Analytical Case Study:

(Unlocking Retail Insights: A Data-Driven Journey)

In the busy world of retail, understanding customer behavior is key to success. As the owner of a thriving retail business, I've always been keen on making data-driven decisions. After all, the more I know about my customers, the better I can serve them.

Query 1: Customer Insights

The first query was all about understanding the customers better. By analyzing their purchasing patterns, aimed to answer question like: Who are your most valuable customers? This information is invaluable for tailoring your product offerings and marketing strategies.

Query 2: Stock Insights

The second query was all about understanding the stock better. By analyzing the quantity sold for each stock code, I wanted to answer questions like: Which products are the most popular? Which products should you stock more of? This information is invaluable for optimizing the stock inventory and ensuring that I always have the right products in stock.

Query 3: Stock Spending Insights

The third query was all about understanding how much customers are spending on different products. By analyzing the total amount spent on each stock code, I was able to answer questions like: Which products are the most profitable? Which products are the most popular among high-spending customers? This information is invaluable for optimizing the product offerings and ensuring that we are maximizing the profits.

Query 4: Yearly Insights

The fourth query was all about understanding how the customers' spending habits change from year to year. By analyzing the total amount spent by customers in each year, I aimed to answer questions like: How has the total amount spent by customers changed over time? Which years saw the highest total amount spent by customers? This information is invaluable for understanding the long-term trends in customers' spending habits and for planning our marketing and sales strategies accordingly.

Query 5: Customer Purchase Insights

The fifth query was all about understanding how often the customers make purchases. By analyzing the first and last purchase dates for each customer, I wanted to answer questions like: How often do the customers make purchases? And, how do the purchase habits of different customer segments differ? This information is crucial for understanding the customers' purchase habits and for planning the marketing and sales strategies accordingly.

Part 2: Customer Segmentation

The second part delved into customer segmentation. I realized that not all customers are the same. Some are our loyal champions, while others might be at risk of churning. By segmenting the customers based on their recency, frequency, and monetary scores, we could tailor our marketing efforts to each group. This approach would help us retain the most valuable customers while also identifying opportunities to re-engage with others.

Part 3: Loyalty and Engagement

Finally, the third part focused on customer loyalty and engagement. We wanted to identify customers who were making consecutive purchases, as well as understand the average transaction frequency for our most loyal customers. This information would help us design loyalty programs that reward your most engaged customers, while also identifying opportunities to increase engagement with other segments.

The Outcome: A Data-Driven Strategy

By running these queries and analyzing the results, we gained invaluable insights into our customers' behavior. Armed with this knowledge, we were able to tailor our product offerings, marketing strategies, and loyalty programs to better meet your customers' needs. The result? A more engaged and loyal customer base, increased sales, and a thriving retail business.

In the fast-paced world of retail, staying ahead of the curve is crucial. And with data as our guide, we’re well on our way to continued success.

This story highlights the importance of data-driven decision-making in retail and how the queries I made helped us gain insights into our customers' behavior, leading to a more successful business strategy.

Queries Delivery

Q1-

1-
```
select customer_id, total_quantity,total_spent, quantity_per_customer_rank
from (select customer_id, sum(quantity) total_quantity, sum(price*quantity) total_spent, rank()
over(order by sum(quantity) desc) quantity_per_customer_rank from tableretail group by
customer_id );
```

2-
```
select stockcode, total_quan,quantity_per_stock_rank
from (select stockcode, sum(quantity) total_quan, rank() over(order by sum(quantity)desc)
quantity_per_stock_rank from tableretail group by stockcode );
```

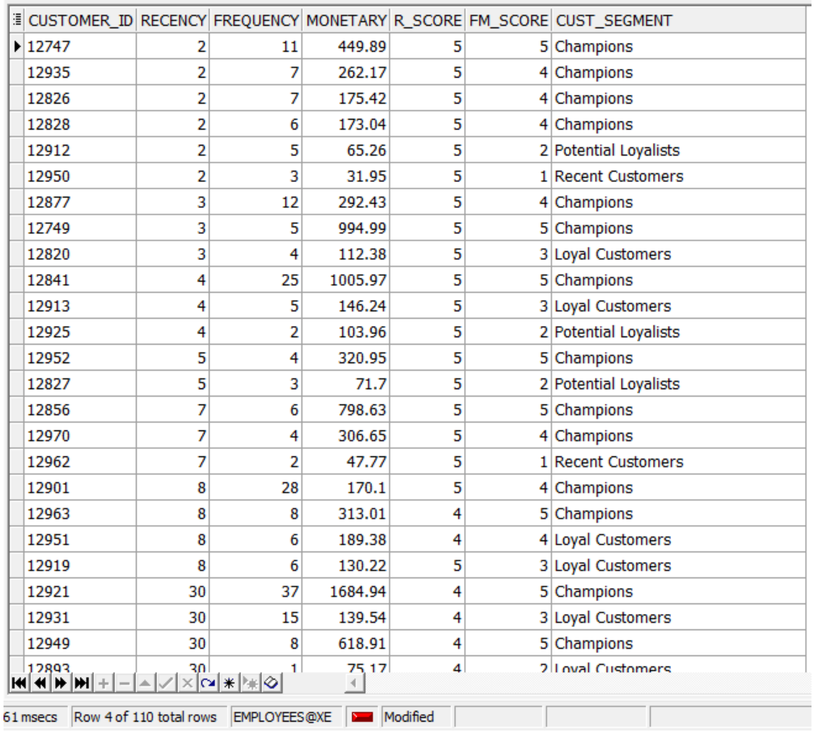
3-
```
select stockcode, total_quantity, total_amount, spent_on_stock_rank
from (select stockcode, sum(quantity) total_quantity, sum(price*quantity) total_amount, rank()
over(order by sum(price*quantity)desc) spent_on_stock_rank from tableretail group by
stockcode );
```

4-
```
select years, total_amount, rank
from(select extract (year from to_date(substr(invoicedate,0,9),'mm/dd/yyyy')) years,
sum(price*quantity) total_amount, rank() over (order by sum(price*quantity)desc) rank from
tableretail group by extract (year from to_date(substr(invoicedate,0,9),'mm/dd/yyyy')));
```

5-
```
select customer_id, first_purchase, last_purchase
from (select customer_id, first_value(to_date(substr(invoicedate,0,9), 'mm/dd/yyyy'))
over(partition by customer_id order by to_date(substr(invoicedate,0,9), 'mm/dd/yyyy'))
first_purchase, last_value(to_date(substr(invoicedate,0,9), 'mm/dd/yyyy')) over(partition by
customer_id order by to_date(substr(invoicedate,0,9), 'mm/dd/yyyy') rows between unbounded
preceding and unbounded following ) last_purchase from tableretail)
group by customer_id, first_purchase, last_purchase
order by customer_id;
```

Q2-

```
select fin.*, case
when ( r_score=5 and (fm_score=5 or fm_score =4))  or  ( r_score=4 and fm_score=5) then
'Champions'
when  ( r_score=3 and (fm_score=5 or fm_score =4)) or ( r_score=4 and r_score=4) or ( r_score=5 and
fm_score=3) then 'Loyal Customers'
when   ( r_score=4 and (fm_score=2 OR fm_score =3)) or  ( r_score=3 and r_score=3) or ( r_score=5
and fm_score=2) then 'Potential Loyalists'
when  (FM_SCORE=1 and ( r_score=3 or r_score=4)) then 'Promising'
when ( r_score=5 and fm_score =1) then 'Recent Customers'
when ( r_score=3 and fm_score =2) or  ( r_score=2 and (fm_score=3 or fm_score =2)) then 'Customers
Needing Attention'
when ( r_score=2 and (fm_score=5 or fm_score =4)) or ( r_score=1 and fm_score =3) then 'At Risk'
when r_score=1 and (fm_score=4 or fm_score =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 'UNKOWN'
end as cust_segment
from (
select customer_id, recency, frequency, monetary, ntile (5) over(order by recency desc) r_score, ntile(5)
over(order by avg) fm_score from(
select customer_id, recency, frequency, monetary, (frequency+monetary)/2 avg
from(
select customer_id, recency, count(distinct invoice) frequency, sum(price) monetary
from (select customer_id, invoice, price, (to_date('12/9/2011','MM/DD/YYYY') - last_value
(to_date(substr(invoicedate,0,9),'MM/DD/YYYY')) over (partition by customer_id order by
to_date(substr(invoicedate,0,9),'MM/DD/YYYY') rows between unbounded preceding and unbounded
following)) recency from tableretail)
group by customer_id, recency)
group by customer_id, recency, frequency, monetary)) fin
order by recency, frequency desc,monetary desc;
```

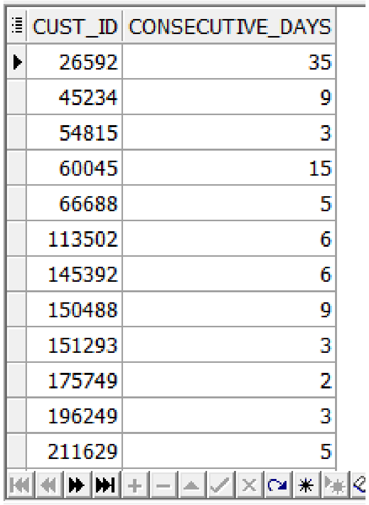


CUSTOMER_ID	RECENCY	FREQUENCY	MONETARY	R_SCORE	FM_SCORE	CUST_SEGMENT
12747	2	11	449.89	5	5	Champions
12935	2	7	262.17	5	4	Champions
12826	2	7	175.42	5	4	Champions
12828	2	6	173.04	5	4	Champions
12912	2	5	65.26	5	2	Potential Loyalists
12950	2	3	31.95	5	1	Recent Customers
12877	3	12	292.43	5	4	Champions
12749	3	5	994.99	5	5	Champions
12820	3	4	112.38	5	3	Loyal Customers
12841	4	25	1005.97	5	5	Champions
12913	4	5	146.24	5	3	Loyal Customers
12925	4	2	103.96	5	2	Potential Loyalists
12952	5	4	320.95	5	5	Champions
12827	5	3	71.7	5	2	Potential Loyalists
12856	7	6	798.63	5	5	Champions
12970	7	4	306.65	5	4	Champions
12962	7	2	47.77	5	1	Recent Customers
12901	8	28	170.1	5	4	Champions
12963	8	8	313.01	4	5	Champions
12951	8	6	189.38	4	4	Loyal Customers
12919	8	6	130.22	5	3	Loyal Customers
12921	30	37	1684.94	4	5	Champions
12931	30	15	139.54	4	3	Loyal Customers
12949	30	8	618.91	4	5	Champions
12893	30	1	75.17	4	2	Loyal Customers

61 msecs | Row 4 of 110 total rows | EMPLOYEES@XE | Modified

Q3-

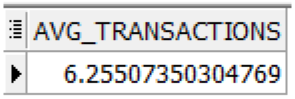
```
with cons_days as (select cust_id, row_number() over(partition by cust_id, cons_days order by
calendar_dt) cons
from (select cust_id, calendar_dt, calendar_dt - row_number() over (partition by cust_id order by
calendar_dt) cons_days
from customers))
select cust_id, max(cons) consecutive_days
from cons_days
group by cust_id;
```



CUST_ID	CONSECUTIVE_DAYS
26592	35
45234	9
54815	3
60045	15
66688	5
113502	6
145392	6
150488	9
151293	3
175749	2
196249	3
211629	5

```
with TransactionAvg as(
select cust_id, min(row_num) frist_250_or_more
from(select cust_id, calendar_dt, cumm_amount, row_number() over (partition by cust_id order by
calendar_dt) row_num
from(select cust_id, calendar_dt, sum(amt_le) over(partition by cust_id order by calendar_dt)
cumm_amount from customers))
where cumm_amount >= 250
group by cust_id)

select avg(frist_250_or_more) avg_transactions
from TransactionAvg;
```



AVG_TRANSACTIONS
6.25507350304769