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.

A report on which film categories have the highest revenue each month would be useful for determining which categories should be stocked and rented more for higher for revenue.

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

For the detailed table:

Table: category

Fields: category\_id, name – to identify categories.

Table: film\_category

Fields: film\_id, category\_id – to identify categories for each film sold.  
Table: inventory

Fields: inventory\_id, film\_id – to identify inventory id for each film sold.

Table: payment

Fields: payment\_id, rental\_id, amount, payment\_date – payment\_id will link rental\_id with amount and payment\_date

Table: rental

Fields: rental\_id, inventory\_id – links rental\_id with inventory\_id.

For the summary table:

payment\_date will filter all payment ids within a month, which can sum the amount of revenue for each category by linking rental\_id, inventory\_id, film\_id, and category\_id throughout the various tables.

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

category\_id: SMALLINT – Number of categories will most likely always be small.

name: VARCHAR (25) – New categories could be added as strings.

payment\_id, film\_id, inventory\_id, rental\_id: INT – Integer size will be suitable for number of positive ids in these categories.

amount: NUMERIC (5,2) – 5-digit dollar may be necessary for complications with rental overdue fees.

payment\_date: TIMESTAMP – Will be transformed to time zone for detailed table.

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.

Table: category

Fields: category\_id, name – to identify categories.

Table: film\_category

Fields: film\_id, category\_id – to identify categories for each film sold.  
Table: inventory

Fields: inventory\_id, film\_id – to identify inventory id for each film sold.

Table: payment

Fields: payment\_id, rental\_id, amount, payment\_date – payment\_id will link rental\_id with amount and payment\_date

Table: rental

Fields: rental\_id, inventory\_id – links rental\_id with inventory\_id.

For the summary table:

payment\_date will filter all payment ids within a month, which can sum the amount of revenue for each category by linking rental\_id, inventory\_id, film\_id, and category\_id throughout the various tables.

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).

payment\_date will be transformed to timestamp with time zone data type to allow for more locality granularity and consistency when pulling the summary report across different time zones.

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

The detailed table section will provide the raw data needed to aggregate specific data needed for the summary table, which answers a specific business question.

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

The summary table will be used in a monthly report, so will be refreshed monthly at end of month. Any changes to fields (i.e., adding new categories) will be captured in the detailed tables and updated accordingly.

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

CREATE OR REPLACE FUNCTION convert\_to\_payment\_timezone (input\_timestamp TIMESTAMP)

RETURNS TIMESTAMP WITH TIME ZONE

AS $$

BEGIN

RETURN input\_timestamp AT TIME ZONE 'UTC' AT TIME ZONE 'America/Chicago';

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.

--Create detailed table.

CREATE TABLE Detailed\_Table (

payment\_id INT,

amount NUMERIC (5,2),

payment\_date TIMESTAMP,

film\_id SMALLINT,

inventory\_id INT,

rental\_id INT,

category\_id SMALLINT,

category\_name VARCHAR (25),

payment\_timezone TIMESTAMP WITH TIME ZONE

);

--Create summary report table.

CREATE TABLE Summary\_Report (

category\_name VARCHAR (25),

total\_amount NUMERIC (8,2)

);

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.

--Create the function for raw data extraction. Transform payment\_date data type.

CREATE OR REPLACE FUNCTION extract\_raw\_data()

RETURNS TABLE (

payment\_id INT,

amount NUMERIC (5,2),

payment\_date TIMESTAMP,

film\_id SMALLINT,

inventory\_id INT,

rental\_id INT,

category\_id SMALLINT,

category\_name VARCHAR (25),

payment\_timezone TIMESTAMP WITH TIME ZONE

) AS $$

BEGIN

RETURN QUERY

SELECT

p.payment\_id,

p.amount,

p.payment\_date,

fc.film\_id,

r.inventory\_id,

p.rental\_id,

fc.category\_id,

c.name AS category\_name,

convert\_to\_payment\_timezone(p.payment\_date) AS payment\_timezone

FROM

payment p

INNER JOIN rental r ON p.rental\_id = r.rental\_id

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

INNER JOIN film\_category fc ON i.film\_id = fc.film\_id

INNER JOIN category c ON fc.category\_id = c.category\_id;

END;

$$ LANGUAGE plpgsql;

--Extract data into detailed table.

INSERT INTO Detailed\_Table (payment\_id, amount, payment\_date, film\_id, inventory\_id, rental\_id, category\_id, category\_name, payment\_timezone)

SELECT \* FROM extract\_raw\_data();

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 the trigger function

CREATE OR REPLACE FUNCTION update\_summary\_report()

RETURNS TRIGGER AS $$

BEGIN

-- Check if the current operation is an INSERT

IF TG\_OP = 'INSERT' THEN

-- Calculate the total amount for each category and update the Summary\_Report table

INSERT INTO Summary\_Report (category\_name, total\_amount)

SELECT

NEW.category\_name,

SUM(amount) AS total\_amount

FROM

Detailed\_Table

WHERE

category\_name = NEW.category\_name

GROUP BY

NEW.category\_name;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

-- Create the INSERT trigger on the Detailed\_Table

CREATE TRIGGER update\_summary\_trigger

AFTER INSERT ON Detailed\_Table

FOR EACH ROW

EXECUTE FUNCTION update\_summary\_report();

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 the stored procedure to refresh data.

CREATE OR REPLACE PROCEDURE refresh\_data()

LANGUAGE plpgsql

AS $$

BEGIN

-- Clear the contents of the Summary\_Report table

DELETE FROM Summary\_Report;

-- Clear the contents of the Detailed\_Table

DELETE FROM Detailed\_Table;

-- Perform the raw data extraction using the function and populate Detailed\_Table

INSERT INTO Detailed\_Table (payment\_id, amount, payment\_date, film\_id, inventory\_id, rental\_id, category\_id, category\_name, payment\_timezone)

SELECT \* FROM extract\_raw\_data();

-- Recalculate the data and populate the Summary\_Report table

INSERT INTO Summary\_Report (category\_name, total\_amount)

SELECT

category\_name,

SUM(amount) AS total\_amount

FROM

Detailed\_Table

GROUP BY

category\_name;

END;

$$;

-- Call the stored procedure to refresh data.

CALL refresh\_data();

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

Task Scheduler is default on Windows OS and can be used to automate the SQL queries.

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.

No sources were used to support my submission.