

DATA ANALYSIS

D191 PERFORMANCE ASSESSMENT

JEREMIAH MURRAY

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.

We can use the dataset to see which store has the least number of rentals compared to the other. With this info, we can see if there is anything that is causing one store to drag behind the other. We would then be able to take a specific action, whether that be focusing advertising on that particular store, adjusting training programs for staff to cover any gaps in customer service, etc.

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

Fields in each table are:

Detailed_table: store_id integer, manager_staff_id integer, staff_id integer, rental_id integer, rental_date timestamp.

Summary_table: store_id integer, staff_id integer, rental_count integer

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

The data used in each table is store and staff ID and all of the necessary rental information needed to compare the rentals between the two stores. The store and staff information data fields are used to tie the respective number of rentals to each particular store, we can then compare the numbers to see which one is falling behind and why. The rental information both counts all the rentals and records the time of the rental as well.

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.

There were three tables used from the dataset to provide the necessary data from the detailed and summary sections: rental, staff, and store.

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 rental_id field will require the transformation. There are over 16,000 rentals between each store, so I used an aggregate function (COUNT) to group all the rentals together by staff and store ID.

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

The detailed table can be used to see if there is any correlation between the number of rentals processed and the time of year said rentals are processed. The summary table shows exactly how many rentals each store has processed. Both tables can be used together to determine which store is lagging behind and allow the stakeholders to target assistance toward that particular store.

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

The report should be run at least bi-monthly in order to keep track of the rental numbers and to see if there is any improvement or decline between each store.

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

```
CREATE FUNCTION store_rentals()
RETURNS TRIGGER
LANGUAGE plpgsql
AS
$$
BEGIN
    DELETE FROM summary_table;
    INSERT INTO summary_table (store_id, staff_id, rental_count)
    SELECT
        store_id,
        staff_id,
        COUNT(rental_id)
    FROM detailed_table
    GROUP BY store_id, staff_id
    ORDER BY store_id, COUNT(rental_id) asc;
RETURN NEW;
END;
$$
```

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

Detailed Table:

```
DROP TABLE IF EXISTS detailed_table;
Create Table detailed_table (
    store_id integer,
    manager_staff_id integer,
    staff_id integer,
    rental_id integer,
    rental_date timestamp

);
```

Summary Table:

```
DROP TABLE IF EXISTS summary_table;
CREATE TABLE summary_table(
    store_id integer,
    staff_id integer,
    rental_count integer

);
```

- 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 detailed_table(
    store_id,
    manager_staff_id,
    staff_id,
    rental_id,
    rental_date
)
SELECT
    s.store_id,
    s.manager_staff_id,
    sta.staff_id,
    r.rental_id,
    r.rental_date

FROM store AS s
INNER JOIN staff AS sta ON sta.store_id = s.store_id
INNER JOIN rental as r ON r.staff_id = sta.staff_id;
```

- 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.**

```
CREATE TRIGGER summary_table_refresh
AFTER INSERT ON detailed_table
FOR EACH STATEMENT
EXECUTE FUNCTION store_rentals();
```

- 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.**

```
CREATE OR REPLACE FUNCTION detailed_refresh()
RETURNS VOID
LANGUAGE plpgsql
AS
$$
BEGIN
    DELETE FROM detailed_table;

    INSERT INTO detailed_table (
        store_id,
        manager_staff_id,
        staff_id,
        rental_id,
        rental_date
    )
    SELECT
        s.store_id,
        s.manager_staff_id,
        sta.staff_id,
        r.rental_id,
        r.rental_date

    FROM store AS s
    INNER JOIN staff AS sta ON sta.store_id = s.store_id
    INNER JOIN rental as r ON r.staff_id = sta.staff_id;

END;
$$
```

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

Extension `pg_cron` is a tool that can be used to automate stored procedures. It allows the execution of database tasks directly from the database. (Dias, 2022)

References

Dias, H. (2022, May 4). *An overview of job scheduling tools for postgresql*. Severalnines.
<https://severalnines.com/blog/overview-job-scheduling-tools-postgresql/>