Carrier Performance & Geo-Churn Risk (Operations)

Language: PostgreSQL

```sql
/*
   BUSINESS QUESTION:
   "In which states are logistics failures costing us the most lifetime
   value — and how much churned revenue is directly tied to late deliveries?"

   WHY THIS MATTERS:
   A late delivery does not just lose one order — it poisons the entire
   customer relationship. This query isolates the At Risk + Hibernating
   cohorts (our churned/churning customers), calculates what % of their
   orders arrived late, and sums the total revenue we are losing per state.
   The output is a heat-map-ready table for supply chain prioritization.
*/

WITH delivered_base AS (
    SELECT
        c.customer_unique_id,
        c.customer_state,
        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 (
```

```sql
    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
),

/* ■■ Geographic: most recent state per customer ■■ */
dim_geo AS (
    SELECT customer_unique_id, customer_state
    FROM (
        SELECT customer_unique_id, customer_state,
               ROW_NUMBER() OVER (PARTITION BY customer_unique_id
                                  ORDER BY order_purchase_timestamp DESC) AS rn
        FROM delivered_base
    ) ranked WHERE rn = 1
),

/* ■■ Operational Friction: late delivery metrics 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,
        COUNT(*)                                        AS total_delivered_orders
    FROM delivered_base
    WHERE order_delivered_customer_date IS NOT NULL
    GROUP BY customer_unique_id
)

/* ■■ FINAL AGGREGATION: Churned revenue by state, split by late-delivery exposure ■■ */
SELECT
    geo.customer_state                                  AS state,
```

```sql
    COUNT(*)                                            AS churned_customers,
    ROUND(SUM(rfm.monetary)::NUMERIC, 2)                AS total_churned_revenue,

    /* Customers in this state who experienced at least one late delivery   */
    SUM(CASE WHEN f.late_delivery_count >= 1 THEN 1 ELSE 0 END)
                                                        AS customers_with_late_deliveries,
    /* % of churned customers who had a late delivery                        */
    ROUND(100.0 * SUM(CASE WHEN f.late_delivery_count >= 1 THEN 1 ELSE 0 END)
          / COUNT(*)::NUMERIC, 1)                       AS pct_churned_with_late_delivery,

    /* Revenue from churned customers who experienced late delivery —
        this is the direct "revenue cost of logistics failure"               */
    ROUND(SUM(CASE WHEN f.late_delivery_count >= 1
                   THEN rfm.monetary ELSE 0 END)::NUMERIC, 2)
                                                        AS revenue_tied_to_late_delivery,
    /* Average late delivery rate among churned customers in this state     */
    ROUND(AVG(
        CASE WHEN f.total_delivered_orders > 0
             THEN 100.0 * f.late_delivery_count / f.total_delivered_orders
             ELSE 0 END
    )::NUMERIC, 1)                                      AS avg_late_delivery_pct

FROM rfm_segments       AS rfm
INNER JOIN dim_geo      AS geo  ON geo.customer_unique_id = rfm.customer_unique_id
LEFT JOIN dim_friction  AS f    ON f.customer_unique_id   = rfm.customer_unique_id

/* Filter to only churned / churning segments                               */
WHERE rfm.customer_segment IN ('At Risk', 'Hibernating')

GROUP BY geo.customer_state
ORDER BY total_churned_revenue DESC;
```