**PART A: Describe one real-world business report that can be created from the attached Data Sets.**

Employee productivity is vital to the health and success of a business. Insuring employees are working appropriately for the company is important, but so is employee happiness. One way of encouraging and rewarding productivity is through the use of bonuses when employees distinguish themselves from the rest. The business report I’d like to create will use the database to show which employee rents the most DVDs and then that employee will receive recognition and a bonus.

**PART A1: Describe the data used for the report.**

There are two main data points that will be required for this report. First I will need the data for the DVD rentals themselves and then the data from the employee table. This data will be compared to determine the most productive employee.

**PART A2: Identify two or more specific tables from the given data set that provide the data necessary for the detailed and summary sections.**

I will only need two tables in order to create this report. The first table is the “rental” table which contains the specific rental information and which employee handled the transaction. I will also need the “staff” table in order to identify the employee responsible for a given rental through their employee ID number.

**PART A3: Identify the specific fields that will be included in the detailed and summary sections of the report.**

The detailed report will include the employee’s name and all relevant contact information. I will use all of the data from the “staff” table with the exception of private information that’s not necessary such as their username and password. The summary report will only need the employee’s name, their employee ID, and the number of rentals.

**PART A4: Identify one field in the detailed section that requires a custom transformation and why it should be transformed.**

There are two fields primarily that will need to be transformed for readability. I will concatenate the first and last names together in one full name. I will also add up all the rentals and group them by employee ID to see who has the most rentals.

**PART A5: Explain the different business uses of the detailed and the summary section of the report.**

There can be multiple uses of these two tables and reports that I will create with this assignment. The summary report can be used in order to get the raw information as to which employee is the most productive with regards to DVD rentals. The detailed table can be analyzed further by seeing what types of DVDs that specific employee rents the most. This information can be used to place employees at stores where they may be more successful.

**PART A6: Explain how frequently your report should be refreshed to remain relevant to stakeholders.**

This procedure can be run every month or quarter depending on how often a company would like to hand out bonuses. This can be completed by the database administrator calling the function on whatever the desired schedule is to insure the data freshness. Additionally, a script can be written in order to run the function automatically.

**PART B: Create the tables needed to hold the information**

This first table will show every single rental that an employee is responsible for.

CREATE TABLE detailed(

rental\_id INT PRIMARY KEY,

staff\_id INT,

first\_name VARCHAR(40),

last\_name VARCHAR(40),

email VARCHAR(100),

store\_id INT,

rental\_date TIMESTAMP

);

The second table is a much more concise version that only gives the employee’s name and total number of rentals.

CREATE TABLE summary(

staff\_id INT PRIMARY KEY,

full\_name VARCHAR(80),

total\_rentals INT

);

**PART C: Extract the raw data into the new tables**

The first statement will populate the detailed table.

INSERT INTO detailed(rental\_id, staff\_id, first\_name, last\_name, email, store\_id, rental\_date)

SELECT r.rental\_id, s.staff\_id, s.first\_name, s.last\_name, s.email, s.store\_id, r.rental\_date

FROM rental AS r

INNER JOIN staff AS s ON s.staff\_id = r.staff\_id;

This statement will populate the summary table from the information in the detailed table.

INSERT INTO summary(

SELECT staff\_id,

CONCAT(first\_name, ‘ ‘, last\_name) AS full\_name,

COUNT(staff\_id)

FROM detailed

GROUP BY staff\_id, full\_name

HAVING COUNT(staff\_id) > 1

ORDER BY COUNT(staff\_id) DESC

LIMIT 10

);

**PART D: Write a function to perform a transformation of the data.**

This function will clear out the summary table and re-populate it. It also performs two transformations. First in concatenates the first and last names into one full name for readability. It also adds up the number of rentals an employee is responsible for.

CREATE FUNCTION refresh\_summary\_function()

RETURNS TRIGGER

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM summary;

INSERT INTO summary(

SELECT staff\_id,

CONCAT(first\_name, ‘ ‘, last\_name) AS full\_name,

COUNT(staff\_id)

FROM detailed

GROUP BY staff\_id, full\_name

HAVING COUNT(staff\_id) > 1

ORDER BY COUNT(staff\_id) DESC

LIMIT 10

);

RETURN NEW;

END; $$

**PART E: Create a trigger that will continually update the summary table as data is added to the detailed table.**

CREATE TRIGGER refresh\_summary\_table

AFTER INSERT ON detailed

FOR EACH STATEMENT

EXECUTE PROCEDURE refresh\_summary\_function();

**PART F: Create a stored procedure to refresh data in both tables.**

This procedure can be run every month or quarter depending on how often a company would like to hand out bonuses. This can be completed by the database administrator calling the function on whatever the desired schedule is to insure the data freshness. Additionally, a script can be written in order to run the function automatically.

CREATE PROCEDURE refresh()

LANGUAGE plpgsql

AS $$

BEGIN

DELETE FROM detailed;

INSERT INTO detailed(rental\_id, staff\_id, first\_name, last\_name, email, store\_id, rental\_date)

SELECT r.rental\_id, s.staff\_id, s.first\_name, s.last\_name, s.email, s.store\_id, r.rental\_date

FROM rental AS r

INNER JOIN staff AS s ON s.staff\_id = r.staff\_id;

DELETE FROM summary;

INSERT INTO summary(

SELECT staff\_id,

CONCAT(first\_name, ‘ ‘, last\_name) AS full\_name,

COUNT(staff\_id)

FROM detailed

GROUP BY staff\_id, full\_name

HAVING COUNT(staff\_id) > 1

ORDER BY COUNT(staff\_id) DESC

LIMIT 10

);

END; $$

**PART H: Web Sources**

For this project I did not use any web sources with the exception of the database required for this project.

**PART I: Sources**

No outsides sources were used for this project. As such, there are no in-text citations and there isn’t a reference list.