Jacob Luttrull

011218373

D326 Advanced Data Management

06/24/24

Business Report for DVD Rental Store

1. For my report I will be a making a month-by-month report on which film category gets the most rentals. I will be using data from June to August 2005. For the summary table I will be showing the top categories for that month and the number of rentals. For the detailed table I will have the rental information with what category the rental is in and all the relevant ids. This report will be helpful for determining which category is most popular at the time and make sure we have the relevant movies to address that.
   1. I will Include these columns in my summary table.

month VARCHAR,

category\_name VARCHAR(50),

total\_rentals INT

I will include these columns in my detailed table.

rental\_date TIMESTAMP,

category\_name VARCHAR(45),

rental\_count INT,

film\_title VARCHAR(255),

store\_id INT,

customer\_id INT,

rental\_id INT

* 1. For this project I used VARCHAR() AND INT for most of my columns but TIMESTAMP for rental\_date for more accuracy.
  2. The summary table in my report will pull from the detailed table. The detailed table will pull from the rental, film, and film\_category tables.
  3. For my report I transformed the rental\_date TIMESTAMP into just TEXT field that shows the ‘Month’ rather than a whole timestamp for more readability for stakeholders or any higher ups.
  4. The summary table is only a couple of columns but overtime as the columns grow, the data can show trends of what categories get rented out the most as times change, and contemporary trends come along. The detail report is more comprehensive and can be used to show what time a particular film was rented or what time. This can be used for inventory management to ensure adequate copies and optimal times that movies get rented out as well could be shown with the rental\_date timestamp.
  5. This data would be refreshed monthly so seeing each other whenever the data refreshes could be a promising idea so we can pivot any actions if needed. This data also refreshes at the end of the month so seeing each other the first week of the month would be optimal to make sure our business is on the right track.

CODING STARTS HERE, I WILL INCLUDE COMMENTS IN THE CODE.

**------ SECTION B USER-DEFINED FUNCTION-----**

CREATE OR REPLACE FUNCTION month\_from\_date(rental\_date TIMESTAMP)

RETURNS TEXT

LANGUAGE plpgsql

AS $$

DECLARE

return\_of\_month TEXT;

BEGIN

return\_of\_month := TO\_CHAR(rental\_date,'Month');

RETURN return\_of\_month;

END;

$$;

-**----TESTING-----**

SELECT month\_from\_date('2021-08-22'); -- AUGUST SHOULD BE THE ANSWER –

**-----CREATING DETAILED TABLE FOR SECTION C-----**

CREATE TABLE detailed\_table\_rentals\_month (

rental\_date TIMESTAMP,

category\_name VARCHAR(45),

rental\_count INT,

film\_title VARCHAR(255),

store\_id INT,

customer\_id INT,

rental\_id INT

);

**-----CHECKING TABLES IN SECTION C-----**

SELECT \* FROM detailed\_table\_rentals\_month;

SELECT \* FROM summary\_table\_rentals\_month ORDER BY month, total\_rentals ASC ;

**----- SECTION E TRIGGER FUNCTION FOR SUMMARY TABLE-----**

CREATE OR REPLACE FUNCTION trigger\_summary()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM summary\_table\_rentals\_month;

INSERT INTO summary\_table\_rentals\_month (month, category\_name, total\_rentals)

SELECT

month,

category\_name,

total\_rentals

FROM (

SELECT

month\_from\_date(d.rental\_date) AS month,

d.category\_name,

COUNT(\*) AS total\_rentals,

ROW\_NUMBER() OVER (PARTITION BY month\_from\_date(d.rental\_date) ORDER BY COUNT(\*) DESC) AS rank --TO SHOW TOP CATEGORY INSTEAD OF ALL CATEGORIES FOR SIMPLICITY

FROM

detailed\_table\_rentals\_month d

GROUP BY

month\_from\_date(d.rental\_date), d.category\_name

) AS ranked

WHERE ranked.rank = 1;

RETURN NEW;

END;

$$;

CREATE TRIGGER summary\_trigger\_fn

AFTER INSERT OR DELETE OR UPDATE ON detailed\_table\_rentals\_month

FOR EACH STATEMENT

EXECUTE FUNCTION trigger\_summary();

**-----SECTION D INSERTING THE RAW DATA INTO THE DETAIL TABLE, THIS WILL ALSO POPULATE THE SUMMARY TABLE AS WELL-----**

INSERT INTO detailed\_table\_rentals\_month (rental\_id, rental\_date, category\_name, rental\_count, film\_title, store\_id, customer\_id)

SELECT

r.rental\_id,

r.rental\_date,

c.name AS category\_name,

COUNT(\*) AS rental\_count,

f.title AS film\_title,

i.store\_id,

r.customer\_id

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\_category fc ON f.film\_id = fc.film\_id

JOIN

category c ON fc.category\_id = c.category\_id

WHERE

r.rental\_date >= '2005-06-01 00:00:00'

AND r.rental\_date <= '2005-09-30 23:59:59'

GROUP BY

r.rental\_id, r.rental\_date, c.name, f.title, i.store\_id, r.customer\_id;

**----- SECTION F STORED PROCEDURE TO REFRESH DATA -----**

CREATE OR REPLACE PROCEDURE rentals\_refresh()

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM detailed\_table\_rentals\_month;

INSERT INTO detailed\_table\_rentals\_month (rental\_id, rental\_date, category\_name, rental\_count, film\_title, store\_id, customer\_id)

SELECT

r.rental\_id,

r.rental\_date,

c.name AS category\_name,

COUNT(\*) AS rental\_count,

f.title AS film\_title,

i.store\_id,

r.customer\_id

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\_category fc ON f.film\_id = fc.film\_id

JOIN

category c ON fc.category\_id = c.category\_id

WHERE

r.rental\_date >= '2005-06-01 00:00:00'

AND r.rental\_date <= '2005-09-30 23:59:59' -- Up to the last second of September 31st

GROUP BY

r.rental\_id, r.rental\_date, c.name, f.title, i.store\_id, r.customer\_id;

RETURN;

END;

$$

CALL rentals\_refresh(); **-----CALLING THAT REFRESH ----**

**END OF CODING**

**F1**

We use PostgreSQL for this project so I would use pgAgent for job scheduling. This software allows you to run your tasks even when someone is not present, and this would be helpful since the refresh would happen around the middle of the night when no store would be open that would be extremely helpful. Also great for backups incase power outage or power surges.

G

Here is a link to my video. <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=4e2af5ea-0d41-4e2f-9cf9-b19b000c3e1c> .

H

I did not use any third-party sources for this project.