The Silent Defector Phenomenon (Strategy)

Language: PostgreSQL

```sql
/*
   BUSINESS QUESTION:
   "How much revenue are we losing from customers who left despite having
   zero late deliveries and positive review sentiment — proving the churn
   is caused by external factors we cannot see in our operational data?"

   WHY THIS MATTERS:
   This is the most important strategic query. When a customer churns
   after a bad experience (late delivery, damaged product), the fix is
   operational. But when a customer churns despite PERFECT execution —
   no late deliveries, 4-5 star reviews — the cause is external:
   competitor pricing, market shift, or unmet needs we are not measuring.

   This revenue figure is the board-level answer to: "How much money
   are we losing to forces outside our control, and how big is the
   strategic gap we need to close with product innovation or pricing?"
*/

WITH delivered_base AS (
    SELECT
        c.customer_unique_id,
        o.order_id,
        o.order_purchase_timestamp,
        o.order_delivered_customer_date,
        o.order_estimated_delivery_date
    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,
        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
),

/* ■■ Sentiment: last review score per customer ■■ */
dim_sentiment AS (
    SELECT customer_unique_id, last_review_score
    FROM (
        SELECT
            db.customer_unique_id,
            r.review_score                                  AS last_review_score,
            ROW_NUMBER() OVER (PARTITION BY db.customer_unique_id
                               ORDER BY r.review_creation_date DESC,
                                        r.review_answer_timestamp DESC) AS rn
        FROM delivered_base                                AS db
        INNER JOIN olist_order_reviews_dataset              AS r ON r.order_id = db.order_id
    ) ranked WHERE rn = 1
),

/* ■■ Operational Friction: late delivery count per customer ■■ */
dim_friction AS (
    SELECT
        customer_unique_id,
        SUM(CASE WHEN order_delivered_customer_date > order_estimated_delivery_date
                 THEN 1 ELSE 0 END)                        AS late_delivery_count
    FROM delivered_base
```

```sql
    WHERE order_delivered_customer_date IS NOT NULL
    GROUP BY customer_unique_id
),

/* ■■ Isolate the "Silent Defectors": churned + perfect operations ■■ */
silent_defectors AS (
    SELECT
        rfm.customer_unique_id,
        rfm.customer_segment,
        rfm.monetary,
        rfm.recency_days,
        rfm.frequency,
        snt.last_review_score,
        COALESCE(f.late_delivery_count, 0)             AS late_delivery_count
    FROM rfm_segments       AS rfm
    LEFT JOIN dim_sentiment AS snt ON snt.customer_unique_id = rfm.customer_unique_id
    LEFT JOIN dim_friction  AS f   ON f.customer_unique_id   = rfm.customer_unique_id
    WHERE rfm.customer_segment IN ('At Risk', 'Hibernating')
      AND COALESCE(f.late_delivery_count, 0) = 0       /* Zero late deliveries   */
      AND snt.last_review_score >= 4                    /* Positive last review   */
)

/* ■■ FINAL: The board-level answer ■■ */
SELECT
    '■■ OVERALL SILENT DEFECTOR IMPACT ■■'            AS analysis_level,
    customer_segment                                  AS segment,
    COUNT(*)                                          AS silent_defector_count,

    /* Total revenue from customers who left despite perfect execution      */
    ROUND(SUM(monetary)::NUMERIC, 2)                  AS total_revenue_lost,

    /* Average LTV of a silent defector — shows they are not low-value      */
    ROUND(AVG(monetary)::NUMERIC, 2)                  AS avg_ltv_per_defector,

    /* Average recency — how long ago they went silent                      */
    ROUND(AVG(recency_days)::NUMERIC, 0)              AS avg_days_since_last_order,

    /* Review score distribution — proves the satisfaction was genuine        */
    SUM(CASE WHEN last_review_score = 5 THEN 1 ELSE 0 END) AS five_star_count,
    SUM(CASE WHEN last_review_score = 4 THEN 1 ELSE 0 END) AS four_star_count,
    ROUND(AVG(last_review_score)::NUMERIC, 2)         AS avg_review_score,

    /* Context: what % of all At Risk / Hibernating are silent defectors?   */
    ROUND(100.0 * COUNT(*)::NUMERIC / (
        SELECT COUNT(*) FROM rfm_segments
        WHERE customer_segment IN ('At Risk', 'Hibernating')
    ), 1)                                             AS pct_of_all_churned
```

```sql

FROM silent_defectors
GROUP BY customer_segment

UNION ALL

/* Grand total row for the board presentation                              */
SELECT
    '■■ GRAND TOTAL ■■'                          AS analysis_level,
    'All Silent Defectors'                       AS segment,
    COUNT(*),
    ROUND(SUM(monetary)::NUMERIC, 2),
    ROUND(AVG(monetary)::NUMERIC, 2),
    ROUND(AVG(recency_days)::NUMERIC, 0),
    SUM(CASE WHEN last_review_score = 5 THEN 1 ELSE 0 END),
    SUM(CASE WHEN last_review_score = 4 THEN 1 ELSE 0 END),
    ROUND(AVG(last_review_score)::NUMERIC, 2),
    ROUND(100.0 * COUNT(*)::NUMERIC / (
        SELECT COUNT(*) FROM rfm_segments
        WHERE customer_segment IN ('At Risk', 'Hibernating')
    ), 1)
FROM silent_defectors

ORDER BY analysis_level, total_revenue_lost DESC;
```