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

Planning staff availability is a hard issue to tackle. A report that shows which day the most rentals occur can assist in planning on when to schedule staff. It also can indicate when we may need to hire more staff to be available on certain days.

**1. Describe the data used for the report.**

The report will use rental date and return date to count how many transactions are on each day of the week. It also uses the staff id to link each rental and return to a store so the data can be broken up by store.

**2. 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 report will use the rental table, the staff table, the address table, and the store table

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

Detailed Section

1. Rental\_Id
2. Rental\_Date
3. Return\_date
4. Staff\_id
5. Store\_id
6. Address\_id
7. Address

Summary Section

1. Store Address
2. Monday Transactions
3. Tuesday Transactions
4. Wednesday Transactions
5. Thursday Transactions
6. Friday Transactions
7. Saturday Transactions
8. Sunday Transactions

**4. Identify one field in the detailed section that will require a custom transformation and explain why it should be transformed. For example, you might translate a field with a value of ‘N’ to ‘No’ and ‘Y’ to ‘Yes’.**

One field that will need to be transformed will be the rental\_date and return\_date. When extracted they are a timestamp. This will need to be formatted into a day of the week, like “Monday”.

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

The detailed section will allow the stakeholder to dive deeper to see a hourly trend, or can see if a staff member is more productive then another. It will store every transaction for every store.

The Summary Section just has the count of rentals and returns per day per store. Good for an overview of where store should put effort in having staff.

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

The report should be updated monthly to quarterly to keep updated the latest staffing requirements for the stores. There is also a potential to have this report to only capture the last year or last quarters data so that shorter term trends may be exposed.

-- B. SQL code that creates the tables to hold your report sections.

DROP TABLE IF EXISTS detailed;

CREATE TABLE detailed (

rental\_id INT,

rental\_date TIMESTAMP,

return\_date TIMESTAMP,

address VARCHAR(50)

);

DROP TABLE IF EXISTS summary;

CREATE TABLE summary (

store\_address VARCHAR(50),

monday\_count INT,

tuesday\_count INT,

wednesday\_count INT,

thursday\_count INT,

friday\_count INT,

saturday\_count INT,

sunday\_count INT

);

-- C. Write a SQL query that will extract the raw data needed for the Detailed section of your report from the source database and verify the data’s accuracy.

INSERT INTO detailed(

rental\_id,

rental\_date,

return\_date,

address

)

SELECT

r.rental\_id, r.rental\_date, r.return\_date, a.address

FROM rental AS r

INNER JOIN staff AS sf ON sf.staff\_id = r.staff\_id

INNER JOIN store AS st ON st.store\_id = sf.store\_id

INNER JOIN address AS a ON a.address\_id = st.address\_id;

SELECT \* FROM detailed; -- Verfiy that the correct data was imported

-- D. Write code for functions that peform the trnaformations you identified in part A4--

CREATE OR REPLACE FUNCTION refresh\_summary()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

BEGIN

INSERT INTO summary(store\_address)

SELECT DISTINCT address

FROM detailed;

PERFORM day\_of\_the\_week\_count('28 MySQL Boulevard');

PERFORM day\_of\_the\_week\_count('47 MySakila Drive');

RETURN NEW;

END;$$;

CREATE OR REPLACE FUNCTION day\_of\_the\_week\_count(in\_address char(50))

RETURNS VOID

LANGUAGE plpgsql

AS $$

DECLARE

sun\_rent int;

mon\_rent int;

tue\_rent int;

wed\_rent int;

thu\_rent int;

fri\_rent int;

sat\_rent int;

sun\_retu int;

mon\_retu int;

tue\_retu int;

wed\_retu int;

thu\_retu int;

fri\_retu int;

sat\_retu int;

BEGIN

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 0 INTO sun\_rent;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 1 INTO mon\_rent;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 2 INTO tue\_rent;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 3 INTO wed\_rent;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 4 INTO thu\_rent;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 5 INTO fri\_rent;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 6 INTO sat\_rent;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 0 INTO sun\_retu;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 1 INTO mon\_retu;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 2 INTO tue\_retu;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 3 INTO wed\_retu;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 4 INTO thu\_retu;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 5 INTO fri\_retu;

SELECT count(\*) FROM detailed

WHERE address = in\_address

AND EXTRACT(DOW from rental\_date) = 6 INTO sat\_retu;

UPDATE summary

SET

sunday\_count = sun\_rent + sun\_retu,

monday\_count = mon\_rent + mon\_retu,

tuesday\_count = tue\_rent + tue\_retu,

wednesday\_count = wed\_rent + wed\_retu,

thursday\_count = thu\_rent + thu\_retu,

friday\_count = fri\_rent + fri\_retu,

saturday\_count = sat\_rent + sat\_retu

WHERE store\_address = in\_address;

END;

$$;

--E. Write a SQL code 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 TRIGGER refresh\_summary

AFTER INSERT ON detailed

FOR EACH STATEMENT

EXECUTE FUNCTION refresh\_summary();

– 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.--

CREATE OR REPLACE PROCEDURE refresh\_detailed()

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM detailed;

DELETE FROM summary;

INSERT INTO detailed(

rental\_id,

rental\_date,

return\_date,

address

)

SELECT

r.rental\_id, r.rental\_date, r.return\_date, a.address

FROM rental AS r

INNER JOIN staff AS sf ON sf.staff\_id = r.staff\_id

INNER JOIN store AS st ON st.store\_id = sf.store\_id

INNER JOIN address AS a ON a.address\_id = st.address\_id;

END;$$;

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

You can call the procedure at any time to refresh the data in both tables. A monthly or quarterly would capture enough new data to make a noticeable difference in the data.

--How to call the procedure

-- CALL refresh\_detailed()

-- To view the resultes

-- SELECT \* FROM detailed;

-- SELECT \* FROM summary;

Link to Panopto Video:

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=93541ee1-52e6-45f9-b244-ae8101840b9b>