ASSIGNMENT 2 SOLUTION

PART 1

a) SELECT COUNT(DISTINCT Customer\_ID) AS Total\_Customers

FROM order\_level

GROUP BY Customer\_ID

HAVING COUNT(Transaction\_number) > 5;

b)-1

SELECT Store\_ID, COUNT(Order\_ID) AS Order\_Count

FROM order\_level

GROUP BY Store\_ID

ORDER BY Order\_Count DESC

LIMIT 1;

2-SELECT Store\_ID, COUNT(DISTINCT Customer\_ID) AS Customer\_Count

FROM order\_level

GROUP BY Store\_ID

ORDER BY Customer\_Count DESC

LIMIT 1;

c) SELECT s.Market, AVG(o.Basket\_Amount) AS Average\_Order\_Value

FROM order\_level o

JOIN store\_level s ON o.Store\_ID = s.Store\_ID

GROUP BY s.Market

ORDER BY Average\_Order\_Value DESC

LIMIT 1;

d) SELECT s.Cohort, COUNT(o.Order\_ID) AS Order\_Count

FROM order\_level o

JOIN store\_level s ON o.Store\_ID = s.Store\_ID

GROUP BY s.Cohort

ORDER BY Order\_Count DESC

LIMIT 1;

e) SELECT COUNT(DISTINCT Store\_ID) AS New\_Stores

FROM order\_level

WHERE Transaction\_Week >= (SELECT MAX(Transaction\_Week) - 8 FROM order\_level);

f) SELECT Transaction\_Week, COUNT(DISTINCT Customer\_ID) AS Total\_Customers

FROM order\_level

GROUP BY Transaction\_Week

ORDER BY Transaction\_Week;

PART 2

A) WITH last\_transaction AS (

SELECT Customer\_ID, MAX(Transaction\_Week) AS Last\_Transaction\_Week

FROM order\_level

GROUP BY Customer\_ID

),

total\_customers AS (

SELECT COUNT(DISTINCT Customer\_ID) AS Total\_Customers

FROM order\_level

),

churned\_customers AS (

SELECT COUNT(Customer\_ID) AS Churned\_Customers

FROM last\_transaction

WHERE Last\_Transaction\_Week <= (SELECT MAX(Transaction\_Week) - 6 FROM order\_level)

)

SELECT (churned\_customers.Churned\_Customers::float / total\_customers.Total\_Customers) AS Churn\_Rate

FROM churned\_customers, total\_customers;

B) WITH first\_transactions AS (

SELECT Customer\_ID, Store\_ID, MIN(Transaction\_Week) AS First\_Transaction\_Week

FROM order\_level

GROUP BY Customer\_ID, Store\_ID

)

SELECT Store\_ID, COUNT(DISTINCT Customer\_ID) AS New\_Customers

FROM first\_transactions

WHERE First\_Transaction\_Week BETWEEN 201644 AND 201648

GROUP BY Store\_ID

ORDER BY New\_Customers DESC

LIMIT 1;

C) WITH order\_gaps AS (

SELECT Customer\_ID, Transaction\_Week,

LAG(Transaction\_Week, 1) OVER (PARTITION BY Customer\_ID ORDER BY Transaction\_Week) AS Previous\_Week

FROM order\_level

)

SELECT Transaction\_Week,

AVG(Transaction\_Week - Previous\_Week) AS Average\_Order\_Gap

FROM order\_gaps

WHERE Previous\_Week IS NOT NULL

GROUP BY Transaction\_Week

ORDER BY Transaction\_Week;

D) SELECT Transaction\_Week, COUNT(Order\_ID) AS Number\_Of\_Orders

FROM order\_level

GROUP BY Transaction\_Week

ORDER BY Transaction\_Week;

E) WITH customer\_transactions AS (

SELECT Customer\_ID, Market, Transaction\_Week,

LAG(Transaction\_Week, 1) OVER (PARTITION BY Customer\_ID, Market ORDER BY Transaction\_Week) AS Previous\_Week

FROM order\_level o

JOIN store\_level s ON o.Store\_ID = s.Store\_ID

WHERE s.Market IN ('Huntsville', 'Baltimore')

)

SELECT Customer\_ID, Market

FROM customer\_transactions

WHERE Transaction\_Week - Previous\_Week <= 7

GROUP BY Customer\_ID, Market;

PART 3

A) SELECT s.Market, o.Customer\_ID, COUNT(o.Order\_ID) AS Transaction\_Count

FROM order\_level o

JOIN store\_level s ON o.Store\_ID = s.Store\_ID

GROUP BY s.Market, o.Customer\_ID

ORDER BY s.Market, Transaction\_Count DESC;

B) SELECT s.Market, o.Customer\_ID, SUM(o.Basket\_Amount) AS Total\_Sales

FROM order\_level o

JOIN store\_level s ON o.Store\_ID = s.Store\_ID

GROUP BY s.Market, o.Customer\_ID

ORDER BY s.Market, Total\_Sales DESC;

C) WITH last\_transaction AS (

SELECT Customer\_ID, s.Market, MAX(o.Transaction\_Week) AS Last\_Transaction\_Week

FROM order\_level o

JOIN store\_level s ON o.Store\_ID = s.Store\_ID

GROUP BY Customer\_ID, s.Market

),

total\_customers AS (

SELECT s.Market, COUNT(DISTINCT o.Customer\_ID) AS Total\_Customers

FROM order\_level o

JOIN store\_level s ON o.Store\_ID = s.Store\_ID

GROUP BY s.Market

),

churned\_customers AS (

SELECT s.Market, COUNT(Customer\_ID) AS Churned\_Customers

FROM last\_transaction lt

JOIN store\_level s ON lt.Store\_ID = s.Store\_ID

WHERE Last\_Transaction\_Week <= (SELECT MAX(Transaction\_Week) - 6 FROM order\_level)

GROUP BY s.Market

)

SELECT tc.Market,

(cc.Churned\_Customers::float / tc.Total\_Customers) AS Churn\_Rate

FROM total\_customers tc

JOIN churned\_customers cc ON tc.Market = cc.Market

ORDER BY Churn\_Rate DESC;

D) SELECT Store\_ID, Transaction\_Week

FROM (

SELECT Store\_ID, Transaction\_Week, COUNT(Order\_ID) AS Order\_Count

FROM order\_level

GROUP BY Store\_ID, Transaction\_Week

) subquery

WHERE Order\_Count >= 5

ORDER BY Store\_ID, Transaction\_Week;

E) WITH active\_customers AS (

SELECT Transaction\_Week, Customer\_ID

FROM (

SELECT Transaction\_Week, Customer\_ID,

LAG(Transaction\_Week, 1) OVER (PARTITION BY Customer\_ID ORDER BY Transaction\_Week) AS Previous\_Week

FROM order\_level

) subquery

WHERE Transaction\_Week - Previous\_Week <= 6

)

SELECT Transaction\_Week, COUNT(DISTINCT Customer\_ID) AS Active\_Customers

FROM active\_customers

GROUP BY Transaction\_Week

ORDER BY Transaction\_Week;

F) WITH active\_customers AS (

SELECT Transaction\_Week, Customer\_ID

FROM (

SELECT Transaction\_Week, Customer\_ID,

LAG(Transaction\_Week, 1) OVER (PARTITION BY Customer\_ID ORDER BY Transaction\_Week) AS Previous\_Week

FROM order\_level

) subquery

WHERE Transaction\_Week - Previous\_Week <= 6

),

weekly\_transactions AS (

SELECT Transaction\_Week, COUNT(DISTINCT Customer\_ID) AS Transactions\_This\_Week

FROM order\_level

GROUP BY Transaction\_Week

)

SELECT ac.Transaction\_Week,

COUNT(DISTINCT ac.Customer\_ID) AS Active\_Customers,

wt.Transactions\_This\_Week,

ROUND((wt.Transactions\_This\_Week::float / COUNT(DISTINCT ac.Customer\_ID)) \* 100, 2) AS Activity\_Rate

FROM active\_customers ac

JOIN weekly\_transactions wt ON ac.Transaction\_Week = wt.Transaction\_Week

GROUP BY ac.Transaction\_Week, wt.Transactions\_This\_Week

ORDER BY ac.Transaction\_Week;

G) WITH weekly\_sales AS (

SELECT Store\_ID, Transaction\_Week, SUM(Basket\_Amount) AS Sales

FROM order\_level

GROUP BY Store\_ID, Transaction\_Week

),

weekly\_trend AS (

SELECT Store\_ID, Transaction\_Week, Sales,

LAG(Sales, 1) OVER (PARTITION BY Store\_ID ORDER BY Transaction\_Week) AS Previous\_Week\_Sales

FROM weekly\_sales

)