D191 Advanced Data Management

Performance Assessment

Western Governors University

Christopher Owens

SID# 010682412

05/01/23

Section A - Introduction:

A DVD rental business would like to know the sum of payments collected by each employee. This can be useful in developing a performance improvement plan for any struggling employees or could be used to incentivize high performance.

Section A1 – Specific Fields:

The specific fields needed to assess the business problem of employee sales totals come from the payment and staff tables included in the business’s dataset. From payment we need to use the data from the amount column and the staff\_id column. Also, from the staff table we need to use the data from the staff\_id column, first\_name column, last\_name column, store\_id column and the active column. These data sets will be sufficient to identify the staff member, determine if the staff member is a current employee and sum the total amount of payments each staff member has collected.

Section A2 – Types of Data Fields Used:

The types of data fields used include: varchar for the first\_name and last\_name fields in staff table, integer for the store id in staff table, integer for staff id fields in the payment and staff tables, a Boolean for the active field in staff table, and a numeric data type for the amount field from payment table.

Section A3 – Identify Two Tables:

The two tables being used are the payment table and the staff table from the DVD rental database. These tables will provide the needed information to populate the detailed table and the summary table.

Section A4 – Identify Fields to Transform:

The fields to transform are the first\_name and last\_name fields form the staff table. The combine\_name function, when called during insertion into detailed table, will pass in the staff\_id as int and concatenate the first\_name and last\_name fields as the full\_name of the staff member. The full\_name variable is then returned to the detailed table insertion query.

Section A5 – Business Uses:

The business uses of finding the total sales by staff member allows the business manager to assess each team member’s performance. The detailed table will have employee name and ID for identification, have the store id for location, an indicator if the employee is still actively employed, and it also gives the amount of each sale made. The detailed table can be used as a reference point to the summary table. If the staff member is an active employee the summary table gives the staff member name, id number and the sum of the sales made. The managers can use this information to reinforce positive sales performance or to create a performance improvement plan for underachieving employees.

Section A6 – Report Refresh:

The employee sales report should be refreshed monthly. Any more frequent and trends in performance would not be as well defined. If the refresh was any less frequently the business could suffer if a staff member has poor performance.

Section B – Function SQL:

-- combine\_name concatenates the first and last name of the employees from staff into full\_name  
  
DROP FUNCTION IF EXISTS combine\_name;  
CREATE OR REPLACE FUNCTION combine\_name(staff\_num int)  
RETURNS varchar(60)  
LANGUAGE plpgsql  
AS  
$name$  
DECLARE  
 full\_name varchar(60);  
BEGIN  
 SELECT CONCAT(s.first\_name||' '||s.last\_name)   
 INTO full\_name  
 FROM staff AS s  
 WHERE s.staff\_id = staff\_num;  
RETURN full\_name;  
END;  
$name$;

Section C – Creating Tables SQL:

-- Create detailed\_report table  
  
DROP TABLE IF EXISTS detailed\_report;  
CREATE TABLE detailed\_report (  
 staff\_id smallint,  
 full\_name varchar(60),  
 store\_id smallint,  
 payment\_amounts numeric(5,2),  
 active bool  
);  
  
-- Create summary\_report table  
  
DROP TABLE IF EXISTS summary\_report;  
CREATE TABLE summary\_report (  
 staff\_id smallint,  
 staff\_name varchar(60),  
 total\_sales numeric(7,2)  
);

Section D – SQL Query raw data for detailed section:

-- This insert calls the combine\_name function (section B) to transform the first\_name and last\_name into full\_name  
-- Insert also checks that the employee is active  
  
INSERT INTO detailed\_report (  
 staff\_id,   
 full\_name,  
 store\_id,   
 payment\_amounts,  
 active  
)  
SELECT s.staff\_id,   
 combine\_name(s.staff\_id) AS full\_name,  
 store\_id,   
 amount,  
 active   
FROM staff s, payment p  
WHERE p.staff\_id = s.staff\_id  
AND active;

Section E – Trigger SQL:

-- The summary\_update trigger will refresh the summary\_report through the use of the sales\_by\_staff() function  
  
CREATE OR REPLACE FUNCTION sales\_by\_staff()  
RETURNS TRIGGER   
AS  
$$  
BEGIN   
 DELETE FROM summary\_report;  
 INSERT INTO summary\_report (staff\_id, staff\_name, total\_sales)  
 SELECT staff\_id,  
 full\_name,  
 sum(payment\_amounts)  
 FROM detailed\_report  
 GROUP BY staff\_id, full\_name;   
  
RETURN new;   
END;  
$$  
LANGUAGE PLPGSQL;  
  
  
CREATE TRIGGER summary\_update  
AFTER INSERT ON detailed\_report  
FOR EACH STATEMENT  
EXECUTE PROCEDURE sales\_by\_staff();

Section F – Report Refresh SQL:  
  
--The summary\_report will be updated by the trigger summary\_update on any statement change in detailed\_report  
--The report\_refresh() procedure is to be called by external tool.  
  
CREATE OR REPLACE PROCEDURE report\_refresh()  
LANGUAGE PLPGSQL  
AS  
$refresh$  
BEGIN  
DELETE FROM detailed\_report;  
INSERT INTO detailed\_report (  
staff\_id,   
full\_name,  
store\_id,   
payment\_amounts,  
active  
)  
SELECT s.staff\_id,   
combine\_name(s.staff\_id) AS full\_name,  
store\_id,   
amount,  
active   
FROM staff s, payment p  
WHERE p.staff\_id = s.staff\_id  
AND active = true;  
END;  
$refresh$;

Section F1 – Report Refresh automated tool:

Pgadmin can be used as the automated job scheduling tool to refresh the employee sales data contained in the detailed and summary tables.

Section G – Panopto Video:

See attached link.

[https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=7bab84e4-dff5-41e6-b402-aff600f44633#](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=7bab84e4-dff5-41e6-b402-aff600f44633%23)

Section H – Sources:

No third-party code, in-text citations and references, or external sources were used to support my submission.