Sprint 3 - Queries

UPDATE segmentation:

BY COUNT:

```
-- Recency calculation
WITH recency_data AS (
   SELECT
       user_crm_id,
       latest_purchase_date,
       DATE_DIFF((SELECT MAX(latest_purchase_date) FROM
`prism-insights.warehouse_PT.users`), latest_purchase_date, MONTH) AS Recency
   FROM `prism-insights.warehouse_PT.users`
),
recency_score_data AS (
   SELECT
       user_crm_id,
       latest_purchase_date,
       CASE
           WHEN Recency BETWEEN 0 AND 2 THEN 1
           WHEN Recency BETWEEN 3 AND 6 THEN 2
           ELSE 3
       END AS recency_score
   FROM recency_data
),

-- Frequency calculation
frequency_data AS (
   SELECT
       user_crm_id,
       COUNT(DISTINCT transaction_id) AS purchase_count
   FROM `prism-insights.warehouse_PT.transactions`
   WHERE user_crm_id IS NOT NULL
   GROUP BY user_crm_id
),
frequency_score_data AS (
   SELECT
       user_crm_id,
       CASE
           WHEN purchase_count > PERCENTILE_CONT(purchase_count, 0.90) OVER() THEN
1
```

```
            WHEN purchase_count > PERCENTILE_CONT(purchase_count, 0.7) OVER() THEN 2
            ELSE 3
        END AS frequency_score
    FROM frequency_data
),

-- Monetary value calculation
monetary_data AS (
    SELECT
        user_crm_id,
        ROUND(SUM(transaction_total), 0) AS total_spent
    FROM `prism-insights.warehouse_PT.transactions`
    WHERE user_crm_id IS NOT NULL
    GROUP BY user_crm_id
),
monetary_score_data AS (
    SELECT
        user_crm_id,
        total_spent,
        CASE
            WHEN total_spent > PERCENTILE_CONT(total_spent, 0.80) OVER() THEN 1
            WHEN total_spent > PERCENTILE_CONT(total_spent, 0.50) OVER() THEN 2
            ELSE 3
        END AS monetary_score
    FROM monetary_data
),

-- Combined RFM segments
combined_rfm AS (
    SELECT
        r.user_crm_id,
        r.recency_score,
        f.frequency_score,
        m.monetary_score,
        (r.recency_score + f.frequency_score + m.monetary_score) AS total_score
    FROM recency_score_data r
    LEFT JOIN frequency_score_data f ON r.user_crm_id = f.user_crm_id
    LEFT JOIN monetary_score_data m ON r.user_crm_id = m.user_crm_id
),

-- Segment classification
segment_data AS (
    SELECT
        user_crm_id,
```

```
        CASE
            WHEN total_score BETWEEN 3 AND 4 THEN '1 - Trendy'
            WHEN total_score BETWEEN 5 AND 6 THEN '2 - Engaged Shoppers'
            WHEN total_score = 7 THEN '3 - Casual Buyers'
            WHEN total_score = 8 THEN '4 - At-Risk Customers'
            ELSE '5 - Lost Causes'
        END AS segment
    FROM combined_rfm
),

-- Final output
main AS (
    SELECT
        u.user_crm_id,
        s.total_spent,
        u.city,
        u.user_gender,
        freq.purchase_count,
        u.latest_purchase_date,
        seg.segment
    FROM `prism-insights.warehouse_PT.users` u
    LEFT JOIN monetary_data s ON u.user_crm_id = s.user_crm_id
    LEFT JOIN segment_data seg ON u.user_crm_id = seg.user_crm_id
    LEFT JOIN frequency_data freq ON u.user_crm_id = freq.user_crm_id
    ORDER BY user_crm_id
)

SELECT segment, COUNT(*)
FROM main
GROUP BY segment
ORDER BY segment;
```

---

```
PER USER:

-- Recency calculation
WITH recency_data AS (
    SELECT
        user_crm_id,
        latest_purchase_date,
        DATE_DIFF((SELECT MAX(latest_purchase_date) FROM
`prism-insights.warehouse_PT.users`), latest_purchase_date, MONTH) AS Recency
```

```
    FROM `prism-insights.warehouse_PT.users`
),
recency_score_data AS (
    SELECT
        user_crm_id,
        latest_purchase_date,
        CASE
            WHEN Recency BETWEEN 0 AND 2 THEN 1
            WHEN Recency BETWEEN 3 AND 6 THEN 2
            ELSE 3
        END AS recency_score
    FROM recency_data
),

-- Frequency calculation
frequency_data AS (
    SELECT
        user_crm_id,
        COUNT(DISTINCT transaction_id) AS purchase_count
    FROM `prism-insights.warehouse_PT.transactions`
    WHERE user_crm_id IS NOT NULL
    GROUP BY user_crm_id
),
frequency_score_data AS (
    SELECT
        user_crm_id,
        CASE
            WHEN purchase_count > PERCENTILE_CONT(purchase_count, 0.90) OVER() THEN
1
            WHEN purchase_count > PERCENTILE_CONT(purchase_count, 0.7) OVER() THEN 2
            ELSE 3
        END AS frequency_score
    FROM frequency_data
),

-- Monetary value calculation
monetary_data AS (
    SELECT
        user_crm_id,
        ROUND(SUM(transaction_total), 0) AS total_spent
    FROM `prism-insights.warehouse_PT.transactions`
    WHERE user_crm_id IS NOT NULL
    GROUP BY user_crm_id
),
```

```sql
monetary_score_data AS (
    SELECT
        user_crm_id,
        total_spent,
        CASE
            WHEN total_spent > PERCENTILE_CONT(total_spent, 0.80) OVER() THEN 1
            WHEN total_spent > PERCENTILE_CONT(total_spent, 0.50) OVER() THEN 2
            ELSE 3
        END AS monetary_score
    FROM monetary_data
),

-- Combined RFM segments
combined_rfm AS (
    SELECT
        r.user_crm_id,
        r.recency_score,
        f.frequency_score,
        m.monetary_score,
        (r.recency_score + f.frequency_score + m.monetary_score) AS total_score
    FROM recency_score_data r
    LEFT JOIN frequency_score_data f ON r.user_crm_id = f.user_crm_id
    LEFT JOIN monetary_score_data m ON r.user_crm_id = m.user_crm_id
),

-- Segment classification
segment_data AS (
    SELECT
        user_crm_id,
        CASE
            WHEN total_score BETWEEN 3 AND 4 THEN '1 - Trendy'
            WHEN total_score BETWEEN 5 AND 6 THEN '2 - Engaged Shoppers'
            WHEN total_score = 7 THEN '3 - Casual Buyers'
            WHEN total_score = 8 THEN '4 - At-Risk Customers'
            ELSE '5 - Lost Causes'
        END AS segment
    FROM combined_rfm
),

-- Final output
main AS (
    SELECT
        u.user_crm_id,
        s.total_spent,
```

```
        u.city,
        u.user_gender,
        freq.purchase_count,
        u.latest_purchase_date,
        seg.segment
    FROM `prism-insights.warehouse_PT.users` u
    LEFT JOIN monetary_data s ON u.user_crm_id = s.user_crm_id
    LEFT JOIN segment_data seg ON u.user_crm_id = seg.user_crm_id
    LEFT JOIN frequency_data freq ON u.user_crm_id = freq.user_crm_id
    ORDER BY user_crm_id
)

SELECT
    user_crm_id,
    total_spent,
    city,
    user_gender,
    purchase_count,
    latest_purchase_date,
    segment
FROM main
ORDER BY user_crm_id;
```