```sql
Select *,
      Quantity * UnitPrice as amount

from `retail.sales`


with bills as (
Select *,
      Quantity * UnitPrice as amount

from `retail.sales`)



Select InvoiceNo,
     sum(bills.amount) as total
from bills
group by 1



Select s.CustomerID,
      s.InvoiceNo,
      s.InvoiceDate,
      b.total,
      ROW_NUMBER() over(partition by s.InvoiceNo order by s.InvoiceNo) as RN

from `retail.sales` s
left join `retail.bills` b
on s.InvoiceNo =b.InvoiceNo
```

```sql
with monetory as (
Select s.CustomerID,
      s.InvoiceNo,
      s.InvoiceDate,
      b.total,
      ROW_NUMBER() over(partition by s.InvoiceNo order by s.InvoiceNo) as RN

from `retail.sales` s
left join `retail.bills` b
on s.InvoiceNo =b.InvoiceNo)

Select CustomerID,
      date(max(InvoiceDate)) as last_purchase_date,
      date(min(InvoiceDate)) as first_purchase_date,
      count(distinct InvoiceNo) as num_purchases,
      Sum(total) as monetory
From monetory
where rn =1
group by 1
```

```sql
with recency as (
Select *,
      max(last_purchase_date) over()+1 as reference_date,
      date_diff(last_purchase_date, first_purchase_date, month)+1 as months_cust
from `retail.monetory`)

Select *,
      date_diff(recency.reference_date, recency.last_purchase_date, day) as recency,
      num_purchases/(months_cust) as frequency
from recency
```

```sql
SELECT
  a.*,
  --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
  `retail.rfm` a,
  (SELECT APPROX_QUANTILES(monetory, 100) percentiles
   FROM `retail.rfm`) b,
  (SELECT APPROX_QUANTILES(frequency, 100) percentiles
   FROM `retail.rfm`) c,
  (SELECT APPROX_QUANTILES(recency, 100) percentiles
   FROM `retail.rfm`) d
ORDER BY CustomerID
```

```sql
SELECT
```

```sql
    CustomerID,
    m_score,
    f_score,
    r_score,
    recency,
    frequency,
    monetory,
    CAST(ROUND((f_score + m_score) / 2, 0) AS INT64) AS fm_score
FROM (
    SELECT *,
        CASE
            WHEN monetory <= m20 THEN 1
            WHEN monetory <= m40 AND monetory > m20 THEN 2
            WHEN monetory <= m60 AND monetory > m40 THEN 3
            WHEN monetory <= m80 AND monetory > m60 THEN 4
            WHEN monetory <= m100 AND monetory > m80 THEN 5
        END AS m_score,
        CASE
            WHEN frequency <= f20 THEN 1
            WHEN frequency <= f40 AND frequency > f20 THEN 2
            WHEN frequency <= f60 AND frequency > f40 THEN 3
            WHEN frequency <= f80 AND frequency > f60 THEN 4
            WHEN frequency <= f100 AND frequency > 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 `retail.quintile`
) AS subquery
```

```sql
SELECT
   CustomerID,
   recency,
   frequency,
   monetory,
   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)
             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
   `retail.score`
ORDER BY
   CustomerID;
```