DATABASE MANAGEMENT

**1.** Write an SQL query to show the top 10 best-selling products by total Qty (i.e., quantity).

**ANSWER:**

SELECT PD."ProductKey", PD."ProductName", SUM("Qty") SumQty FROM "Products" PD, "RetailSales" RS WHERE RS."ProductKey" = PD."ProductKey" GROUP BY PD."ProductKey", PD."ProductName" ORDER BY SumQty DESC LIMIT 10

A screenshot of a computer

Description automatically generated

**2.** Write an SQL query to show the total gross sales revenues each department makes by selling each brand.

**ANSWER:**

SELECT PD."Dept" AS DEPARTMENT, PD."Brands" AS BRANDS,SUM(RS."Grosssales") AS GrossSale  
FROM "Products" PD  
JOIN "RetailSales" RS ON RS."ProductKey" = pd."productkey"

GROUP BY PD."Dept", PD."Brands"

A screenshot of a computer

Description automatically generated

**3.** Write an SQL query to show the total net sales revenues each department makes in-store and online respectively.

**ANSWER:**

SELECT pd."Dept",

SUM(CASE WHEN s."OLStore" = 'In-Store' THEN rs."NetSales" ELSE 0 END) AS total\_instore\_revenue,

SUM(CASE WHEN s."OLStore" = 'Online' THEN rs."NetSales" ELSE 0 END) AS total\_online\_revenue

FROM "Products" pd

JOIN "RetailSales" rs ON rs."ProductKey" = pd."productkey"

JOIN "Stores" s ON rs."StoreKey" = s."StoreKey"

GROUP BY pd."Dept"

ORDER BY pd."Dept";

A screenshot of a computer

Description automatically generated

4. Write an SQL query to show how many customers are served by all cash registers of Store #1 per hour? (Note: StoreKey should be 1 because only Store #1’s cash registers are of analytical interest. The fact table records data from each scan. So, you should not use SUM(Qty) because the question asks the total number of customers per hour, not the total number of scanned products per hour. Hint: The degenerate dimension can help!)

**ANSWER:**

SELECT T.Hour AS Hour,

COUNT(DISTINCT RS.CustomerKey) AS CustomersPerHour

FROM Time AS T

LEFT JOIN RetailSales AS RS

ON T.TimeKey = RS.TimeKey

WHERE RS.StoreKey = 1

GROUP BY T.Hour

A screenshot of a computer

Description automatically generated

**5.** Use “Graph Visualiser” from PostgreSQL to visualize the results of question 4. Make a screenshot of the chart and save it into your submission.

**ANSWER:**

SELECT T.Hour AS Hour,

COUNT(DISTINCT RS.CustomerKey) AS CustomersPerHour

FROM Time AS T

LEFT JOIN RetailSales AS RS

ON T.TimeKey = RS.TimeKey

WHERE RS.StoreKey = 1

GROUP BY T.Hour

A screenshot of a computer

Description automatically generated

**6.** Write an SQL query to show how many customers are served by Store #1 in each day of a week. (Note: The SQL query should show the total number of customers served in Store #1 on Monday, Tuesday, ... Sunday. The number of customers and week day information must be in the same table. You should not write a query for Monday, and then another query for Tuesday, and then another query for Wednesday, etc. Using just one query to show the number of customers served by Store #1 in each day of a week is required).

**ANSWER:**

SELECT D.dayofweek, RS . storekey ,

COUNT(DISTINCT RS.customerkey) AS CustomersPerDay

FROM retailsales AS RS

JOIN dates AS D

ON RS.datekey = D.dateKey

WHERE RS.storekey = 1

GROUP BY D.dayofweek , RS . storekey

A screenshot of a computer

Description automatically generated

7. Write an SQL query to show what are the best-selling products by total Qty in the fourth quarter in the “Beauty&Health” department.

**ANSWER:**

SELECT PD.ProductName, D.quarter ,

SUM(RS.qty) AS TotalQty

FROM products AS PD

JOIN retailsales AS RS

ON PD.productkey = RS.productkey

JOIN dates AS D

ON RS.datekey = D.datekey

WHERE D.quarter = 4 AND PD.dept = 'Beauty&Health'

GROUP BY PD.productname, D.quarter

ORDER BY TotalQty DESC;

A screenshot of a computer

Description automatically generated

**8.** Write an SQL query to show whether there is a monthly increasing total GrossSales trend in the “Electronics” department.

**ANSWER:**

SELECT D.monthofyear,

SUM(RS.grosssales) AS GrossSales,

LAG(SUM(RS.grosssales)) OVER (ORDER BY D.monthofyear) AS PreviousMonthGrossSales,

CASE

WHEN SUM(RS.grosssales) > LAG(SUM(RS.grosssales)) OVER (ORDER BY D.monthofyear)

THEN 'Increasing'

WHEN SUM(RS.grosssales) < LAG(SUM(RS.grosssales)) OVER (ORDER BY D.monthofyear)

THEN 'Decreasing'

ELSE 'No Change'

END AS Trend

FROM retailsales AS RS

JOIN products AS P

ON RS.productkey = P.productkey

JOIN dates AS D

ON RS.datekey = D.datekey

WHERE P.dept = 'Electronics'

GROUP BY D.monthofyear

ORDER BY D.monthofyear;

A screenshot of a computer

Description automatically generated

SUBMITTED BY:

NAME: CWID

K. AMUKTHA 50339541

SAMEEN SIDDIQUA 50338823

JILLELLA REVATHI 50358787