**WHAT FILM Category Is More Profitable**

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.   
  
 My project today is about creating a detailed business report that can be created using the DVD database. The question I will be using is, “What Film Category Is More Profitable” and I will be answering that with the provided data from the dataset. I believe with this information the DVD rental business can benefit by prioritizing which category of movie is more profitable. Not only will they gain more profit but also more customers along the way.

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

Detailed\_Rental\_Transactions:

- rental\_id

- rental\_date

- customer\_id

- customer\_name

- film\_id

- film\_title

- category\_id

- category\_name

- sale\_amount

Category\_Summary:

- category\_id

- category\_name

- total\_rentals

- total\_sales\_amount

2. Describe the types of data fields used for the report.  
  
Detailed\_Rental\_Transactions:

- `rental\_id`: Unique identifier for each rental transaction. INT

- `rental\_date`: Date when the rental occurred. DATE

- `customer\_id`: Identifier for the customer who rented the film. INT

- `customer\_name`: Concatenated field showing the customer's first and last names. VARCHAR

- `film\_id`: Identifier for the film rented. INT

- `film\_title`: Title of the film rented. VARCHAR

- `category\_id`: Identifier for the category of the film. INT

- `category`: Name of the category of the film. VARCHAR

- `sale\_amount`: Rental rate or sale amount of the film. INT

Category\_Profits:

- `category\_id`: Identifier for the category. INT

- `category\_name`: Name of the category. VARCHAR

- `total\_rentals`: Total number of rentals within each category. INT

- `total\_sales\_amount`: Total sales amount generated from rentals within each category. INT

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.  
  
FILM - Summary Table  
PAYMENT - Detailed Table  
CATEGORY - Summary Table  
FILM\_CATEGORY - Detailed Table  
RENTAL - Detailed Table

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).  
  
Category\_name will be used for this, We will be creating a transformation to make sure that the category is uppercased.  
  
Transforming category\_name to uppercase ensures that all category names are presented in a uniform format. This is especially useful when dealing with reports or summaries where case inconsistencies can lead to confusion or misinterpretation.

5. Explain the different business uses of the detailed table section and the summary table section of the report.   
  
**Detailed Table Section:**

**Uses:**

1. **Transaction-level Analysis:** Provides detailed insights into individual rental transactions, aiding in customer behavior analysis, film performance evaluation, and operational efficiency.
2. **Data Validation:** Ensures data accuracy and integrity across rental operations, supporting customer service and compliance efforts.
3. **Customer Service:** Facilitates personalized customer interactions by tracking rental histories and preferences.
4. **Compliance and Reporting:** Provides detailed data for regulatory compliance and internal reporting requirements.  
     
   **Summary Table Section:**

**Uses:**

1. **Strategic Decision Making:** Aggregates data to highlight rental trends and profitability by film category, aiding in strategic planning and investment decisions.
2. **Executive Reporting:** Presents key performance indicators in a concise format for executive-level decision-making.
3. **Marketing Insights:** Informs targeted marketing campaigns based on popular film categories and customer preferences.
4. **Operational Efficiency:** Supports inventory management, pricing strategies, and resource allocation based on category-specific rental patterns.

6. Explain how frequently your report should be refreshed to remain relevant to stakeholders.  
  
 Determining the best refresh rate for our DVD rental business report relies on balancing operational dynamics, business needs, and industry trends. I believe refreshing the data every week would benefit the business by having plenty of data to parse through and judge the upcoming week on what to focus on and what isn't being rented out as much.   
  
B. Provide original code for function(s) in text format that perform the transformation(s) you identified in part A4.

| CREATE OR REPLACE FUNCTION format\_category\_name(name TEXT) RETURNS TEXT AS $$   BEGIN    RETURN UPPER(name); END;  $$ LANGUAGE plpgsql; |
| --- |

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

| CREATE TABLE Detailed\_Rental\_Transactions AS SELECT   r.rental\_id,  r.rental\_date,  c.customer\_id,  CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,  f.film\_id,  f.title AS film\_title,  cat.category\_id AS category\_id,  format\_category\_name(cat.name) AS category\_name,  f.rental\_rate AS sale\_amount FROM   rental r JOIN   inventory i ON r.inventory\_id = i.inventory\_id JOIN   film f ON i.film\_id = f.film\_id JOIN   film\_category fc ON f.film\_id = fc.film\_id JOIN   category cat ON fc.category\_id = cat.category\_id JOIN   customer c ON r.customer\_id = c.customer\_id; |
| --- |

Summarized Table

| Create Table Category\_Profits AS Select  cat.category\_id,   format\_category\_name(cat.name) AS category\_name,  COUNT(\*) AS total\_rentals,   SUM(f.rental\_rate) AS total\_sales\_amount From  rental r Join  inventory i ON r.inventory\_id = i.inventory\_id Join  film f ON i.film\_id = f.film\_id Join  film\_category fc ON f.film\_id = fc.film\_id Join  category cat ON fc.category\_id = cat.category\_id Group BY  cat.category\_id, cat.name Order By  total\_sales\_amount DESC; |
| --- |

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.

| SELECT   r.rental\_id,  r.rental\_date,  c.customer\_id,  CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,  f.film\_id,  f.title AS film\_title,  cat.category\_id AS category\_id,  format\_category\_name(cat.name) AS category\_name,  f.rental\_rate AS sale\_amount FROM   rental r JOIN   inventory i ON r.inventory\_id = i.inventory\_id JOIN   film f ON i.film\_id = f.film\_id JOIN   film\_category fc ON f.film\_id = fc.film\_id JOIN   category cat ON fc.category\_id = cat.category\_id JOIN   customer c ON r.customer\_id = c.customer\_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 OR REPLACE FUNCTION Update\_Category\_Profits()  RETURNS TRIGGER AS $$  BEGIN  INSERT INTO Category\_Profits (category\_id, category\_name, total\_rentals, total\_sales\_amount)  VALUES (  NEW.category\_id,  format\_category\_name(NEW.category),  1,  NEW.sale\_amount  )  ON CONFLICT (category\_id)  DO UPDATE  SET  total\_rentals = Category\_Profits.total\_rentals + 1,  total\_sales\_amount = Category\_Profits.total\_sales\_amount + EXCLUDED.total\_sales\_amount;  RETURN NEW;  END;  $$  LANGUAGE plpgsql; |
| --- |

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 PROCEDURE Refresh\_Rental\_Data() LANGUAGE plpgsql AS $$   BEGIN   TRUNCATE TABLE Detailed\_Rental\_Transactions;   INSERT INTO Detailed\_Rental\_Transactions (rental\_id, rental\_date, customer\_id, customer\_name, film\_id, film\_title, category\_id, category, sale\_amount)  SELECT   r.rental\_id,  r.rental\_date,  c.customer\_id,  CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,  f.film\_id,  f.title AS film\_title,  cat.category\_id AS category\_id,  format\_category\_name(cat.name) AS category,  f.rental\_rate AS sale\_amount  FROM   rental r  JOIN   inventory i ON r.inventory\_id = i.inventory\_id  JOIN   film f ON i.film\_id = f.film\_id  JOIN   film\_category fc ON f.film\_id = fc.film\_id  JOIN   category cat ON fc.category\_id = cat.category\_id  JOIN   customer c ON r.customer\_id = c.customer\_id;    TRUNCATE TABLE Category\_Profits;   INSERT INTO Category\_Profits (category\_id, category\_name, total\_rentals, total\_sales\_amount)  SELECT  cat.category\_id,   format\_category\_name(cat.name) AS category,  COUNT(\*) AS total\_rentals,   SUM(f.rental\_rate) AS total\_sales\_amount  FROM  rental r  JOIN   inventory i ON r.inventory\_id = i.inventory\_id  JOIN   film f ON i.film\_id = f.film\_id  JOIN   film\_category fc ON f.film\_id = fc.film\_id  JOIN   category cat ON fc.category\_id = cat.category\_id  GROUP BY  cat.category\_id, cat.name  ORDER BY  total\_sales\_amount DESC; END; $$; |
| --- |

1. Identify a relevant job scheduling tool that can be used to automate the stored procedure.  
  
PGCron/ PGAgent, pgcron will refresh data and scheduling tools.

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 did not use any external or internal sources for this assignment.