



Ahmed Abdulwahid

⌚ Cohort Analysis
with SQL: Unlocking
Hidden Business
Insights 📊



#DataGenius



Cohort analysis is like a time machine 🕒 for businesses – it helps uncover patterns in customer behavior over time. If you're serious about customer retention, revenue growth, and strategic decision-making, mastering cohort analysis with SQL is non-negotiable!💡

🔍 What is Cohort Analysis?

Cohort analysis groups users based on a shared characteristic (e.g., first purchase date) and tracks their behavior over time. This helps businesses answer crucial questions like:

- ◆ Are customers coming back? 
- ◆ How does revenue change per cohort? 
- ◆ What's the Lifetime Value (LTV) of a user? 
- ◆ How do different marketing campaigns impact user retention? 

🛠️ Setting Up Cohort Analysis in SQL

1. Define the Cohort

A cohort is typically defined by the first purchase date of a customer. Let's start by creating a cohort table:

```
SELECT  
    customer_id,  
    MIN(DATE_TRUNC('month', purchase_date)) AS cohort_month  
FROM orders  
GROUP BY customer_id;
```

🎯 This assigns each user to a cohort month based on their first purchase.

2. Tracking Retention Over Time

JUL
17

Once we have cohorts, let's track how they behave month by month:

```
WITH cohort_data AS (
    SELECT
        o.customer_id,
        DATE_TRUNC('month', MIN(o.purchase_date)) AS cohort_month,
        DATE_TRUNC('month', o.purchase_date) AS purchase_month
    FROM orders o
    GROUP BY o.customer_id, o.purchase_date
)
SELECT
    cohort_month,
    purchase_month,
    COUNT(DISTINCT customer_id) AS active_users,
    COUNT(DISTINCT customer_id) * 100.0 /
    FIRST_VALUE(COUNT(DISTINCT customer_id)) OVER
        (PARTITION BY cohort_month ORDER BY purchase_month
         ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)
        AS retention_rate
FROM cohort_data
GROUP BY cohort_month, purchase_month;
```

🔊 Retention Rate Formula:

$$\text{Retention Rate} = \left(\frac{\text{Unique Customers in a Given Month}}{\text{Unique Customers in Cohort's First Month}} \right) \times 100$$

Where:

- *Unique Customers in a Given Month* → *The number of distinct customers who made a purchase in a specific month.*
- *Unique Customers in Cohort's First Month* → *The total number of distinct customers in the cohort's initial month.*

🔥 This allows us to track retention — how many customers from each cohort are still making purchases over time.



Key Cohort Metrics for Business Growth

1. Customer Lifetime Value (LTV)

LTV estimates how much revenue a customer generates throughout their relationship with the business. We calculate LTV using Average Revenue Per User (ARPU) and Churn Rate:

```
-- Step 1: Calculate ARPU (Average Revenue Per User) for each cohort month
WITH arpu AS (
    SELECT
        cohort_month, -- Cohort month
        SUM(total_revenue) / COUNT(DISTINCT customer_id) AS ARPU
    GROUP BY cohort_month
),

-- Step 2: Calculate churn rate for each cohort month
churn AS (
    -- Cohort data: number of active users each month
    WITH cohort AS (
        SELECT
            cohort_month,
            purchase_month,
            COUNT(DISTINCT customer_id) AS active_users
        FROM orders
        GROUP BY cohort_month, purchase_month
    )
    -- Calculate churn rate
    SELECT
        cohort_month,
        1 - (MAX(active_users) / MIN(active_users)) AS churn_rate
    FROM cohort
    GROUP BY cohort_month
)

-- Step 3: Calculate LTV (Lifetime Value) by dividing ARPU by churn rate
SELECT
    a.cohort_month,
    a.ARPU / c.churn_rate AS LTV -- LTV: ARPU divided by churn rate
FROM arpu a
JOIN churn c ON a.cohort_month = c.cohort_month;
```

- ❖ Why it matters? Higher LTV means more profitable and engaged customers!

2. Average Order Value (AOV)

AOV tells us how much customers spend per order, a crucial KPI for e-commerce:

```
SELECT
    cohort_month,
    SUM(total_revenue) / COUNT(order_id) AS AOV
FROM orders
GROUP BY cohort_month;
```

- 💡 Increasing AOV through upselling or bundling can boost overall revenue.

3. Revenue Per User (RPU)



Revenue Per User (RPU) measures how much each customer contributes in revenue:

```
SELECT
    cohort_month,
    SUM(total_revenue) / COUNT(DISTINCT customer_id) AS revenue_per_user
FROM orders
GROUP BY cohort_month;
```

🎯 **This helps identify high-value customers and optimize pricing strategies!**

4. Churn Rate



Churn rate tells us how many customers stop making purchases over time:

```
WITH first_last_purchase AS (
    SELECT
        customer_id,
        MIN(purchase_date) AS first_purchase,
        MAX(purchase_date) AS last_purchase
    FROM orders
    GROUP BY customer_id
)
SELECT
    COUNT(DISTINCT customer_id) AS churned_customers,
    COUNT(DISTINCT customer_id) * 100.0 /
    (SELECT COUNT(DISTINCT customer_id) FROM orders WHERE
        purchase_date <= NOW() - INTERVAL '3 months') AS churn_rate
FROM first_last_purchase
WHERE last_purchase < NOW() - INTERVAL '3 months';
```

⚠️ High churn? Time to rethink retention strategies!

🎯 Visualizing Cohort Analysis: Heatmaps 🔥

To make insights actionable, we need visualization! Here's how to create a retention heatmap in SQL:

```
SELECT
    cohort_month,
    purchase_month,
    COUNT(DISTINCT customer_id) AS active_users
FROM orders
GROUP BY cohort_month, purchase_month;
```

📊 Export this to Excel or Tableau for a stunning heatmap!

🏆 Why Cohort Analysis is a Game-Changer

- ◆ Helps identify retention trends 
- ◆ Optimizes marketing spend 
- ◆ Unveils high-value customers 
- ◆ Informs pricing & discount strategies 

Cohort analysis in SQL is a must-have tool in every data scientist's arsenal. Start analyzing, optimizing, and growing your business like a pro! 

R^epost it



Thank you