Analytical SQL Case Study

First Question:

1- Top Selling Products By Sales

```sql
-- top selling products by sales
SELECT STOCKCODE,SUM(QUANTITY*PRICE) AS TOTAL_SOLD
FROM tableretail
GROUP BY STOCKCODE
ORDER BY TOTAL_SOLD DESC;
```

Output

STOCKCODE	TOTAL_SOLD
84879	9114.69
22197	4323.1
21787	4059.35
22191	3461.2
23203	3357.44
21479	2736.01
23215	2697.36
22970	2493.6
22570	2458.08
22992	2308.05

Business Meaning

The business meaning behind this query is to identify the best-performing products in terms of sales revenue. By aggregating sales data and sorting products based on their total sales, businesses can gain insights into which products are most popular or lucrative. This information enables strategic decision-making, such as optimizing inventory management, identifying trends, and focusing marketing efforts on high-demand items. Ultimately, it helps businesses maximize profitability and allocate resources effectively.

2-Top Selling Products By Quantity

```
-------Top Selling Products By Quantity
select distinct stockcode ,
sum(Quantity) over(partition by stockcode) as Total_Quantity_by_Product
from tableretail
order by Total_Quantity_by_Product DESC;
```

Output

STOCKCODE	TOTAL_QUANTITY_BY_PRODUCT
84077	7824
84879	6117
22197	5918
21787	5075
21977	4691
21703	2996
17096	2019
15036	1920
23203	1803
21790	1579

Business Meaning

The business meaning behind this query is to identify the top-selling products based on the total quantity sold. By aggregating sales data and calculating the total quantity of each product sold, businesses can determine which items are in high demand among customers. This information is essential for inventory management, as it allows businesses to ensure they have sufficient stock of popular products to meet customer demand. Additionally, it provides insights into consumer preferences and helps businesses tailor their product offerings and marketing strategies accordingly, ultimately driving sales and profitability.

3-Top Selling Products By Sales (Different Prices)

```
SELECT distinct STOCKCODE,
sum(QUANTITY) over(partition by stockcode,price) as Quantity , PRICE ,
SUM(QUANTITY*PRICE) over(partition by stockcode,price) AS TOTAL_SOLD
FROM tableretail
ORDER BY TOTAL_SOLD DESC;
```

Output

STOCKCODE	QUANTITY	PRICE	TOTAL_SOLD
84879	5096	1.45	7389.2
22197	5440	0.72	3916.8
22191	438	7.65	3350.7
21787	3803	0.85	3232.55
23215	1400	1.79	2506
23203	1350	1.79	2416.5
22570	672	3.39	2278.08
22970	1032	2.1	2167.2
22569	576	3.39	1952.64
22991	1152	1.65	1900.8
22992	1140	1.65	1881
23084	1020	1.79	1825.8
48138	230	7.65	1759.5
84879	1021	1.69	1725.49

Business Meaning

The business meaning behind this query is to analyze the sales performance of products at different price points. By partitioning the data based on both the stock code and price of each product, the query calculates the total quantity sold, total sales revenue, and quantity sold for each unique combination of product and price. This analysis provides insights into how variations in pricing impact product sales. Businesses can use this information to optimize pricing strategies, identify price-sensitive customer segments, and determine the most profitable price points for their products. Additionally, it helps businesses understand the demand elasticity of their products and make informed decisions to maximize revenue and profitability.

4- Number of Customers Per Month

```
select distinct TO_CHAR(TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI'),'MM/YYYY') as INVOICEDATE ,
 count(distinct customer_id) over (partition by TO_CHAR(TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI'),'MM/YYYY')) as NumberOfCustomers
from tableretail
order by INVOICEDATE;
```

Output

INVOICEDATE	NUMBEROFCUSTOMERS
01/2011	22
02/2011	21
03/2011	31
04/2011	15
05/2011	34
06/2011	25
07/2011	24
08/2011	20
09/2011	37
10/2011	30
11/2011	45
12/2010	24
12/2011	24

Business Meaning

Monthly Customer Count: By grouping transactions based on the month and year they occurred, the query provides insights into customer activity trends over time. It helps businesses understand variations in customer engagement and identify peak periods of customer interaction.

Customer Retention Analysis: Tracking the number of unique customers each month enables businesses to analyze customer retention and churn rates. Fluctuations in the number of customers from month to month may indicate changes in customer behavior or satisfaction levels.

Marketing and Sales Insights: Monthly customer counts offer valuable information for marketing and sales strategies. Businesses can tailor promotional campaigns, product launches, and sales initiatives based on customer engagement patterns observed over different months.

Performance Evaluation: Comparing monthly customer counts over time allows businesses to evaluate the effectiveness of marketing efforts and customer acquisition strategies. It helps assess the impact of various initiatives on customer acquisition and retention rates.

Forecasting and Planning: Analyzing trends in monthly customer counts facilitates forecasting future customer demand and revenue projections. Businesses can use this information to make informed decisions regarding resource allocation, inventory management, and strategic planning.

Overall, this analysis provides actionable insights into customer behavior, enabling businesses to optimize their operations, enhance customer relationships, and drive sustainable growth.

5- Customer Lifetime Value

```sql
SELECT
  distinct CUSTOMER_ID,
  LAST_VAL,
  FIRST_VAL,
  DIFF,
  TOTAL_SALES,
  round((TOTAL_SALES / NULLIF(DIFF, 0)),2) as CLV -- Added NULLIF to handle division by zero
  FROM (
  SELECT
    CUSTOMER_ID,
    LAST_VALUE(TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI')) OVER (PARTITION BY CUSTOMER_ID ORDER BY TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI')
    RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS LAST_VAL,
    FIRST_VALUE(TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI')) OVER (PARTITION BY CUSTOMER_ID ORDER BY TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI')) AS FIRST_VAL,
    LAST_VALUE(trunc(TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI'))) OVER (PARTITION BY CUSTOMER_ID ORDER BY TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI')
    RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) -
    FIRST_VALUE(trunc(TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI'))) OVER (PARTITION BY CUSTOMER_ID ORDER BY TO_DATE(INVOICEDATE,'MM/DD/YYYY HH24:MI')) AS DIFF,
    SUM(QUANTITY * PRICE) OVER (PARTITION BY CUSTOMER_ID) AS TOTAL_SALES FROM tableretail)
    order by customer_id;
```

Output

CUSTOMER_ID	LAST_VAL	FIRST_VAL	DIFF	TOTAL_SALES	CLV
12747	12/7/2011 2:34:00 PM	12/5/2010 3:38:00 PM	367	4196.01	11.43
12748	12/9/2011 12:20:00 PM	12/1/2010 12:48:00 PM	373	33719.73	90.4
12749	12/6/2011 9:56:00 AM	5/10/2011 3:25:00 PM	210	4090.88	19.48
12820	12/6/2011 3:12:00 PM	1/17/2011 12:34:00 PM	323	942.34	2.92
12821	5/9/2011 3:51:00 PM	5/9/2011 3:51:00 PM	0	92.72	
12822	9/30/2011 10:04:00 AM	9/13/2011 1:46:00 PM	17	948.88	55.82
12823	9/26/2011 7:35:00 AM	2/16/2011 12:15:00 PM	222	1759.5	7.93
12824	10/11/2011 12:49:00 PM	10/11/2011 12:49:00 PM	0	397.12	
12826	12/7/2011 10:25:00 AM	12/9/2010 3:21:00 PM	363	1474.72	4.06
12827	12/4/2011 12:17:00 PM	10/26/2011 3:44:00 PM	39	430.15	11.03
12828	12/7/2011 8:45:00 AM	8/1/2011 4:16:00 PM	128	1018.71	7.96
12829	1/7/2011 11:13:00 AM	12/14/2010 2:54:00 PM	24	293	12.21
12830	11/2/2011 11:54:00 AM	6/21/2011 10:53:00 AM	134	6814.64	50.86
12831	3/22/2011 1:02:00 PM	3/22/2011 1:02:00 PM	0	215.05	

Business Meaning

the query calculates Customer Lifetime Value (CLV), a vital metric indicating the total revenue a customer is expected to generate over their entire relationship with the business. CLV offers valuable insights:

Customer Spending: It quantifies the total amount a customer spends, providing a comprehensive view of their financial impact on the business.

Long-Term Value: CLV considers the entire duration of the customer relationship, offering insights beyond immediate transactions to assess long-term profitability.

Valuable Customer Identification: Higher CLV identifies customers who consistently contribute significant revenue, guiding businesses to prioritize resources and tailor strategies to retain and nurture these relationships.

Strategic Decision-Making: CLV informs decisions on customer acquisition costs, pricing strategies, and resource allocation, optimizing profitability and return on investment.

Customer Relationship Management: It serves as a key metric for CRM, enabling businesses to personalize engagement efforts, enhance customer satisfaction, and foster loyalty.

Overall, CLV analysis empowers businesses to cultivate lasting customer relationships, drive revenue growth, and maintain a competitive edge in the market.

6- Basket Analysis

```sql
with Transactions as (
    select Invoice, StockCode
    from tableRetail
    group by Invoice, StockCode
),
Basket_Analysis as (
    select t1.StockCode "First Item", t2.StockCode "Second Item", count(*) "Times Bought Together"
    from Transactions t1
    join Transactions t2 on t1.Invoice = t2.Invoice and t1.StockCode < t2.StockCode
    group by t1.StockCode, t2.StockCode
)
select "First Item", "Second Item", "Times Bought Together"
from Basket_Analysis
where "Times Bought Together" >= 8
order by "Times Bought Together" desc;
```

Output

First Item	Second Item	Times Bought Together
20724	22355	23
20725	20728	22
20725	22384	21
20719	22355	21
20725	22382	21
22355	22661	20
20719	20724	20
20724	22661	19
82482	82494L	19
20725	20726	19
20726	22382	18
20723	20724	18
23202	23203	18
20726	22384	18
23199	85099B	18
20723	22355	17

Business Meaning

Market Basket Analysis: Basket analysis is a valuable technique in retail for understanding customer purchasing behavior. By identifying which items are frequently bought together, businesses can gain insights into cross-selling opportunities, optimize product placement strategies, and design targeted promotions or product bundles to increase sales and enhance customer satisfaction.

Identifying Product Affinities: The query helps businesses identify associations between products that may not be immediately obvious. For example, it may reveal complementary items or products that are commonly used together, allowing businesses to create cohesive product offerings and improve the overall shopping experience.

Optimizing Inventory Management: Understanding which items are frequently bought together enables businesses to better manage inventory levels and stock replenishment. By stocking frequently co-purchased items closer together or ensuring adequate supply of these items, businesses can minimize stockouts, optimize shelf space, and improve operational efficiency.

Personalized Marketing: Insights from basket analysis can inform personalized marketing campaigns. Businesses can leverage knowledge of product associations to tailor recommendations, promotions, and advertisements to individual customers or segments, increasing the relevance and effectiveness of marketing efforts.

Overall, basket analysis provides actionable insights that can drive various aspects of retail operations, from product assortment and inventory management to marketing and sales strategies, ultimately contributing to improved customer satisfaction and business performance.

Second Question:

```
WITH RAC AS (
  SELECT
    Customer_ID,
    "Last Value",
    "Reference Date",
    trunc("Reference Date" - "Last Value") AS "Recency",
    "Frequency",
    round("monetary"/1000,2) AS "monetary"
  FROM (
    SELECT DISTINCT
      Customer_ID,
      LAST_VALUE(TO_DATE(invoicedate, 'mm/dd/yyyy HH24:MI')) OVER (ORDER BY TO_DATE(invoicedate, 'mm/dd/yyyy HH24:MI')
      RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS "Reference Date",
      LAST_VALUE(TO_DATE(invoicedate, 'mm/dd/yyyy HH24:MI')) OVER (PARTITION BY Customer_ID ORDER BY TO_DATE(invoicedate, 'mm/dd/yyyy HH24:MI')
      RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS "Last Value",
      COUNT(DISTINCT invoice) OVER (PARTITION BY customer_id) AS "Frequency",
      SUM(QUANTITY*PRICE) OVER (PARTITION BY Customer_ID) AS "monetary"
    FROM
      tableRetail
  )
)
SELECT Customer_ID,
  "Recency",
  "Frequency",
  "monetary",
  "r_score",
  "fm_score",
  CASE
    WHEN "r_score" = 5 AND "fm_score" IN (5, 4) THEN 'Champions'
    WHEN "r_score" = 4 AND "fm_score" = 5 THEN 'Champions'
    WHEN "r_score" = 5 AND "fm_score" = 2 THEN 'Potential Loyalists'
    WHEN "r_score" = 4 AND "fm_score" IN (2 , 3) THEN 'Potential Loyalists'
    WHEN "r_score" = 3 AND "fm_score" = 3 THEN 'Potential Loyalists'
    WHEN "r_score" = 5 AND "fm_score" = 3 THEN 'Loyal Customers'
    WHEN "r_score" = 4 AND "fm_score" = 4 THEN 'Loyal Customers'
```

```
    WHEN "r_score" = 3 AND "fm_score" IN (4 , 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" = 1 THEN 'Promising'
    WHEN "r_score" = 3 AND "fm_score" = 2 THEN 'Customers Needing Attention'
    WHEN "r_score" = 2 AND "fm_score" IN (2, 3) THEN 'Customers Needing Attention'
    WHEN "r_score" = 1 AND "fm_score" = 3 THEN 'At Risk'
    WHEN "r_score" = 2 AND "fm_score" IN (4, 5) THEN 'At Risk'
    WHEN "r_score" = 1 AND "fm_score" = 2 THEN 'Hibernating'
    WHEN "r_score" = 1 AND "fm_score" IN (4, 5) THEN 'Cant Lose Them'
    WHEN "r_score" = 1 AND "fm_score" = 1 THEN 'Lost'
    ELSE 'Undefined'
  END AS "cust_segment"
  -- AVG("Frequency" +"monetary") as fm
  from (select NTILE(5) OVER (ORDER BY "fm_avg" DESC) AS "fm_score",Customer_ID,
  "Recency",
  "Frequency",
  "monetary",
  "r_score"
  from (select
  Customer_ID,
  "Recency",
  "Frequency",
  "monetary",
  NTILE(5) OVER (ORDER BY "Recency" DESC) AS "r_score",
  --ntile(5) over(order by "Frequency" desc) as "f_score",
  --ntile(5) over(order by "monetary" desc) as "m_score",
  ("Frequency" + "monetary") / 2 AS "fm_avg"
FROM
  RAC
ORDER BY
  Customer_ID));
```

Output

CUSTOMER_ID	Recency	Frequency	monetary	r_score	fm_score	cust_segment
12748	0	210	33.72	5	1	Recent Customers
12931	20	15	42.06	4	1	Promising
12921	8	37	16.59	4	1	Promising
12971	167	45	5.19	2	1	Undefined
12901	8	28	17.65	4	1	Promising
12841	4	25	4.02	5	1	Recent Customers
12839	1	14	5.59	5	1	Recent Customers
12939	63	8	11.58	3	1	Promising
12955	0	11	4.76	5	1	Recent Customers
12747	1	11	4.2	5	1	Recent Customers
12877	3	12	1.54	5	1	Recent Customers
12830	37	6	6.81	3	1	Promising
12949	30	8	4.17	3	1	Promising
12957	9	8	4.02	4	1	Promising
12910	22	8	3.08	4	1	Promising
12867	25	7	4.04	4	1	Promising

Business Meaning

This query calculates RFM (Recency, Frequency, Monetary) scores and segments customers based on these scores. Here's the breakdown of the business meaning:

1. RAC CTE (Recent, Frequency, Monetary):

   - This part calculates the recency, frequency, and monetary values for each customer. Recency indicates how recently a customer made a purchase, frequency represents the number of purchases, and monetary indicates the total amount spent by the customer.

2. Main Query:

- The main query calculates RFM scores and segments customers based on these scores.

- RFM scores are calculated by dividing customers into quintiles (5 groups) based on their recency and the average of frequency and monetary values.

- Customers are segmented into different categories based on their RFM scores. Each segment represents a different level of customer engagement and potential value to the business.

3. Customer Segmentation:

- Customers are segmented into categories such as "Champions," "Loyal Customers," "Potential Loyalists," "Recent Customers," "Promising," "Customers Needing Attention," "At Risk," "Hibernating," "Can't Lose Them," and "Lost" based on their RFM scores.

- These segments help businesses understand the characteristics and behaviors of different customer groups and tailor marketing strategies, retention efforts, and customer service initiatives accordingly.

4. Business Implications:

- Customer segmentation based on RFM scores allows businesses to prioritize their efforts and resources effectively. For example, they can focus on retaining high-value customers ("Champions" and "Loyal Customers") by offering personalized incentives or enhancing their customer experience.

- Segmentation also helps identify at-risk customers ("At Risk" and "Hibernating") who may need targeted interventions to prevent churn.

- By understanding customer segments, businesses can tailor communication strategies, product offerings, and promotions to better meet the needs and preferences of different customer groups, ultimately driving customer satisfaction and loyalty.

Overall, this RFM analysis provides valuable insights into customer behavior and enables businesses to make data-driven decisions to optimize customer engagement, retention, and profitability.

Third Question:

1)

```
WITH PURCHASE_DATA AS (
    SELECT
        cust_id,
        Calendar_Dt,
        ROW_NUMBER() OVER (PARTITION BY cust_id ORDER BY Calendar_Dt) AS rn
    FROM
        customers_amount
),
consecutive_days  AS (
    SELECT
        cust_id,
        Calendar_Dt,
        Calendar_Dt - rn AS date_diff
    FROM
        PURCHASE_DATA
)
SELECT
    cust_id,
    MAX(consecutive_days) AS max_consecutive_days
FROM (
    SELECT
        cust_id,
        COUNT(date_diff) AS consecutive_days
    FROM
        consecutive_days
    GROUP BY
        cust_id, date_diff
)
GROUP BY
    cust_id
ORDER BY
    cust_id;
```

Output

CUST_ID	MAX_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
217534	25
232210	6
233119	2
247965	2

This query analyzes customer purchase data to determine the maximum number of consecutive days each customer made purchases. Here's the breakdown of its business meaning:

Business Meaning

1. PURCHASE_DATA CTE:

- This part selects data from the "customers_amount" table, which likely contains information about customer IDs, calendar dates, and possibly purchase amounts.

- It assigns a row number to each record within each customer partition, ordered by calendar date. This row number (rn) helps identify consecutive days.

2. consecutive_days CTE:

- This section calculates the difference in calendar dates between consecutive purchases for each customer. It determines the gap between purchase dates, indicating consecutive days of purchase activity.

3. Main Query:

- This part of the query calculates the maximum number of consecutive days each customer made purchases.

- It counts the occurrences of consecutive purchase days for each customer and selects the maximum value.

4. Business Implications:

- Understanding the maximum consecutive days of purchases for each customer provides insights into their purchasing behavior patterns.

- Customers with a high number of consecutive purchase days may exhibit consistent buying habits and higher engagement with the business.

- Businesses can leverage this information to tailor marketing strategies, promotions, and loyalty programs to retain and further engage customers who demonstrate consistent purchasing behavior.

- Identifying customers with longer spans of consecutive purchases may also help predict future buying patterns and optimize inventory management and resource allocation accordingly.

Overall, this analysis helps businesses gain a deeper understanding of customer purchasing behavior and provides actionable insights to enhance customer engagement and drive sales.

2)

```
WITH purchase_data AS (
  SELECT
    CUST_ID,
    CALENDAR_DT AS PURCHASE_DATE,
    AMT_LE,
    SUM(AMT_LE) OVER (PARTITION BY CUST_ID ORDER BY CALENDAR_DT) AS CUMULATIVE_AMT
  FROM
    customers_amount
)
SELECT
  AVG(DAYS_TO_THRESHOLD) AS AVG_DAYS_TO_THRESHOLD,
  trunc(AVG(TRANSACTIONS_TO_THRESHOLD)) AS AVG_TRANSACTIONS_TO_THRESHOLD
FROM (
  SELECT
    CUST_ID,
    MIN(DAYS_TO_THRESHOLD) AS DAYS_TO_THRESHOLD,
    MIN(TRANSACTIONS_TO_THRESHOLD) AS TRANSACTIONS_TO_THRESHOLD
```

```
  FROM (
    SELECT
      CUST_ID,
      MIN(DAYS) AS DAYS_TO_THRESHOLD,
      MIN(TRANSACTIONS) AS TRANSACTIONS_TO_THRESHOLD
    FROM (
      SELECT
        CUST_ID,
        PURCHASE_DATE - FIRST_PURCHASE_DATE AS DAYS,
        ROW_NUMBER() OVER (PARTITION BY CUST_ID ORDER BY PURCHASE_DATE) AS TRANSACTIONS
      FROM (
        SELECT
          CUST_ID,
          PURCHASE_DATE,
          CUMULATIVE_AMT,
          MIN(CASE WHEN CUMULATIVE_AMT >= 250 THEN PURCHASE_DATE END)
          OVER (PARTITION BY CUST_ID ORDER BY PURCHASE_DATE) AS FIRST_PURCHASE_DATE
```

```
        FROM
          purchase_data
      )
    )
    WHERE DAYS IS NOT NULL
    GROUP BY CUST_ID
  )
  GROUP BY CUST_ID
)
```

Output

AVG_TRANSACTIONS_TO_THRESHOLD
6

Business Meaning

This query analyzes customer purchase data to calculate average days and transactions needed for customers to reach a spending threshold of $250. Here's the breakdown of its business meaning:

1. purchase_data CTE:

- This section selects data from the "customers_amount" table, likely containing customer IDs, purchase dates, and purchase amounts.

- It calculates the cumulative amount spent by each customer over time using a window function.

2. Main Query:

- This part calculates the average days and transactions needed for customers to reach a spending threshold of $250.

- It identifies the first purchase date when the cumulative amount spent by a customer exceeds or equals $250.

- For each customer, it calculates the number of days and transactions required to reach the spending threshold.

- It then calculates the average days and transactions needed across all customers.

3. Business Implications:

- Understanding the average time and transactions required for customers to reach a spending threshold provides insights into customer purchasing behavior and preferences.

- Businesses can use this information to optimize marketing strategies, such as targeted promotions or loyalty programs, to encourage customers to reach the spending threshold faster.

- Knowing the average time and transactions needed can also help businesses forecast revenue and plan inventory management more effectively.

- Identifying customers who take longer to reach the spending threshold may prompt businesses to offer incentives or personalized recommendations to increase engagement and spending.

Overall, this analysis helps businesses understand the dynamics of customer spending behavior and enables them to tailor strategies to enhance customer engagement and drive sales.