Aidan Strawder

D191 Advanced Data Management

Date: 01/09/2023

Business Report

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

There are a various number of reports that can be created using the attached data sets and associated dictionaries. My report covers number of rental sales by each staff member. This information is useful to any business that measures employee/staff performance which is normally present on performance evaluation reports.

Describe the data used for the report.

The data used for this report consists of rental information and staff information. The staff information is needed to identify each staff member and possibly contact them through email if needed. The rental information associated with each staff member is used to measure performance.

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 rental table and the staff table will provide the data necessary for the detailed and summary sections of the report. Combining and measuring information from these two tables will allow us to measure the performance of each staff member.

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

For the detailed section of the report:

- rental_id integer
- rental_date timestamp
- return_date timestamp
- store_id integer
- staff_id integer
- first_name varchar (45)
- last_name varchar (45)
- email varchar (90)

For the summary section of the report:

- staff_full_name varchar (90)
- email varchar (90)
- sales_per_staff integer

Identify one field in the detailed section that will require a custom transformation and explain why it should be transformed.

The first_name and last_name fields will be transformed when extracted from the detailed table before being loaded into the summary table. These fields in the detailed table require a custom transformation to form a full name for each staff member. This improves the readability of staff member names when viewing the report and reduces the number of unnecessary columns within the report.

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

The detailed section of the report will be used to see every rental sale and the staff member associated with that sale and can be traced back to each store location the transaction

occurred. This can help track the performance of staff members and trace the performance down to each store if needed.

The summary section of the report will be used to see individual performance among staff members. This information can be used to make decisions involving staff members such as promotions or possible dismissal. This may also raise other questions and can also aid in improving the training of staff.

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

The report should be refreshed every 6-12 months when an evaluation report is needed across the entire business which would cover performance of the business which normally consists of staff/employee performance. This will earn or keep the trust and confidence of stakeholders on how performance is being managed and improved.

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

The stored procedure can be automated using the pgAgent software that can be used with PostgreSQL. A job can be created using pgAgent that will run the stored procedure on a defined schedule. This will ensure that the tables stay updated on schedule, so performance data of staff members is accurate.

SQL Code

```
-- Business Question: Total sales per staff member


-- CREATE detailed table
DROP TABLE IF EXISTS detailed;
CREATE TABLE detailed (
	rental_id integer
	rental_date timestamp,
	return_date timestamp,
	store_id integer,
	staff_id integer,
	first_name varchar(45),
	last_name varchar(45),
	email varchar(90)
)
-- To view empty detailed table
-- SELECT * FROM detailed;


-- CREATE summary table


DROP TABLE IF EXISTS summary;
CREATE TABLE summary (
	staff_full_name varchar(90),
```

```
        email varchar(90),
        sales_per_staff integer
)

-- To view empty summary table
-- SELECT * FROM summary;

-- Extract raw data from database into detailed table
INSERT INTO detailed(rental_id, rental_date, return_date, store_id, staff_id, first_name,
last_name, email)
SELECT s.store_id, r.rental_id, r.rental_date, r.return_date, r.staff_id, s.first_name, s.last_name,
s.email FROM rental AS r
JOIN staff AS s ON r.staff_id = s.staff_id;

-- To view contents of detailed table
-- SELECT * FROM detailed;

-- CREATE FUNCTION refreshing the summary table with data transformations
-- Transform first_name and last_name with concatenation into staff_full_name
-- Transform staff_id using aggregation to COUNT the staff IDs associated with each rental
transaction

CREATE FUNCTION summary_transform()
```

```
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN

DELETE FROM summary;

INSERT INTO summary(
        SELECT concat_ws(' ', first_name, last_name) AS staff_full_name, email,
        COUNT(staff_id) AS sales_per_staff FROM detailed
        GROUP BY staff_id, staff_full_name, email
);
RETURN NEW;
END;$$

-- CREATE TRIGGER
CREATE TRIGGER summary_transform_trigger
AFTER INSERT
ON detailed
FOR EACH STATEMENT
EXECUTE PROCEDURE summary_transform();
```

```
-- CREATE STORED PROCEDURE

-- To be automated to run every 6-12 months for performance evaulation purposes

-- Use pgAgent as an external scheduler for running the procedure


CREATE PROCEDURE refresh_data()

LANGUAGE plpgsql

AS $$

BEGIN


DELETE FROM detailed;


INSERT INTO detailed(store_id, rental_id, rental_date, return_date, staff_id, first_name,

last_name, email)

SELECT s.store_id, r.rental_id, r.rental_date, r.return_date, r.staff_id, s.first_name, s.last_name,

s.email FROM rental AS r

JOIN staff AS s ON r.staff_id = s.staff_id;


END;$$


-- To call the stored procedure

-- CALL refresh_data();
```

```
-- For verification run this code and compare with the results in the summary table to verify
accuracy of each staff member's total sales

SELECT staff_id, COUNT(staff_id) FROM rental
GROUP BY staff_id;

-- To view results
-- SELECT * FROM detailed;
-- SELECT * FROM summary;
```

Works Cited

No sources, web sources or third-party code were cited, referenced, or used.