Cash Flow vs. Loyalty Assessment (Finance)

Language: PostgreSQL

```postgresql
/*
   BUSINESS QUESTION:
   "If we restrict high-installment credit plans, will it hurt our
   Champions or mainly impact low-value segments?"

   WHY THIS MATTERS:
   High-installment plans (8-12 months) are expensive for the business
   (interchange fees, credit risk, cash flow delay). But if Champions
   depend on them, restricting installments could backfire. This query
   breaks down credit dependency by segment so Finance can model the
   impact before changing payment policy.
*/

WITH delivered_base AS (
    SELECT
        c.customer_unique_id,
        o.order_id,
        o.order_purchase_timestamp
    FROM olist_customers_dataset        AS c
    INNER JOIN olist_orders_dataset      AS o
        ON o.customer_id = c.customer_id
    WHERE o.order_status = 'delivered'
),

rfm_metrics AS (
    SELECT
        db.customer_unique_id,
        (SELECT MAX(order_purchase_timestamp)::DATE FROM delivered_base)
            - MAX(db.order_purchase_timestamp)::DATE        AS recency_days,
        COUNT(DISTINCT db.order_id)                         AS frequency,
        ROUND(SUM(p.payment_value)::NUMERIC, 2)             AS monetary
    FROM delivered_base                     AS db
    INNER JOIN olist_order_payments_dataset  AS p
        ON p.order_id = db.order_id
    GROUP BY db.customer_unique_id
),

rfm_scores AS (
    SELECT *,
        NTILE(5) OVER (ORDER BY recency_days DESC)     AS r_score,
        NTILE(5) OVER (ORDER BY frequency     ASC)     AS f_score,
```

```sql
        NTILE(5) OVER (ORDER BY monetary      ASC)     AS m_score
    FROM rfm_metrics
),

rfm_segments AS (
    SELECT *,
        CASE
            WHEN r_score >= 4 AND f_score >= 4 AND m_score >= 4 THEN 'Champions'
            WHEN r_score <= 2 AND f_score >= 4 AND m_score >= 4 THEN 'At Risk'
            WHEN r_score >= 4 AND f_score <= 2 AND m_score <= 2 THEN 'New Customers'
            WHEN r_score >= 4 AND (f_score >= 3 OR m_score >= 3) THEN 'Potential Loyalists'
            WHEN r_score >= 3 AND f_score >= 3 AND m_score >= 3 THEN 'Loyal Customers'
            WHEN r_score <= 2 AND f_score <= 2 AND m_score <= 2 THEN 'Hibernating'
            ELSE 'Need Attention'
        END AS customer_segment
    FROM rfm_scores
),

/* ■■ Financial dimension: dominant payment type + avg installments ■■ */
dim_financial AS (
    SELECT
        f_type.customer_unique_id,
        f_type.dominant_payment_type,
        COALESCE(f_inst.avg_installments, 0)            AS avg_installments,
        COALESCE(f_inst.max_installments, 0)            AS max_installments
    FROM (
        SELECT customer_unique_id, payment_type AS dominant_payment_type
        FROM (
            SELECT db.customer_unique_id, p.payment_type,
                   ROW_NUMBER() OVER (PARTITION BY db.customer_unique_id
                                      ORDER BY COUNT(*) DESC, SUM(p.payment_value) DESC) AS rn
            FROM delivered_base                         AS db
            INNER JOIN olist_order_payments_dataset      AS p ON p.order_id = db.order_id
            GROUP BY db.customer_unique_id, p.payment_type
        ) ranked WHERE rn = 1
    ) f_type
    LEFT JOIN (
        SELECT db.customer_unique_id,
               ROUND(AVG(p.payment_installments)::NUMERIC, 1) AS avg_installments,
               MAX(p.payment_installments)                    AS max_installments
        FROM delivered_base                             AS db
        INNER JOIN olist_order_payments_dataset          AS p ON p.order_id = db.order_id
        WHERE p.payment_type = 'credit_card' AND p.payment_installments > 0
        GROUP BY db.customer_unique_id
    ) f_inst ON f_inst.customer_unique_id = f_type.customer_unique_id
),

```

```sql
/* ■■ Revenue by payment type per customer (for boleto % calc) ■■ */
payment_revenue AS (
    SELECT
        db.customer_unique_id,
        SUM(p.payment_value)                            AS total_paid,
        SUM(CASE WHEN p.payment_type = 'boleto'
                 THEN p.payment_value ELSE 0 END)       AS boleto_paid,
        SUM(CASE WHEN p.payment_type = 'credit_card'
                 THEN p.payment_value ELSE 0 END)       AS credit_card_paid
    FROM delivered_base                                 AS db
    INNER JOIN olist_order_payments_dataset              AS p ON p.order_id = db.order_id
    GROUP BY db.customer_unique_id
)

/* ■■ FINAL: Credit dependency profile by segment ■■ */
SELECT
    rfm.customer_segment,
    COUNT(*)                                            AS total_customers,
    ROUND(SUM(rfm.monetary)::NUMERIC, 2)                AS total_segment_revenue,

    /* Average installments used (credit card only)                          */
    ROUND(AVG(fin.avg_installments)::NUMERIC, 1)        AS avg_cc_installments,

    /* % of customers in this segment who max out at 6+ installments —
        these are the "installment-dependent" buyers                          */
    ROUND(100.0 *
        SUM(CASE WHEN fin.max_installments >= 6 THEN 1 ELSE 0 END)::NUMERIC
        / COUNT(*)
    , 1)                                                AS pct_using_6plus_installments,

    /* Revenue concentration by payment method                               */
    ROUND(100.0 * SUM(pr.boleto_paid)::NUMERIC
          / NULLIF(SUM(pr.total_paid), 0), 1)           AS pct_revenue_via_boleto,
    ROUND(100.0 * SUM(pr.credit_card_paid)::NUMERIC
          / NULLIF(SUM(pr.total_paid), 0), 1)           AS pct_revenue_via_credit_card,

    /* Dominant payment type distribution                                    */
    SUM(CASE WHEN fin.dominant_payment_type = 'credit_card' THEN 1 ELSE 0 END)
                                                        AS count_dominant_cc,
    SUM(CASE WHEN fin.dominant_payment_type = 'boleto'      THEN 1 ELSE 0 END)
                                                        AS count_dominant_boleto

FROM rfm_segments       AS rfm
LEFT JOIN dim_financial AS fin ON fin.customer_unique_id = rfm.customer_unique_id
LEFT JOIN payment_revenue AS pr ON pr.customer_unique_id = rfm.customer_unique_id

GROUP BY rfm.customer_segment
```

```sql
ORDER BY total_segment_revenue DESC;
```