D191 Business Report

Business Problem

As a stakeholder or a data analyst in the DVD Rental Business, it could prove to be highly valuable to have a convenient way to aggregate a report of all active rentals, as well as a summary of all active rentals. DVD rental companies have a need to find all active rentals by customer and by store, for inventory purposes and for tracking late rentals. Additionaly, DVD rental companies have a need to not lend out too many DVD's to any one customer, so the summary will be used to aggregate a quantity of DVD rentals, grouped by customer. Customer information will also be stored in the `active_rentals_summary` table for convenient retrieval. As such, a functionality to generate a report which extracts this data from the existing, live, production `dvd_rental` database has been implemented.

Business Reports

Two reports have been made available, `active_rentals_detail` and `active_rentals_summary`, in the form of tables in the database which can be queried.

`active_rentals_detail`

The `active_rentals_detail` report will be comprised of data from many tables in the `dvd_rentals` database.

- `rental`
  - `rental_id`
  - `rental_date`
  - `return_date` (not included in report, used to determine that the rental is not returned)
  - `inventory_id`
  - `customer_id`
- `inventory`
  - `inventory_id`
  - `film_id`
  - `store_id`
- `film`
  - `film_id`
  - `title`
  - `rental_duration`
- `customer`
  - `customer_id`
  - `first_name`
  - `last_name`
  - `email`
  - `address_id` {JOIN}
- address
  - `address_id` {JOIN}
  - `phone`

Columns

In order to meet the businesses' needs, this table will contain the following columns:

- `rental_id`
- `rental_date`
- `rental_duration`
- `film_title`
- `inventory_id`
- `due_date`
- `customer_id`
- `customer_full_name` (concatenation of `first_name` and `last_name`, from `customer` table). This value should be transformed because it will be more convenient to the consumers of this report to have the full name of the customer in one column.
- `customer_phone`
- `customer_email`
- `store_id`
- `last_updated`

Use cases

There are many potential uses for the `active_rentals_detail` report. The most obvious use cases are as follows:

- Find all active rentals by customer and store
- For inventory purposes, keep track of all pieces of inventory that are actively rented out (thus missing from inventory)
- To track late rentals

`active_rentals_summary`

The `active_rentals_summary` report will be comprised of data coming directly from the `active_rentals_detail` table.

Data Sources

The following columns from the `active_rentals_detail` table will be used for the `active_rentals_summary` table:

- `customer_id`
- `customer_full_name`
- `customer_phone`
- `customer_email`
- `store_id`

Columns

- `customer_id`
- `customer_full_name`

- `customer_phone`
- `customer_email`
- `store_id`
- `quantity_active_rentals` (leverages `COUNT()` function in the postgresql database)
- `last_updated`

Use cases

The most prevalent use case for the `active_rentals_summary` is to get the quantity of rentals for any given customer, so that they don't exceed any limitation policies. This report could also be used for determining if a customer is renting from multiple stores, though this use-case is not nearly as useful as the first.

Refresh procedure and frequency

A stored procedure, `active_rentals_refresh`, will be used for extracting, transforming and loading data into the appropriate tables in the database. This procedure will truncate the tables, clearing them of any potentially stale data, then will extract data from appropriate tables, and load it into the `active_rentals_detail` and `active_rentals_summary` tables.

This procedure should be ran nightly, after close of business, so that data is up to date in the morning, for any stores which need to access it. Additionally, the procedure can be ran, as needed, in the case of alleged discrepancies in the accuracy of data at the time of sale (rental). Any sort of automation tool could be used, whether that be Windows task scheduler, or even a simple java or python application could be used to refresh the data nightly.

F. 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.

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

G. Provide a Panopto video recording that includes a demonstration of the functionality of the code used for the analysis and a summary of the programming environment.