**Analytical SQL Case Study**

Company customers has purchasing transactions that we are monitoring them to get intuition behind each customer behavior to target the customers in the most efficient and proactive way to increase sales/revenue, improve customer retention and decrease churn.

The stakeholders wanted to know which month in the year made the most sales so they can stock up before, and also wanted to know which month is making the least sales so they reduce the human resources in these month:

* **What is the most and the least selling months?**

The most selling month is November and that was expected because November has many holidays and family time.

The least selling month is January and maybe that’s because people become bankrupt because of November.

**The Query:**

**SELECT DISTINCT TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'Month') month, SUM(price) OVER(PARTITION BY TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'Month')) total\_month\_sales**

**FROM tableretail**

**ORDER BY SUM(price) OVER(PARTITION BY TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'Month')) DESC;**

Then in order to support the marketing team decisions we also gave them more details about each customer purchase dates especially their first and last purchasing dates so they can manage their campaigns based on customers activity:

* **What was their first and last purchase date?**

We noticed that we get new customers more in November and December

**The Query:**

**SELECT DISTINCT CUSTOMER\_ID, FIRST\_VALUE(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI') IGNORE NULLS) OVER(PARTITION BY CUSTOMER\_ID ORDER BY TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')) ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS FIRST\_PURCHASE,**

**LAST\_VALUE(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI') IGNORE NULLS) OVER(PARTITION BY CUSTOMER\_ID ORDER BY TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')) ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS LAST\_PURCHASE**

**FROM tableretail;**

Now after understanding each customer behavior with us, we would like to go deeper in analysing their behavior so we would like to see how much did each customer spend with us to differentiate them as segments and assign a marketing campaign for each segment:

* **How much did each customer spend with us?**

Well, it really varies for each customer but the average for each customer total spent with us is 2324 L.E so we need to make customers segments to get more accurate result for each segment

**The Query:**

**SELECT DISTINCT CUSTOMER\_ID, SUM(PRICE \* QUANTITY) OVER (PARTITION BY CUSTOMER\_ID) AS TOTAL\_SPENT**

**FROM tableretail**

**ORDER BY SUM(PRICE \* QUANTITY) OVER (PARTITION BY CUSTOMER\_ID) DESC;**

And while analysing customers behavior the stakeholders wondered if we are really making a good and stable revenue? So to answer this question we had to compare the sales in each month and compare it with the previous month of each month:

* **What is the change in month over month in sales?**

Good news! Yes we are making a good profit except in December we had a loss of

-6567 which is pretty big loss! We need to investigate more about it.

**The Query:**

**SELECT TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'fmMonth yyyy') Month, EXTRACT(MONTH FROM TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')) AS Month\_number, SUM(price) total\_sales, LAG(SUM(price)) OVER(ORDER BY TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'fmMonth yyyy')) AS last\_month\_sales,**

**SUM(price) - LAG(SUM(price)) OVER(ORDER BY EXTRACT(MONTH FROM TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'))) AS change**

**FROM tableretail**

**GROUP BY TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'fmMonth yyyy'), EXTRACT(MONTH FROM TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'))**

**ORDER BY EXTRACT(MONTH FROM TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')) DESC;**

While understanding our customers behavior we noticed that we have some CHAMPIONS not just normal customers! Their purchases from us are insane! So we were very curious to know how much do these champions spend with us:

* **Total sales of the highest 10% customers in sales?**

Champions spend with us more than 83204.51, keep going champs! We need to do our best to keep them.

**The Query:**

**SELECT SUM(TOTAL\_SALES) total\_sales\_highest**

**FROM (**

**SELECT CUSTOMER\_ID, SUM(QUANTITY \* PRICE) AS TOTAL\_SALES, PERCENT\_RANK() OVER(ORDER BY SUM(price) DESC) AS its\_a\_rank**

**FROM tableretail**

**GROUP BY CUSTOMER\_ID**

**)**

**WHERE its\_a\_rank <= 0.1;**

Now, and after answering some initial business questions we were asked from the marketing team to implement a Monetary model for customers behavior for product purchasing and segment each customer based on Recency, Frequency and Monetary and then labeling them based on their score to be : **( Champions - Loyal Customers - Potential Loyalists – Recent Customers – Promising - Customers Needing Attention - At Risk – Can’t Lose Them – Hibernating – Lost )**

**The Query:**

**WITH cte AS (**

**SELECT**

**customer\_id,**

**recency,**

**frequency,**

**monetary,**

**NTILE(5) OVER (ORDER BY recency) AS r\_score,**

**NTILE(5) OVER (ORDER BY TRUNC((frequency + monetary)/2)) AS fm\_score**

**FROM (**

**SELECT DISTINCT**

**customer\_id,**

**last\_value(to\_date(InvoiceDate, 'mm-dd-yyyy HH24:MI') IGNORE NULLS)**

**OVER(PARTITION BY customer\_id ORDER BY to\_date(InvoiceDate, 'mm-dd-yyyy HH24:MI')**

**ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS last\_purchase,**

**ROUND( (SELECT MAX(to\_date(InvoiceDate, 'mm-dd-yyyy HH24:MI')) FROM tableretail) - last\_value(to\_date(InvoiceDate, 'mm-dd-yyyy HH24:MI') IGNORE NULLS) OVER(PARTITION BY customer\_id ORDER BY to\_date(InvoiceDate, 'mm-dd-yyyy HH24:MI') ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)) AS recency,**

**COUNT(INVOICE) OVER(PARTITION BY CUSTOMER\_ID) AS frequency,**

**SUM(PRICE\*QUANTITY) OVER(PARTITION BY CUSTOMER\_ID) AS monetary**

**FROM tableRetail**

**)**

**)**

**SELECT customer\_id, recency, frequency, monetary, NTILE(5) OVER (ORDER BY recency) AS r\_score,**

**NTILE(5) OVER (ORDER BY TRUNC((frequency + monetary)/2)) AS fm\_score,**

**CASE**

**WHEN (r\_score = 5 AND fm\_score IN (4, 5)) OR**

**(r\_score = 4 AND fm\_score = 5) THEN 'Champions'**

**WHEN (r\_score = 5 AND fm\_score = 2) OR**

**(r\_score = 4 AND fm\_score = 2) OR**

**(r\_score = 3 AND fm\_score IN (3, 4)) THEN 'Potential Loyalists'**

**WHEN (r\_score = 5 AND fm\_score = 3) OR**

**(r\_score = 4 AND fm\_score = 4) OR**

**(r\_score = 3 AND fm\_score = 5) THEN 'Loyal Customers'**

**WHEN r\_score = 5 AND fm\_score = 1 THEN 'Recent Customers'**

**WHEN r\_score = 4 AND fm\_score = 1 THEN 'Promising'**

**WHEN (r\_score = 3 AND fm\_score = 2) OR**

**(r\_score = 2 AND fm\_score = 3) OR**

**(r\_score = 2 AND fm\_score = 2) THEN 'Customers Needing Attention'**

**WHEN (r\_score = 2 AND fm\_score IN (4, 5)) OR**

**(r\_score = 1 AND fm\_score = 3) THEN 'At Risk'**

**WHEN (r\_score = 1 AND fm\_score IN (4, 5)) THEN 'Cant Lose Them'**

**WHEN r\_score = 1 AND fm\_score = 2 THEN 'Hibernating'**

**WHEN r\_score = 1 AND fm\_score = 1 THEN 'Lost'**

**ELSE NULL**

**END AS segment**

**FROM cte;**

After knowing that we have champions we kept digging in the data and found that we also have LOYAL customers! They have a pattern which is purchasing in consecutive days and that’s our favorite type of customers, stakeholders wondered what is the maximum day streaks did a loyal customer make?

* **Now we would like to know what is the maximum number of consecutive days a customer made purchases?**

It’s 61 days! A great streak for loyal customers! they deserve the title.

**The Query:**

**SELECT cust\_id, MAX(consecutive\_days) AS max\_consecutive\_days**

**FROM (**

**SELECT cust\_id, CALENDAR\_DT,**

**SUM(CASE WHEN CALENDAR\_DT = prev\_purchase\_date + 1 THEN 1 ELSE 0 END) OVER (PARTITION BY cust\_id ORDER BY CALENDAR\_DT) +1 AS consecutive\_days**

**FROM (**

**SELECT cust\_id, CALENDAR\_DT,**

**LAG(CALENDAR\_DT) OVER (PARTITION BY cust\_id ORDER BY CALENDAR\_DT) AS prev\_purchase\_date**

**FROM transactions )**

**)**

**GROUP BY cust\_id**

**ORDER BY MAX(consecutive\_days) DESC;**

The last question we had received from the stakeholders was about knowing how many days or transactions does a customer take to reach a total spent of 250 L.E, this question was a bit unclear as our customers segments variesfrom each other a lot but here is our result of investigating this question:

* **How many days/transactions does it take a customer to reach a spent threshold of 250 L.E?**

It Takes an average of 6 transactions or 11 days for a customer to reach a total spent of 250 L.E! it’s pretty long for a threshold of 250 L.E but we will be working on improving it by more successful marketing campaigns

**The Query:**

**SELECT AVG(num\_transactions) as avg\_transaction , AVG(num\_days) avg\_days**

**FROM (**

**SELECT DISTINCT cust\_id,**

**MIN(rn) OVER (PARTITION BY cust\_id ORDER BY calendar\_dt) AS num\_transactions,**

**MIN(num\_days) OVER (PARTITION BY cust\_id ORDER BY calendar\_dt) AS num\_days**

**FROM (**

**SELECT transactions.\*,**

**SUM(amt\_le) OVER (PARTITION BY cust\_id ORDER BY calendar\_dt) AS its\_a\_sum,**

**ROW\_NUMBER() OVER (PARTITION BY cust\_id ORDER BY calendar\_dt) AS rn,**

**calendar\_dt - MIN(calendar\_dt) OVER (PARTITION BY cust\_id ORDER BY calendar\_dt) AS num\_days**

**FROM transactions**

**)**

**WHERE its\_a\_sum >= 250**

**) ;**