**MAZSQL – HOMEWORK**

**SESSION 7 – Cohort Analysis & User Segmentation**

|  |
| --- |
| **Description:**  *Paytm is an Indian multinational financial technology company. It specializes in digital payment systems, e-commerce and financial services. Paytm wallet is a secure and RBI (Reserve Bank of India)-approved digital/mobile wallet that provides a myriad of financial features to fulfill every consumer’s payment needs. Paytm wallet can be topped up through UPI (Unified Payments Interface), internet banking, or credit/debit cards. Users can also transfer money from a Paytm wallet to the recipient's bank account or their own Paytm wallet.*  Below is a small database of payment transactions from 2019 to 2020 of Paytm Wallet. The database includes 6 tables:   * fact\_transaction: Store information of all types of transactions: Payments, Top-up, Transfers, Withdrawals * dim\_scenario: Detailed description of transaction types * dim\_payment\_channel: Detailed description of payment methods * dim\_platform: Detailed description of payment devices * dim\_status: Detailed description of the results of the transaction |

## **Basic Retention Curve:**

## **Task A:** As you know that 'Telco Card' is the most popular product in the Telco group (accounting for more than 99% of the total). You want to evaluate the quality of user acquisition in Jan 2019 by the retention metric. First, you need to know how many users are retained in each subsequent month from the first month (Jan 2019) they pay the successful transaction (only get data of 2019).

***The purpose of the case study*** *below: is to find out the retention rate of the "Telco Card" customer group after each month since that customer first used the service in January.*

WITH jan\_table AS (

    SELECT DISTINCT(fact\_19.customer\_id) AS original\_user

    , month(transaction\_time) AS [month]

    FROM fact\_transaction\_2019 AS fact\_19

    LEFT JOIN dim\_status AS stat

    ON fact\_19.status\_id = stat.status\_id

    LEFT JOIN dim\_scenario as scen

    ON fact\_19.scenario\_id = scen.scenario\_id

    WHERE month(fact\_19.transaction\_time) = 1 AND fact\_19.status\_id = 1 AND scen.sub\_category = 'Telco Card'

)

, fy\_table AS (

    SELECT DISTINCT(fact\_19.customer\_id) AS original\_user

    , month(transaction\_time) AS [month1]

    FROM fact\_transaction\_2019 AS fact\_19

    LEFT JOIN dim\_status AS stat

    ON fact\_19.status\_id = stat.status\_id

    LEFT JOIN dim\_scenario as scen

    ON fact\_19.scenario\_id = scen.scenario\_id

    WHERE fact\_19.status\_id = 1 AND scen.sub\_category = 'Telco Card'

)

SELECT DISTINCT(month1) - 1 AS subsequent\_month

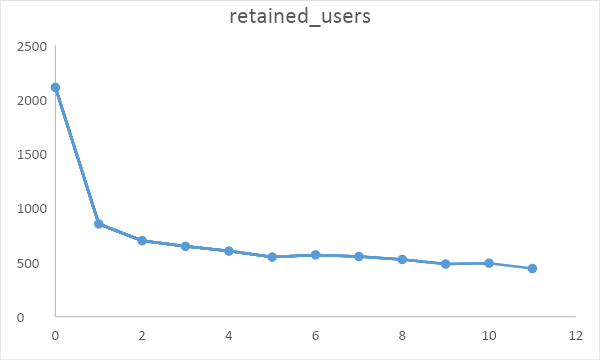
    , COUNT (jan\_table.original\_user) OVER ( PARTITION BY [month1]) AS retained\_users

FROM jan\_table

LEFT JOIN fy\_table

ON jan\_table.original\_user = fy\_table.original\_user

ORDER BY [month1] - 1  ASC



**Task B.**

You realize that the number of retained customers has decreased over time. Let’s calculate **retention** **= number of retained customers / total users of the first month**.

WITH jan\_table AS (

    SELECT DISTINCT(fact\_19.customer\_id) AS original\_user

    , month(transaction\_time) AS [month]

    FROM fact\_transaction\_2019 AS fact\_19

    LEFT JOIN dim\_status AS stat

    ON fact\_19.status\_id = stat.status\_id

    LEFT JOIN dim\_scenario as scen

    ON fact\_19.scenario\_id = scen.scenario\_id

    WHERE month(fact\_19.transaction\_time) = 1 AND fact\_19.status\_id = 1 AND scen.sub\_category = 'Telco Card'

)

, fy\_table AS (

    SELECT DISTINCT(fact\_19.customer\_id) AS original\_user

    , month(transaction\_time) AS [month1]

    FROM fact\_transaction\_2019 AS fact\_19

    LEFT JOIN dim\_status AS stat

    ON fact\_19.status\_id = stat.status\_id

    LEFT JOIN dim\_scenario as scen

    ON fact\_19.scenario\_id = scen.scenario\_id

    WHERE fact\_19.status\_id = 1 AND scen.sub\_category = 'Telco Card'

)

, joined\_table AS (

    SELECT DISTINCT(month1) - 1 AS subsequent\_month

    , COUNT (jan\_table.original\_user) OVER ( PARTITION BY [month1]) AS retained\_users

FROM jan\_table

LEFT JOIN fy\_table

ON jan\_table.original\_user = fy\_table.original\_user

)

SELECT \*

    , MAX(retained\_users) OVER () AS original\_users

    , FORMAT(retained\_users\*1.0/(MAX(retained\_users) OVER ())\*1.0, 'p') AS pct\_retained

FROM joined\_table

ORDER BY [subsequent\_month] - 1  ASC

## *Let’s see the desired outcome:*

