1.
```sql
SELECT
  InvoiceNo,StockCode,Quantity,UnitPrice,
  (Quantity*UnitPrice) AS amount
FROM
  `customer-segmentation-373712.retail.sales`
```

2.
```sql
WITH
bills AS (
SELECT
  InvoiceNo,
  (Quantity*UnitPrice) AS amount
FROM
  `customer-segmentation-373712.retail.sales` )
SELECT
InvoiceNo,
SUM(amount) AS total
FROM
bills
GROUP BY
InvoiceNo
```

3.
```sql
SELECT
s.CustomerID,
DATE(MAX(s.InvoiceDate)) AS last_purchase_date,
DATE(MIN(s.InvoiceDate)) AS first_purchase_date,
COUNT(DISTINCT s.InvoiceNo) AS num_purchases,
SUM(b.total) AS monetary,
FROM
`customer-segmentation-373712.retail.sales` s
LEFT JOIN
`customer-segmentation-373712.retail.bill` b
ON
```

```sql
s.InvoiceNo=b.InvoiceNo
GROUP BY
CustomerID
```

4.

```sql
WITH cte AS
(SELECT
s.CustomerID,
DATE(MAX(s.InvoiceDate)) AS last_purchase_date,
DATE(MIN(s.InvoiceDate)) AS first_purchase_date,
COUNT(DISTINCT s.InvoiceNo) AS num_purchases,
SUM(b.total) AS monetary,
FROM
`customer-segmentation-373712.retail.sales` s
LEFT JOIN
`customer-segmentation-373712.retail.bill` b
ON
s.InvoiceNo=b.InvoiceNo
GROUP BY
CustomerID
)SELECT
*,
DATE_DIFF(reference_date, last_purchase_date, DAY) AS recency,
num_purchases/ (months_cust) AS frequency,
FROM
(
SELECT *,
MAX(last_purchase_date) OVER () + 1 AS reference_date,
DATE_DIFF(cte.last_purchase_date, cte.first_purchase_date, month)+1 AS months_cust
FROM cte);
```

5.
```sql
SELECT
    a.*,
```

```sql
    --All percentiles for MONETARY
    b.percentiles[offset(20)] AS m20,
    b.percentiles[offset(40)] AS m40,
    b.percentiles[offset(60)] AS m60,
    b.percentiles[offset(80)] AS m80,
    b.percentiles[offset(100)] AS m100,
    --All percentiles for FREQUENCY
    c.percentiles[offset(20)] AS f20,
    c.percentiles[offset(40)] AS f40,
    c.percentiles[offset(60)] AS f60,
    c.percentiles[offset(80)] AS f80,
    c.percentiles[offset(100)] AS f100,
    --All percentiles for RECENCY
    d.percentiles[offset(20)] AS r20,
    d.percentiles[offset(40)] AS r40,
    d.percentiles[offset(60)] AS r60,
    d.percentiles[offset(80)] AS r80,
    d.percentiles[offset(100)] AS r100
FROM
    `customer-segmentation-373712.retail.RFM` a,
    (SELECT APPROX_QUANTILES(monetary, 100) percentiles FROM
    `customer-segmentation-373712.retail.RFM`) b,
    (SELECT APPROX_QUANTILES(frequency, 100) percentiles FROM
    `customer-segmentation-373712.retail.RFM`) c,
    (SELECT APPROX_QUANTILES(recency, 100) percentiles FROM
    `customer-segmentation-373712.retail.RFM`) d
```

6.
```sql
SELECT CustomerID,
m_score,f_score,r_score,
recency, freq,monetary,
  CAST(ROUND((f_score + m_score) / 2, 0) AS INT64) AS fm_score
  FROM (
      SELECT *,
      CASE WHEN monetary <= m20 THEN 1
          WHEN monetary <= m40 AND monetary > m20 THEN 2
          WHEN monetary <= m60 AND monetary > m40 THEN 3
```

```sql
        WHEN monetary <= m80 AND monetary > m60 THEN 4
        WHEN monetary <= m100 AND monetary > m80 THEN 5
    END AS m_score,
    CASE WHEN freq <= f20 THEN 1
        WHEN freq <= f40 AND freq > f20 THEN 2
        WHEN freq <= f60 AND freq > f40 THEN 3
        WHEN freq <= f80 AND freq > f60 THEN 4
        WHEN freq <= f100 AND freq > f80 THEN 5
    END AS f_score,
    --Recency scoring is reversed
    CASE WHEN recency <= r20 THEN 5
        WHEN recency <= r40 AND recency > r20 THEN 4
        WHEN recency <= r60 AND recency > r40 THEN 3
        WHEN recency <= r80 AND recency > r60 THEN 2
        WHEN recency <= r100 AND recency > r80 THEN 1
    END AS r_score,
    FROM `customer-segmentation-373712.retail.Quintiles`
)
```

7.

```sql
SELECT
    CustomerID,
    recency,freq,monetary,
    r_score, f_score, m_score,
    fm_score,
    CASE WHEN (r_score = 5 AND fm_score = 5)
        OR (r_score = 5 AND fm_score = 4)
        OR (r_score = 4 AND fm_score = 5)
    THEN 'Champions'
    WHEN (r_score = 5 AND fm_score =3)
        OR (r_score = 4 AND fm_score = 4)
        OR (r_score = 3 AND fm_score = 5)
        OR (r_score = 3 AND fm_score = 4)
    THEN 'Loyal Customers'
    WHEN (r_score = 5 AND fm_score = 2)
```

```sql
            OR (r_score = 4 AND fm_score = 2)
            OR (r_score = 3 AND fm_score = 3)
            OR (r_score = 4 AND fm_score = 3)
        THEN 'Potential Loyalists'
        WHEN r_score = 5 AND fm_score = 1 THEN 'Recent Customers'
        WHEN (r_score = 4 AND fm_score = 1)
            OR (r_score = 3 AND fm_score = 1)
        THEN 'Promising'
        WHEN (r_score = 3 AND fm_score = 2)
            OR (r_score = 2 AND fm_score = 3)
            OR (r_score = 2 AND fm_score = 2)
        THEN 'Customers Needing Attention'
        WHEN r_score = 2 AND fm_score = 1 THEN 'About to Sleep'
        WHEN (r_score = 2 AND fm_score = 5)
            OR (r_score = 2 AND fm_score = 4)
            OR (r_score = 1 AND fm_score = 3)
        THEN 'At Risk'
        WHEN (r_score = 1 AND fm_score = 5)
            OR (r_score = 1 AND fm_score = 4)
        THEN 'Cant Lose Them'
        WHEN r_score = 1 AND fm_score = 2 THEN 'Hibernating'
        WHEN r_score = 1 AND fm_score = 1 THEN 'Lost'
        END AS rfm_segment
FROM score
```