO\_CHAR(rts, 'Month');

END;

$$;

C: CREATE TABLE rental\_summary (

store\_id INT,

month VARCHAR(9),

total\_rentals INT,

PRIMARY KEY (store\_id, month)

);

CREATE TABLE rental\_details (

store\_id INT,

staff\_id INT,

month VARCHAR(9),

total\_rentals INT,

PRIMARY KEY (store\_id, staff\_id, month),

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

FOREIGN KEY (staff\_id) REFERENCES staff(staff\_id)

);

D: INSERT INTO rental\_details (staff\_id, store\_id, month, total\_rentals)

SELECT

staff.staff\_id,

store.store\_id,

rental\_timestamp\_to\_month(rental\_date) AS month,

COUNT(\*) AS total\_rentals

FROM

staff

INNER JOIN

rental ON staff.staff\_id = rental.staff\_id

INNER JOIN

store ON staff.store\_id = store.store\_id

GROUP BY

store.store\_id, staff.staff\_id, month

ORDER BY

store.store\_id, staff.staff\_id, month;

E: CREATE OR REPLACE FUNCTION update\_summary()

RETURNS TRIGGER

LANGUAGE plpgsql AS $$

BEGIN

DELETE FROM rental\_summary;

INSERT INTO rental\_summary

SELECT store\_id, month, total\_rentals

FROM rental\_details

GROUP BY store\_id, month, total\_rentals

ORDER BY store\_id, month;

RETURN NEW;

END;

$$;

CREATE TRIGGER after\_rental\_details\_update

AFTER INSERT ON rental\_details

FOR EACH ROW

EXECUTE FUNCTION update\_summary();

F: CREATE OR REPLACE PROCEDURE data\_refresh()

LANGUAGE plpgsql AS $$

BEGIN

DELETE FROM rental\_details;

DELETE FROM rental\_summary;

INSERT INTO rental\_details (staff\_id, store\_id, month, total\_rentals)

SELECT

staff.staff\_id,

store.store\_id,

rental\_timestamp\_to\_month(rental\_date) AS month,

COUNT(\*) AS total\_rentals

FROM

staff

INNER JOIN

rental ON staff.staff\_id = rental.staff\_id

INNER JOIN

store ON staff.store\_id = store.store\_id

GROUP BY

store.store\_id, staff.staff\_id, month

ORDER BY

store.store\_id, staff.staff\_id, month;

RETURN;

END;

$$;

--Because of the trigger, the summary report is automatically updated with the detailed report.

--Queries that are used to demonstrate the code is working as intended:

--SELECT \* FROM rental\_details;

--SELECT \* FROM rental\_summary;

--DELETE FROM rental\_details;

--CALL data\_refresh();

F1: A job scheduling tool that can be used is pgAgent. It is designed for PostgreSQL and is used to schedule and automate tasks, such as the stored procedure from Part F. This can be scheduled for the first of the month every month so that the reports can be updated with the previous month's data.