CSN1 Task:1 Data Analysis

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.   
As the owner of a DVD rental store, I'm really keen on finding out which movie categories are the most popular among our customers each month. This information will help us figure out which types of movies we should stock more of. This will increase sales and customer satisfaction.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable name | Summary/Detailed | Database Table | Datatype | Description |
| genre | Summary | category | VARCHAR(100) | Movie category |
| totalRentals | Summary | film | INT | Total rentals/category |
| category\_id | Detailed | category | INT | Category ID Number |
| film\_id | Detailed | film\_category | INT | Film ID Number |
| title | Detailed | film | VARCHAR(255) | Film title |
| rental\_id | Detailed | Rental | INT | Rental ID |
| customer\_id | Detailed | Customer | INT | Customer ID |
| film\_category | Detailed | category | VARCHAR(25) | Film category |
| payment\_id | Detailed | Payment | INT | Payment ID |
| rental\_month | Detailed | Rental | TIMESTAMP | Rental month |

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

The summary table will encompass the following fields: 'Genre' and 'Total Rentals'. The 'Genre' category name shall be represented as a variable string data type with a maximum capacity of 100 characters, exhibiting the genre label. The 'Total Rentals' will undertake the task of enumerating the aggregate count of rentals associated with each respective genre.

On the other hand, the detailed table will feature the following attributes: 'Category ID', 'Film ID', 'Title', 'Rental ID', 'Customer ID', 'Film Category', 'Payment ID', and 'Rental Month'. 'Category ID,' 'Film ID,' 'Rental ID,' 'Customer ID,' and 'Payment ID' will be of integer data type, while 'Title' and 'Film Category' will be represented as variable strings. Furthermore, 'Rental Month' shall be designated as a timestamp data type, providing a comprehensive record of the precise month, day, year, and time at which the rental transaction occurred.

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.

For the detailed table, I will incorporate the following attributes: 'Category ID,' 'Film ID,' 'Title,' 'Rental ID,' 'Customer ID,' 'Film Category,' 'Payment ID,' and 'Rental Month.'

In the summary table, I shall include the subsequent fields: 'Genre' (derived from the 'Category Name') and 'Total Rentals' (computed from the count of rentals).

This analysis will draw upon data from the 'Category,' 'Film Category,' 'Payment,' 'Inventory,' and 'Film' database tables. The 'Category' table will furnish the category name and category ID. The 'Film Category' table will provide the 'Film Category' ID and its corresponding 'Category ID.' The 'Film' table will serve as the primary source of rental records.

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).

I am going to extract the month from the rental\_date column in the rental table. So, only the month number will show. By performing this transformation, it would be easier to identify the rentals for each month.

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

The detailed table allows us to dig deeper for insights. For example, it helps us find out which customers are renting the most movies from the popular categories.

On the other hand, the summary table is handy for a quick overview of what's currently trending. It's especially useful because the most popular category can change monthly. Having this quick summary helps us adjust inventory levels more effectively.6. Explain how frequently your report should be refreshed to remain relevant to stakeholders.

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

Since the stakeholders are interested in the most popular category each month, the report should be refreshed monthly to analyze the rentals.

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

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

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.

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.

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.

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

Extension pg\_cron can be used as a scheduling tool to automate my stored procedure. It can run several jobs at the same time, and it can execute database tasks directly from the database. This will be very helpful to use while performing stored procedures, and PostgreSQL commands.

H. Acknowledge all utilized sources, including any sources of third-party code, using in-text citations and references. If no sources are used, clearly declare that no sources were used to support your submission.

I will not be using outside sources for this assignment.

SQL CODE

--Part B: Transformations

CREATE OR REPLACE FUNCTION rentalMonth(rental\_date TIMESTAMP)

RETURNS INT

LANGUAGE plpgsql

AS $$

DECLARE rentalMonth INT;

BEGIN

SELECT EXTRACT(MONTH FROM rental\_date) INTO rentalMonth;

RETURN rentalMonth;

END;

$$

--Part C: Creates Detailed and Summary table—

DROP TABLE IF EXISTS Detailed;

CREATE TABLE Detailed(

rental\_id INT ,

customer\_id INT,

film\_id INT ,

category\_id INT,

title VARCHAR(255),

film\_category VARCHAR(25),

payment\_id INT,

rental\_month TIMESTAMP

);

DROP TABLE IF EXISTS Summary;

CREATE TABLE Summary(

genre VARCHAR(100),

totalRentals INT

);

--Part D: Extracts raw data for detailed table--

INSERT INTO Detailed(

rental\_id,

customer\_id,

film\_id,

category\_id,

title,

film\_category,

payment\_id,

rental\_month

)

SELECT

r.rental\_id,

r.customer\_id,

f.film\_id,

cg.category\_id,

f.title,

cg.name,

p.payment\_id,

r.rental\_date

FROM rental AS r

INNER JOIN inventory AS i ON r.inventory\_id = i.inventory\_id

INNER JOIN film AS f ON i.film\_id = f.film\_id

INNER JOIN film\_category AS fc ON f.film\_id = fc.film\_id

INNER JOIN category AS cg ON fc.category\_id = cg.category\_id

INNER JOIN payment AS p ON r.rental\_id = p.rental\_id;

--Part E: Procedure--

--Creates trigger function--

CREATE OR REPLACE FUNCTION summary\_trigger\_function()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM summary;

INSERT INTO summary

SELECT film\_category, COUNT(rental\_id)

FROM detailed

GROUP BY film\_category;

RETURN NEW;

END;

$$;

-- Part E: Creates Trigger--

CREATE TRIGGER update\_summary

AFTER INSERT

ON

Detailed

FOR EACH STATEMENT

EXECUTE PROCEDURE summary\_trigger\_function();

--Part F: Stored Procedures--

CREATE OR REPLACE PROCEDURE refresh\_tables()

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM Detailed;

DELETE FROM Summary;

INSERT INTO Detailed(

rental\_id,

customer\_id,

film\_id,

category\_id,

title,

film\_category,

payment\_id,

rental\_month

)

SELECT

r.rental\_id,

r.customer\_id,

f.film\_id,

cg.category\_id,

f.title,

cg.name,

p.payment\_id,

r.rental\_date

FROM rental AS r

INNER JOIN inventory AS i ON r.inventory\_id = i.inventory\_id

INNER JOIN film AS f ON i.film\_id = f.film\_id

INNER JOIN film\_category AS fc ON f.film\_id = fc.film\_id

INNER JOIN category AS cg ON fc.category\_id = cg.category\_id

INNER JOIN payment AS p ON r.rental\_id = p.rental\_id;

INSERT INTO summary

SELECT

film\_category,

COUNT(rental\_id)

FROM detailed

GROUP BY film\_category;

END; $$;

CALL refresh\_tables();

--PART F TEST CALL refresh\_tables();--

SELECT COUNT(\*) FROM detailed;

DELETE FROM detailed WHERE film\_category = 'New';

SELECT COUNT(\*) FROM detailed;

CALL refresh\_tables();

SELECT COUNT(\*) FROM detailed;