Gateway Category LTV Analysis (Marketing)

Language: PostgreSQL

```sql
/*
   BUSINESS QUESTION:
   "Which product categories produce Champions versus one-hit-wonder
   Hibernators — and what is the lifetime value gap between them?"

   WHY THIS MATTERS:
   Marketing budgets are often allocated by category GMV, but GMV
   ignores customer quality. A category that generates R$1M from
   Champions is worth far more than one generating R$1M from
   Hibernating buyers who never return. This query reveals which
   categories are "gateway drugs" to loyalty and which are dead ends.
*/

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,
```

```
        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
),

/* ■■ Product Affinity: top category per customer (anti-fan-out) ■■ */
dim_product_affinity AS (
    SELECT customer_unique_id, top_category_english
    FROM (
        SELECT
            db.customer_unique_id,
            COALESCE(t.product_category_name_english, 'unknown')
                                                    AS top_category_english,
            ROW_NUMBER() OVER (
                PARTITION BY db.customer_unique_id
                ORDER BY COUNT(*) DESC, SUM(oi.price) DESC
            ) AS rn
        FROM delivered_base                         AS db
        INNER JOIN olist_order_items_dataset         AS oi ON oi.order_id = db.order_id
        INNER JOIN olist_products_dataset            AS pr ON pr.product_id = oi.product_id
        LEFT JOIN product_category_name_translation  AS t  ON t.product_category_name = pr.product_category_name
        GROUP BY db.customer_unique_id,
                 COALESCE(t.product_category_name_english, 'unknown')
    ) ranked WHERE rn = 1
)

/* ■■ FINAL: Category × Segment distribution with LTV metrics ■■ */
SELECT
    pa.top_category_english                         AS gateway_category,
    COUNT(*)                                        AS total_customers,
    ROUND(SUM(rfm.monetary)::NUMERIC, 2)            AS total_category_revenue,
    ROUND(AVG(rfm.monetary)::NUMERIC, 2)            AS avg_ltv,

    /* Segment distribution within each category                    */
```

```sql
    SUM(CASE WHEN rfm.customer_segment = 'Champions'   THEN 1 ELSE 0 END) AS champions_count,
    SUM(CASE WHEN rfm.customer_segment = 'Hibernating' THEN 1 ELSE 0 END) AS hibernating_count,

    /* "Loyalty Ratio": Champions / (Champions + Hibernating).
        Higher = category produces loyal buyers.
        Lower  = category produces one-hit-wonders.                        */
    ROUND(100.0 *
        SUM(CASE WHEN rfm.customer_segment = 'Champions' THEN 1 ELSE 0 END)::NUMERIC /
        NULLIF(
            SUM(CASE WHEN rfm.customer_segment IN ('Champions','Hibernating') THEN 1 ELSE 0 END)
        , 0)
    , 1)                                                AS loyalty_ratio_pct,

    /* Average LTV of Champions vs Hibernating in this category            */
    ROUND(AVG(CASE WHEN rfm.customer_segment = 'Champions'
                   THEN rfm.monetary END)::NUMERIC, 2)  AS avg_ltv_champions,
    ROUND(AVG(CASE WHEN rfm.customer_segment = 'Hibernating'
                   THEN rfm.monetary END)::NUMERIC, 2)  AS avg_ltv_hibernating

FROM rfm_segments           AS rfm
INNER JOIN dim_product_affinity AS pa
    ON pa.customer_unique_id = rfm.customer_unique_id

GROUP BY pa.top_category_english
HAVING COUNT(*) >= 100          /* ← Filter out tiny categories with no statistical weight */
ORDER BY total_category_revenue DESC;
```