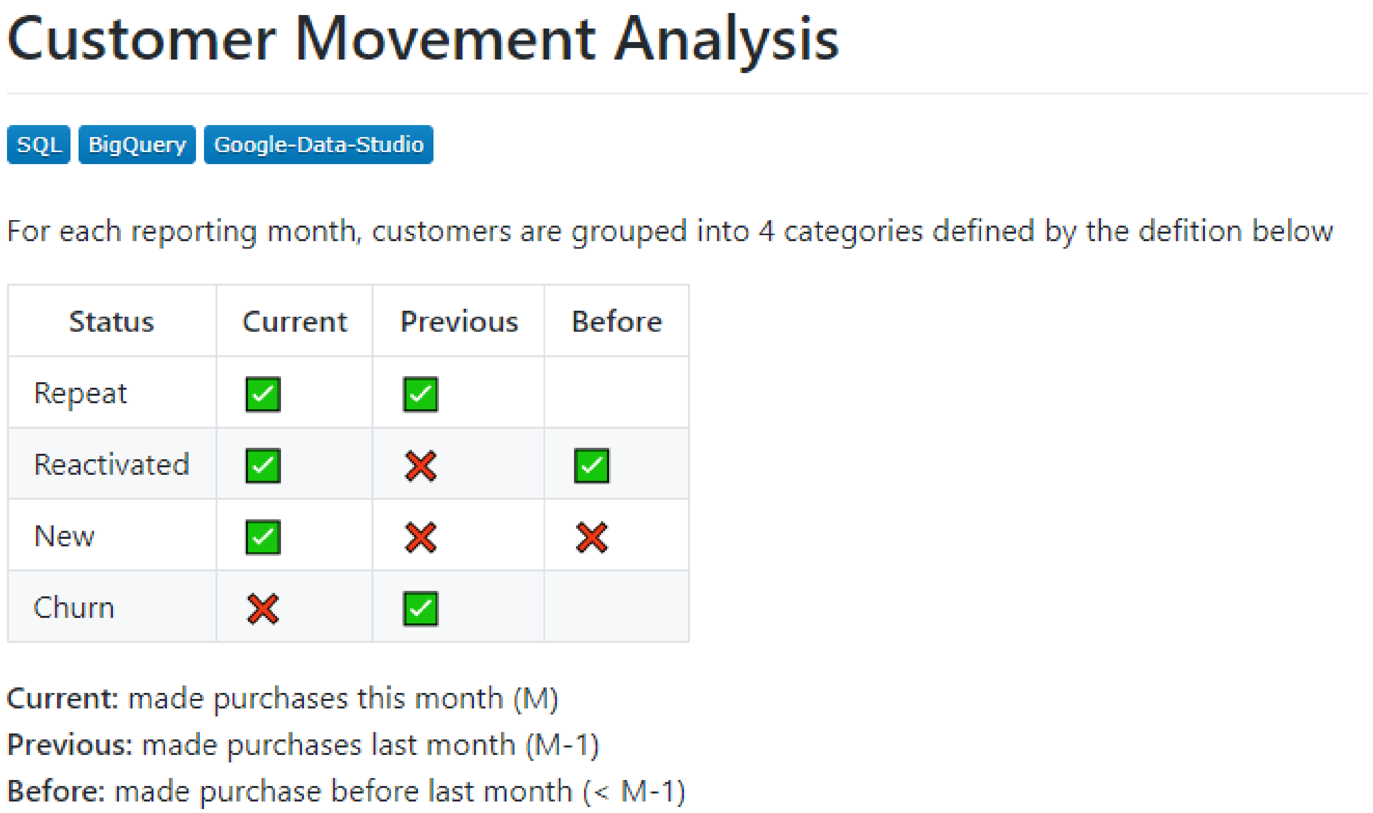
# *Customer Movement Analysis*



## Customer Movement Analysis

SQL BigQuery Google-Data-Studio

For each reporting month, customers are grouped into 4 categories defined by the defition below

| Status | Current | Previous | Before |
|---|---|---|---|
| Repeat | ✅ | ✅ | |
| Reactivated | ✅ | ❌ | ✅ |
| New | ✅ | ❌ | ❌ |
| Churn | ❌ | ✅ | |

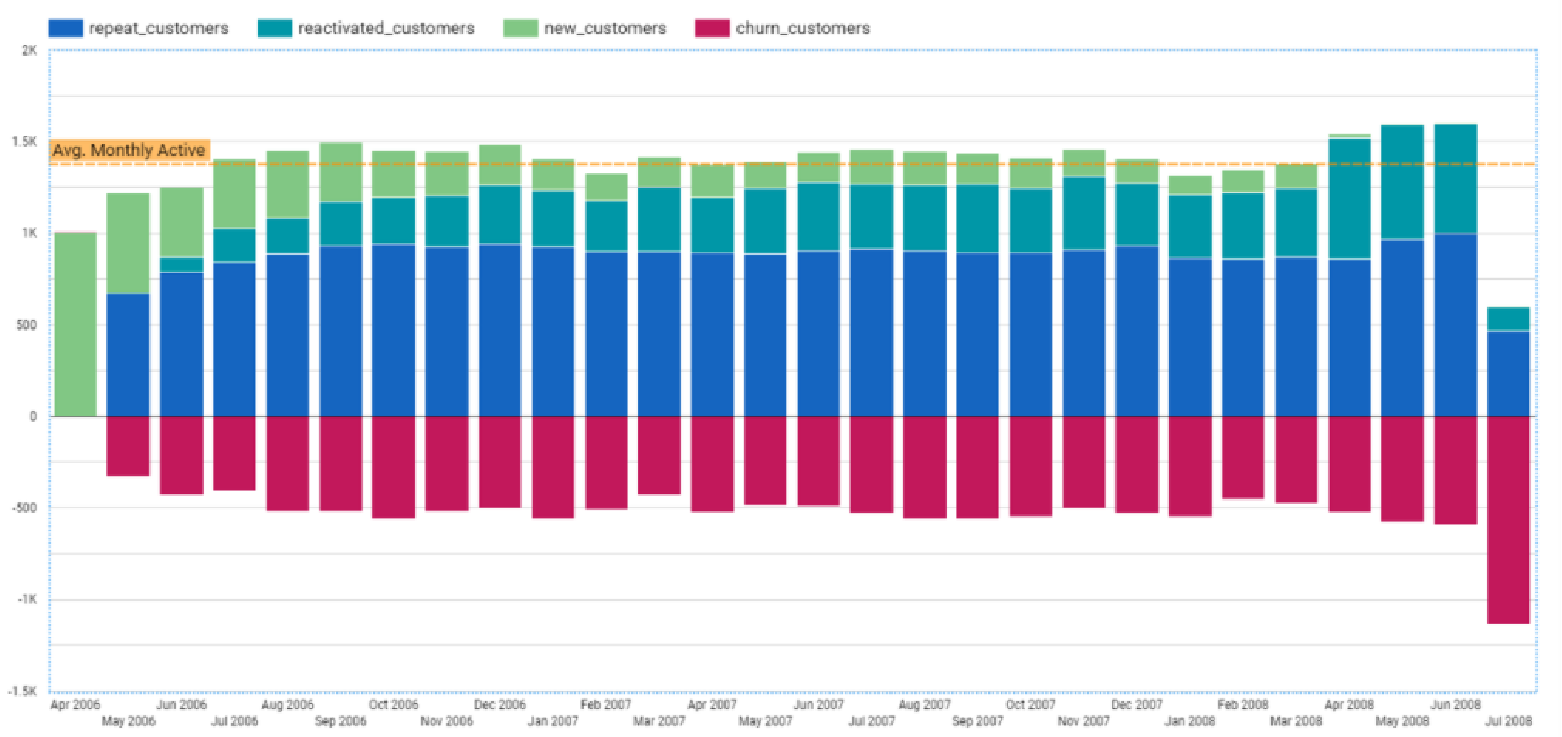
**Current:** made purchases this month (M)
**Previous:** made purchases last month (M-1)
**Before:** made purchase before last month (< M-1)

Reference : Tanat & Sukit

# *Customer Movement Analysis*

## Result

The visualization can be made right after executing the SQL using *Google Data Studio*.



Reference : Tanat & Sukit

# *Study  SQL command by following step-by-step*

## Path I : Select customer code and month for sub-query in next step

- Convert date int to datetime type

```
date_trunc(PARSE_DATE('%Y%m%d', CAST(shop_date AS STRING)), month)
```

- Select customer code and shop month for sub query in next step

```
WITH cust_month AS (
        select distinct cust_code, date_trunc(PARSE_DATE('%Y%m%d', CAST(shop_date AS STRING)), month) shop_month
        from `idyllic-creek-308203.BAD7105.supermarket`
        where cust_code is not null
)
```

# *Study  SQL command step-by-step*

## Path II : Create new columns for status of customer movement (New,Repeat,Reactived).

- Create Prev_month in partition table by LAG function (-1 month).

```
SELECT cust_code, shop_month, LAG(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month) AS prev_month
FROM cust_month
```

- Create status columns by Data_diff function between shop_month and prev_month.

```
SELECT cust_code, shop_month AS report_month, prev_month
,CASE
        WHEN DATE_DIFF(shop_month, prev_month, MONTH) IS NULL THEN 'New'
        WHEN DATE_DIFF(shop_month, prev_month, MONTH) = 1 THEN 'Repeat'
        WHEN DATE_DIFF(shop_month, prev_month, MONTH) > 1 THEN 'Reactivated'
ELSE NULL END AS status
FROM (
        SELECT cust_code, shop_month, LAG(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month) AS prev_month
        FROM cust_month
        )
```

# *Study  SQL command step-by-step*

## Path III : Create date diff between current month and next month.

- Create Prev_month in partition table by LAG function (-1 month).

```
SELECT cust_code, shop_month,LEAD(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month) AS next_trans_date
        ,DATE_DIFF(LEAD(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month), shop_month, MONTH) AS n_months
FROM cust_month
```

- Select churn status base on max(shop_month).

```
SELECT cust_code, DATE_ADD(shop_month, INTERVAL 1 MONTH) report_month, shop_month as prev_month,'Churn' AS status
FROM (
        SELECT cust_code, shop_month,LEAD(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month) AS next_trans_date
                ,DATE_DIFF(LEAD(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month), shop_month, MONTH) AS n_months
        FROM cust_month
             )
```

- Filter date diff > 1 or null value and excluded last month.

```
SELECT cust_code, DATE_ADD(shop_month, INTERVAL 1 MONTH) report_month, shop_month as prev_month,'Churn' AS status
FROM (
             SELECT cust_code, shop_month,LEAD(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month) AS next_trans_date
                     ,DATE_DIFF(LEAD(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month), shop_month, MONTH) AS
                      n_months
             FROM cust_month
         ) WHERE
             (n_months > 1 or n_months is null)
             AND shop_month < (SELECT MAX(shop_month) FROM cust_month)-- excluding last month
```

# *Study  SQL command step-by-step*

## Path IV : Union all status and then count no. of customer by status.

```
WITH cust_month AS (
    select distinct cust_code, date_trunc(PARSE_DATE('%Y%m%d', CAST(shop_date AS STRING)), month) shop_month
    from `idyllic-creek-308203.BAD7105.supermarket`
    where cust_code is not null
)
SELECT report_month
    ,COUNT(DISTINCT CASE WHEN status = 'New' THEN cust_code ELSE NULL END) AS new_customers
    ,COUNT(DISTINCT CASE WHEN status = 'Repeat' THEN cust_code ELSE NULL END) AS repeat_customers
    ,COUNT(DISTINCT CASE WHEN status = 'Reactivated' THEN cust_code ELSE NULL END) AS reactivated_customers
    ,-COUNT(DISTINCT CASE WHEN status = 'Churn' THEN cust_code ELSE NULL END) AS churn_customers
FROM (
    SELECT cust_code, shop_month AS report_month, prev_month
        ,CASE
            WHEN DATE_DIFF(shop_month, prev_month, MONTH) IS NULL THEN 'New'
            WHEN DATE_DIFF(shop_month, prev_month, MONTH) = 1 THEN 'Repeat'
            WHEN DATE_DIFF(shop_month, prev_month, MONTH) > 1 THEN 'Reactivated'
        ELSE NULL END AS status
    FROM (
        SELECT cust_code, shop_month, LAG(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month) AS prev_month
        FROM cust_month
    )
    UNION ALL
    SELECT cust_code, DATE_ADD(shop_month, INTERVAL 1 MONTH) report_month, shop_month as prev_month
        ,'Churn' AS status
    FROM (
        SELECT cust_code, shop_month
            ,LEAD(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month) AS next_trans_date
            ,DATE_DIFF(LEAD(shop_month, 1) OVER (PARTITION BY cust_code ORDER BY shop_month), shop_month, MONTH) AS n_months
        FROM cust_month
    ) WHERE
        (n_months > 1 or n_months is null)
        AND shop_month < (SELECT MAX(shop_month) FROM cust_month)-- excluding last month
) GROUP BY report_month
order by report_month
```