```
Q1  --Total Sales Per Invoice


SELECT "InvoiceNo",
       SUM("Quantity") OVER (PARTITION BY
"InvoiceNo") AS total_quantity,
       SUM("Quantity" * "UnitPrice") OVER (PARTITION
BY "InvoiceNo") AS total_sales
FROM "Online_Retail"
ORDER BY total_sales DESC;
```

```
 --Top 5 Customers by Total Spending

SELECT "CustomerID",
       SUM("Quantity" * "UnitPrice") OVER (PARTITION
BY "CustomerID", "InvoiceNo") AS
total_spending
FROM "Online_Retail"
ORDER BY total_spending desc
LIMIT 5;
```

```
 --  Number of Invoices per Customer

SELECT "CustomerID",
       COUNT( "InvoiceNo") OVER (PARTITION BY
"CustomerID") AS number_of_invoices
FROM "Online_Retail"
ORDER BY number_of_invoices DESC;
```

```sql
 --Ranking Customers by Total Purchase Amountsql

SELECT cus.*, RANK() OVER (ORDER BY
"rank_of_customers" DESC)
FROM (
  SELECT *, SUM("UnitPrice" * "Quantity") OVER
(PARTITION BY "CustomerID") AS
"rank_of_customers"
  FROM "Online_Retail"
) cus;

 --First Product Purchased Per Customer

SELECT "CustomerID",
       FIRST_VALUE("Description") OVER (PARTITION BY
"CustomerID" ORDER BY "InvoiceDate")
AS first_product
FROM "Online_Retail";


 -- Country with the Most Customers

SELECT "Country",
       COUNT( "CustomerID") OVER (PARTITION BY
"Country") AS customers_in_country
FROM "Online_Retail"
ORDER BY customers_in_country DESC; -- Monthly Sales
Trends
```

Q2

```sql
WITH LatestInvoiceDate AS (
    -- Get the most recent invoice date to use as the reference point
    SELECT MAX(CAST(InvoiceDate AS DATE)) AS ReferenceDate
    FROM online_retail
),

RFM AS (
    -- Calculate Recency, Frequency, and Monetary for each customer
    SELECT
        CustomerID,
        -- Recency: Difference between the last purchase date and the reference date
        (SELECT ReferenceDate FROM LatestInvoiceDate) - MAX(CAST(InvoiceDate AS DATE)) AS Recency,
        -- Frequency: Count of unique invoices for each customer
        COUNT(DISTINCT InvoiceNo) AS Frequency,
        -- Monetary: Total money spent by multiplying quantity by price
        SUM(Quantity * UnitPrice) AS Monetary
    FROM online_retail
    GROUP BY CustomerID
),
RFM_Scores AS (
    -- Assign scores for Recency, Frequency, and
Monetary
    SELECT
        CustomerID,
        Recency,
        Frequency,
        Monetary,
        -- Recency scoring based on predefined
thresholds
        CASE
            WHEN Recency <= 30 THEN 5
            WHEN Recency <= 60 THEN 4
            WHEN Recency <= 90 THEN 3
            WHEN Recency <= 180 THEN 2
            ELSE 1
        END AS RecencyScore,

        -- Frequency scoring based on thresholds
        CASE
            WHEN Frequency >= 50 THEN 5
            WHEN Frequency >= 20 THEN 4
            WHEN Frequency >= 10 THEN 3
            WHEN Frequency >= 5 THEN 2
            ELSE 1
        END AS FrequencyScore,

        -- Monetary scoring based on spending
thresholds
        CASE
            WHEN Monetary >= 500 THEN 5
            WHEN Monetary >= 200 THEN 4
            WHEN Monetary >= 100 THEN 3
            WHEN Monetary >= 50 THEN 2
            ELSE 1
        END AS MonetaryScore
    FROM RFM
),

SegmentScores AS (
    -- Calculate an average score for Frequency and
Monetary
    SELECT
        CustomerID,
        RecencyScore,
        (FrequencyScore + MonetaryScore) / 2 AS
FreqMonScore  -- Average of Frequency and
Monetary scores
    FROM RFM_Scores
),

CustomerSegmentation AS (
    -- Segment customers based on their scores
    SELECT
        CustomerID, RecencyScore, FreqMonScore,
        CASE
            WHEN RecencyScore = 5 AND FreqMonScore >=
4 THEN 'Champions'
            WHEN RecencyScore BETWEEN 3 AND 4 AND
FreqMonScore BETWEEN 2 AND 3 THEN
'Potential Loyalists'
            WHEN RecencyScore = 5 AND FreqMonScore =
2 THEN 'Potential Loyalists'
            WHEN RecencyScore BETWEEN 3 AND 5 AND
FreqMonScore BETWEEN 3 AND 5 THEN
'Loyal Customers'
```

```
            WHEN RecencyScore = 5 AND FreqMonScore =
1 THEN 'Recent Customers'
            WHEN RecencyScore BETWEEN 3 AND 4 AND
FreqMonScore = 1 THEN 'Promising'
            WHEN RecencyScore BETWEEN 2 AND 3 AND
FreqMonScore BETWEEN 2 AND 3 THEN
'Customers Needing Attention'
            WHEN RecencyScore BETWEEN 1 AND 2 AND
FreqMonScore BETWEEN 3 AND 5 THEN 'At
Risk'
            WHEN RecencyScore = 1 AND FreqMonScore
BETWEEN 4 AND 5 THEN 'Can’t Lose Them'
            WHEN RecencyScore = 1 AND FreqMonScore =
2 THEN 'Hibernating'
            ELSE 'Lost'
        END AS CustomerSegment
    FROM SegmentScores
)
 -- Final Output: Show the segmented customers
SELECT * FROM CustomerSegmentation;
```