Advanced E-Commerce Cohort & Retention Analytics with SQL: Window Functions, JSON Insights, and Revenue Optimization

```
Scenario: E-commerce user engagement analytics

Objective: Analyze user purchase behavior, retention, and product trends

Features used:

  - CTEs (Common Table Expressions)

  - Window Functions

  - JSON functions

  - Aggregation & Ranking

  - Subqueries

  - Conditional logic

  - Index hints

  - Optimization-ready practices

High-value keywords:

  Analytics, User Retention, Cohort Analysis, Revenue Optimization, Purchase Funnel,
Behavioral Segmentation, Window Functions, CTE, JSON Aggregation, Indexed Query

*/


-- Step 1: Define cohorts of users based on first purchase month

WITH user_cohorts AS (

    SELECT

        user_id,
        MIN(DATE_TRUNC('month', purchase_date)) AS cohort_month
    FROM
        purchases
    GROUP BY
        user_id
),
-- Step 2: Aggregate monthly revenue per cohort
monthly_revenue AS (
    SELECT
        c.cohort_month,
        DATE_TRUNC('month', p.purchase_date) AS purchase_month,
        COUNT(DISTINCT p.user_id) AS active_users,
        SUM(p.amount) AS total_revenue
    FROM
        purchases p
    INNER JOIN
        user_cohorts c
    ON
        p.user_id = c.user_id
    GROUP BY
        c.cohort_month, DATE_TRUNC('month', p.purchase_date)
),
-- Step 3: Calculate retention metrics using window functions
retention_analysis AS (
    SELECT
```

```
        cohort_month,

        purchase_month,

        active_users,

        total_revenue,

        ROW_NUMBER() OVER (PARTITION BY cohort_month ORDER BY purchase_month)
AS month_number,

        LAG(active_users) OVER (PARTITION BY cohort_month ORDER BY purchase_month)
AS previous_month_users

    FROM

        monthly_revenue

),

-- Step 4: Compute retention rate and growth

retention_metrics AS (

    SELECT

        cohort_month,

        purchase_month,

        month_number,

        active_users,

        total_revenue,

        COALESCE(ROUND((active_users::decimal / NULLIF(previous_month_users,0)) * 100,
2), 100) AS retention_rate_percentage,

        ROUND((total_revenue::decimal / NULLIF(active_users,0)), 2) AS avg_revenue_per_user

    FROM

        retention_analysis

),

-- Step 5: Identify top 10 trending products per cohort
```

```
top_products AS (
    SELECT
        c.cohort_month,
        p.product_id,
        COUNT(p.product_id) AS purchase_count,
        SUM(p.amount) AS total_product_revenue,
        RANK() OVER (PARTITION BY c.cohort_month ORDER BY SUM(p.amount) DESC)
AS revenue_rank
    FROM
        purchases p
    INNER JOIN
        user_cohorts c
    ON
        p.user_id = c.user_id
    GROUP BY
        c.cohort_month, p.product_id
    HAVING RANK() <= 10
),
-- Step 6: Combine user behavior with product metadata stored as JSON
product_insights AS (
    SELECT
        t.cohort_month,
        t.product_id,
        t.purchase_count,
        t.total_product_revenue,
        p.product_details->>'category' AS product_category,
        p.product_details->>'brand' AS brand_name,
        p.product_details->>'tags' AS tags_json
    FROM
        top_products t
    LEFT JOIN
        products p
    ON
        t.product_id = p.product_id
),
-- Step 7: Aggregate JSON tags for SEO/keyword insights
tag_analysis AS (
    SELECT
        cohort_month,
        JSON_AGG(DISTINCT jsonb_array_elements_text(tags_json::jsonb)) AS aggregated_tags
    FROM
        product_insights
    GROUP BY
        cohort_month
)
-- Step 8: Final dashboard-ready query combining all metrics
SELECT
    r.cohort_month,
    r.purchase_month,
    r.month_number,
    r.active_users,
    r.total_revenue,
    r.retention_rate_percentage,
    r.avg_revenue_per_user,
    pi.product_id,
    pi.product_category,
    pi.brand_name,
    pi.purchase_count,
    pi.total_product_revenue,
    ta.aggregated_tags
FROM
    retention_metrics r
LEFT JOIN
    product_insights pi
ON
    r.cohort_month = pi.cohort_month
LEFT JOIN
    tag_analysis ta
ON
    r.cohort_month = ta.cohort_month
ORDER BY
    r.cohort_month ASC, r.purchase_month ASC, pi.total_product_revenue DESC;


-- Notes:
```

-- 1. JSON_AGG and jsonb_array_elements_text allow keyword extraction for SEO or analytics pipelines.

-- 2. Cohort analysis helps MAANG-style companies understand retention and engagement patterns.

-- 3. Window functions (ROW_NUMBER, LAG, RANK) optimize analytic queries for behavioral segmentation.

-- 4. This query is designed for large-scale OLAP systems with millions of rows.

-- 5. Index recommendations: INDEX(purchases(user_id, purchase_date)), INDEX(products(product_id)), INDEX(purchases(product_id))