

✅ Task 4: Determine the Total Amount Spent (Monetary Value)

◆ Goal:

Calculate the **monetary value** — total amount spent by each customer in 2022.

🧠 SQL Query:

```
SQL
SELECT
    o.user_id AS customer_id,
    SUM(oi.sale_price) AS monetary
FROM `thelook_ecommerce.orders` o
INNER JOIN `thelook_ecommerce.order_items` oi
ON o.order_id = oi.order_id
WHERE o.created_at >= '2022-01-01' AND o.created_at < '2023-01-01'
GROUP BY customer_id
LIMIT 10;
```

✅ Correct Outcome:

- Returns the top 10 customers based on **total spending** in 2022.
 - Uses an **inner join** to combine orders and order_items tables.
 - Uses **SUM()** to calculate total sale_price per customer.
-

✅ Task 5: Create a Common Table Expression (CTE)

◆ Goal:

Combine **recency**, **frequency**, and **monetary** calculations into a single query using a CTE.

🧠 SQL Query:

```
SQL
WITH rfm_calc AS (
    SELECT
        o.user_id AS customer_id,
        DATE_DIFF(CURRENT_TIMESTAMP(), MAX(o.created_at), DAY) AS
recency,
        COUNT(o.order_id) AS frequency,
        ROUND(SUM(oi.sale_price)) AS monetary
    FROM `thelook_ecommerce.orders` o
    INNER JOIN `thelook_ecommerce.order_items` oi
```

```
    ON o.order_id = oi.order_id
  GROUP BY customer_id
)
```

```
SELECT *
FROM rfm_calc;
```

Afficher plus de lignes

✅ **Correct Answer: What is the highest frequency of visits for a single customer?**

- ✅ 17

Correct: Sorting the frequency column in descending order reveals the highest number of orders placed by a single customer.

✅ Task 6: Apply a Statistical Method to RFM Calculations

◆ Goal:

Use **quantiles** and the **NTILE()** function to segment customers based on their **RFM scores**.

🧠 SQL Query:

```
SQL
WITH
rfm_calc AS (
  SELECT
    o.user_id AS customer_id,
    DATE_DIFF(CURRENT_TIMESTAMP(), MAX(o.created_at), DAY) AS
recency,
    COUNT(o.order_id) AS frequency,
    ROUND(SUM(oi.sale_price)) AS monetary
  FROM `thelook_ecommerce.orders` o
  INNER JOIN `thelook_ecommerce.order_items` oi
  ON o.order_id = oi.order_id
  GROUP BY customer_id
),
```

```
rfm_quant AS (
  SELECT
    customer_id,
    NTILE(4) OVER (ORDER BY recency) AS recency_quantile,
    NTILE(4) OVER (ORDER BY frequency) AS frequency_quantile,
```

```

        NTILE(4) OVER (ORDER BY monetary) AS monetary_quantile
    FROM rfm_calc
)

SELECT
    customer_id,
    recency_quantile,
    frequency_quantile,
    monetary_quantile,
    CASE
        WHEN monetary_quantile >= 3 AND frequency_quantile >= 3 THEN
"High Value Customer"
        WHEN frequency_quantile >= 3 THEN "Loyal Customer"
        WHEN recency_quantile <= 1 THEN "At Risk Customer"
        WHEN recency_quantile >= 3 THEN "Persuadable Customer"
    END AS customer_segment
FROM rfm_quant;

```

Afficher plus de lignes

✅ **Correct Answer: What is likely true about customers with a recency_quantile of 4?**

- None of these options
- The customer has made a more recent purchase than 25% of customers
- The customer has made a purchase within 30 days
- **✅ The customer has made a purchase within the most recent 25% of all customers**

Correct: A quantile of 4 (in NTILE(4)) means the customer is in the top 25% for recency — i.e., they purchased most recently.

Conclusion

 **Great work!** You've completed a full RFM analysis using BigQuery.

✅ **What You Accomplished:**

1. **Explored the dataset** (orders, order_items).
2. Calculated **recency** (last purchase date).
3. Measured **frequency** (number of orders in 2022).
4. Computed **monetary value** (total spend in 2022).
5. Created a **CTE** to consolidate RFM metrics.
6. Applied **quantiles and CASE logic** to segment customers into:
 - High Value

- Loyal
 - At Risk
 - Persuadable
-

Key Takeaways:

- RFM segmentation helps tailor marketing strategies.
- BigQuery and SQL are powerful tools for customer analytics.
- CTEs and NTILE functions simplify complex transformations