Love Patel  
D191 Advanced Data Management

PA

1. **Summarize one real-world business report that can be created from the attached Data Sets and Associated Dictionaries**.

From the Attached data sets and associated dictionaries one real world business report that can be created is to see *how many rental sales each employee had*. This can help us in a multitude of ways. One is that we can see which employee is consistently performing better. A manager can also incentivize employees with additional rewards and bonus payments for a good work. It can also help the DVD rental business to do a friendly competition between the employees. It also gives data on which employee is selling the most rentals through which we can further deduce and observe what this particular staff member does to generate such results consistently and that information can be used to train the remaining or new staff to boost the rental sales.

1. **Describe the data used for the report.**

The data used for this report are the rental and staff information from the dvd rental database. These two data sets will give us all the information we need to deduce for our reports. Through this data we can count the total rental sales done by each staff and see how each staff member is performing in descending order.

1. **Identify two or more specific tables from the given dataset that will provide the data necessary for the detailed and the summary sections of the report.**

The two tables we need for the detailed and the summary sections of the report are rental table and staff table. Combining these two will give us our desired report.

1. **Identify the specific fields that will be included in the detailed and the summary sections of the report.**

For the detailed table, these fields will be included:

* rental\_id
* rental\_date
* inventory\_id
* staff\_id
* first\_name
* last\_name
* email

For the summary table, these fields will be included:

* staff\_name
* staff\_id
* rental\_count

1. **Identify one field in the detailed section that will require a custom transformation and explain why it should be transformed. For example, you might translate a field with a value of ‘N’ to ‘No’ and ‘Y’ to ‘Yes’.**

One field that will need to be transformed is the first\_name and the last\_name of the staff. To present it more coherently in the summary table these fields will be concatenated such that it shows “first\_name last\_name”. So both fields will be transformed into one separated by a space for easier reading of the staff member’s full name.

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

The detailed section stores all the necessary information required to show what staff member sold which rentals and which inventory are rented out by whom. It also shows the date the particular dvd was rented and who sold the rental. It also shows the details of the staff member that sold the dvd rental like their first and last name, their email, their id etc.

The summary section of the report shows the total rental sold by each staff member and it arranges the data in descending order to see who sold the highest and who sold the lowest. IT also helps give more incentives and bonuses internally to the staff members to motivate them to be better with their job. It also facilitates the knowledge of what works and how to implement it among all staff members to be like the best one. This section also shows the most important data needed for a quick viewing.

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

The report should be refreshed atleast monthly. It can also be refreshed weekly. It can be refreshed depending on what the employee pay out period is. If the salaries are paid weekly it would be wise to do a weekly refresh of the report. This refresh helps us get the accurate representation of which staff members did the highest rental sales and to give bonuses accordingly.

1. **Write a SQL code that creates the tables to hold your report sections.**

--Detailed Table

DROP TABLE if EXISTS detailed;

CREATE TABLE detailed (

rental\_id integer primary key, --rental

rental\_date DATE, --rental

inventory\_id integer, --rental

staff\_id integer, --staff

first\_name varchar, --staff

last\_name varchar, --staff

email varchar --staff

);

--Summary Table

DROP TABLE if EXISTS summary;

CREATE TABLE summary (

staff\_name varchar, --concatenate first and last name for transformation

staff\_id integer primary key, --staff

rental\_count integer

);

1. **Write a SQL query that will extract the raw data needed for the Detailed section of your report from the source database and verify the data’s accuracy.**

INSERT INTO detailed (

rental\_id, --rental

rental\_date, --rental

inventory\_id, --rental

staff\_id, --staff

first\_name, --staff

last\_name, --staff

email --staff

)

SELECT

r.rental\_id, r.rental\_date, r.inventory\_id,

s.staff\_id, s.first\_name, s.last\_name, s.email

FROM staff AS s

INNER JOIN rental AS r ON s.staff\_id = r.staff\_id;

-- This loads information into the detailed table.

1. **Write code for function(s) that perform the transformation(s) you identified in part A4.**

CREATE FUNCTION refreshSummary()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM summary; --emptying summary

INSERT INTO summary

SELECT

concat\_ws(' ', first\_name, last\_name) AS staff\_name,

staff\_id,

COUNT(staff\_id)

FROM detailed

GROUP BY staff\_id, staff\_name

ORDER BY count(staff\_id)DESC

LIMIT 90;

RETURN NEW;

END;

$$; --function To refresh summary and also performs the transformation mentioned in A4

1. **Write a SQL code 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.**

CREATE TRIGGER summaryRefresh --This trigger continually update the summary table as data is added

AFTER INSERT ON detailed

FOR EACH STATEMENT

EXECUTE PROCEDURE refreshSummary();

1. **Create a stored procedure that can be used to refresh the data in both your detailed and summary tables. The procedure should clear the contents of the detailed and summary tables and perform the ETL load process from part C and include comments that identify how often the stored procedure should be executed.**

CREATE PROCEDURE refreshBoth() --execute weekly/monthly depending on salary payout period

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM detailed; --to empty the detailed table

INSERT INTO detailed (

rental\_id, --rental

rental\_date, --rental

inventory\_id, --rental

staff\_id, --staff

first\_name, --staff

last\_name, --staff

email --staff

)

SELECT

r.rental\_id, r.rental\_date, r.inventory\_id,

s.staff\_id, s.first\_name, s.last\_name, s.email

FROM staff AS s

INNER JOIN rental AS r ON s.staff\_id = r.staff\_id;

-- new data gets back in the detailed table

END; $$;

CALL refreshBoth(); -- to call the procedure above

--below code is to view the result

SELECT\* FROM detailed;

SELECT\* FROM summary;

1. **Explain how the stored procedure can be run on a schedule to ensure data freshness.**

The stored procedure can be automatically run on a schedule by creating a job with the SQL server agent. While creating the job we can put a recurring schedule to make it run automatically on our desired repeat schedule.  
The stored procedure can be run monthly at the very least. Ideally it will be run weekly or bi-weekly. The frequent use of the freshness of the data will be when the employee salary pay out period comes. Using the fresh data, rewards and bonuses can be assigned. It can also be refreshed during stakeholder meetings. Although that one will not be as frequent as the salary period. It can also be used when a new staff member is hired to see who the best trainer can be. It can be refreshed when the manager wants to see how productive the staff has been in the past period. Shortening the run schedule of the stored procedure on a weekly basis can also help with a fun friendly staff competition to boost the productivity and motivation of the staff members.