
Wel- Come
Credit Card Sales Analysis

Project Overview

1. The analysis examines customer behavior across credit card categories, focusing on utilization ratios and transaction volumes.

2. It identifies customers with high utilization ratios and ranks the top customers by total transaction amounts each quarter.

3. The analysis tracks the contribution of annual fees by card categories and monitors transaction trends over time.

CREDIT_CARD Table:

- **CLIENT_NUM**: Unique identifier for each customer.
- **Card_Category**: Type of credit card.
- **Annual_Fees**: Yearly fee charged for the card.
- **Activation_30_Days**: Whether the card was activated within 30 days of issuance.
- **Customer_Acq_Cost**: Cost incurred for acquiring the customer.
- **Week_Start_Date**: Start date of the week for the data entry.
- **Week_Num**: Week number in the year.
- **Qtr**: The fiscal quarter in which the data was recorded.
- **current_year**: The current year for the record.
- **Credit_Limit**: Maximum amount of credit available on the card.
- **Total_Revolving_Bal**: Total balance carried forward month-to-month.
- **Total_Trans_Amt**: Total amount spent using the card in each period.
- **Total_Trans_Vol**: Total number of transactions made.
- **Avg_Utilization_Ratio**: Average percentage of credit limit used.
- **Use Chip:** Card usage type (online . Swipe , chip)
- **Exp Type**: Type of expenditure for transaction.
- **Interest_Earned**: Total interest earned by the credit card issuer.
- **Delinquent_Acc**: Number of delinquent (due or pending payment) accounts associated with the client.

CUSTOMER Table:

- **CLIENT_NUM**: Unique identifier for each customer.
- **CUSTOMER**: Name or identifier of the customer.
- **GENDER**: Gender of the customer.
- **DEPENDENT**: Number of dependents the customer has.
- **EDUCATION**: Education level of the customer .
- **MARITAL_STATUS**: Marital status of the customer .
- **STATE_CD**: The state code where the customer resides.
- **ZIPCODE**: Postal code of the customer’s address.
- **CAR_OWNER**: Whether the customer owns a car.
- **HOUSE_OWNER**: Whether the customer owns a house.
- **PERSONAL_LOAN**: Whether the customer has a personal loan .
- **CONTACT**: Contact details of the customer.
- **CUSTOMER_JOB**: Customer’s job or occupation.
- **INCOME**: Annual income of the customer.
- **CUSTOMER_SATISFACTION_SCORE**: Score representing the customer’s satisfaction level.

- Renaming Column Names in SQL

SQL Query

```
ALTER TABLE CREDIT_CARD
CHANGE COLUMN `ï»¿Client_Num`  CLIENT_NUM INT ;

ALTER TABLE CC_ADD
CHANGE COLUMN `ï»¿Client_Num`  CLIENT_NUM INT ;

ALTER TABLE CUST_ADD
CHANGE COLUMN `ï»¿Client_Num`  CLIENT_NUM INT ;
```

- **Q1:** Write a SQL query to retrieve the total number of customers, grouped by their GENDER and MARITAL STATUS.

SQL Query

```
SELECT GENDER , MARITAL_STATUS ,  COUNT(CLIENT_NUM) AS TOTAL_CLIENTS
 FROM CUSTOMER
 GROUP BY GENDER  , MARITAL_STATUS
 ORDER BY TOTAL_CLIENTS DESC;
```

Output

Result Grid | Filter Rows:

GENDER	MARITAL_STATUS	TOTAL_CLIENTS
F	Married	2953
F	Single	2516
M	Married	2175
M	Single	1720
F	Unknown	411
M	Unknown	333

Question 2 : Calculate the total Annual Fees and average Credit Limit for each Card Category

SQL Query

```
SELECT CARD_CATEGORY , SUM(ANNUAL_FEES) AS TOTAL_FEE ,
ROUND(AVG(CREDIT_LIMIT),2) AS AVERAGE_CREDIT_LIMIT
FROM CREDIT_CARD
GROUP BY CARD_CATEGORY;
```

Output

CARD_CATEGORY	TOTAL_FEE	AVERAGE_CREDIT_LIMIT
Blue	2685635	7285.66
Platinum	20665	16455.13
Silver	187505	23391.64
Gold	56210	21857.84

Question 3 : Find customers with an Avg Utilization Ratio greater than 0.5 and list their CLIENT NUM and Credit Limit.

SQL Query

```
SELECT CLIENT_NUM  , CREDIT_LIMIT , AVG_UTILIZATION_RATIO
FROM CREDIT_CARD
WHERE AVG_UTILIZATION_RATIO  >  0.5
ORDER BY AVG_UTILIZATION_RATIO DESC ;
```

Output

Result Grid | Filter Rows: | Export:

CLIENT_NUM	CREDIT_LIMIT	AVG_UTILIZATION_RATIO
709444983	2435	0.999
715847058	1863	0.995
778885158	2472	0.994
816476658	1812	0.992
789777408	1442	0.99
710382108	2452	0.988
720997383	1857	0.987
712292133	2330	0.985
778327908	2419	0.984
803205258	2420	0.983
789121233	2329	0.983
717592758	1606	0.983
709153008	2523	0.983
712190358	1843	0.978
741517458	2575	0.977
788370258	2578	0.976

4 Question: Find the top 5 customers with the highest Total Trans Amount in each quarter

SQL Query

```
SELECT * FROM
(SELECT CARD_CATEGORY , CLIENT_NUM , QTR ,  SUM(TOTAL_TRANS_AMT)AS TOTAL_TRANSACTION,
ROW_NUMBER() OVER (PARTITION BY QTR ORDER BY  SUM(TOTAL_TRANS_AMT) DESC ) AS RN
  FROM CREDIT_CARD
GROUP BY CLIENT_NUM , CARD_CATEGORY ,  QTR) RANKING
WHERE RN <= 5;
```

Output

Result Grid | Filter Rows: | Export: | Wrap Cell C

CARD_CATEGORY	CLIENT_NUM	QTR	TOTAL_TRANSACTION	RN
Platinum	712503408	Q1	17498	1
Platinum	708942408	Q1	16937	2
Platinum	710232483	Q1	16908	3
Platinum	709094358	Q1	16737	4
Platinum	711398433	Q1	16736	5
Platinum	717642633	Q2	17995	1
Platinum	716004258	Q2	17634	2
Gold	713758758	Q2	17628	3
Platinum	713965683	Q2	17258	4
Platinum	717783483	Q2	17038	5
Platinum	718140783	Q3	18484	1
Platinum	721220583	Q3	17390	2
Gold	756658083	Q3	17350	3

QUESTION 5: Identify customers whose CUSTOMER_SATISFACTION_SCORE is below 3 and whose income is below the average income of all customers

SQL Query

```
select CLIENT_NUM from customer
WHERE CUSTOMER_SATISFACTION_SCORE < 3 AND INCOME < (SELECT AVG(INCOME) FROM CUSTOMER) ;
```

Output

CLIENT_NUM
708083283
708084558
708085458
708098133
708099183
708104658
708112008
708117933
708119658
708129933
708134283
708145908
708147108

QUESTION 6 :Calculate the rolling average of Avg Utilization Ratio over the past 3 weeks for each customer

SQL Query

```
SELECT
    CLIENT_NUM,
    Week_Start_Date,
    AVG(Avg_Utilization_Ratio)
        OVER (PARTITION BY CLIENT_NUM ORDER BY Week_Start_Date ROWS BETWEEN 2 PRECEDING AND CURRENT ROW) AS Rolling_Avg_Utilization
FROM CREDIT_CARD;
```

Output

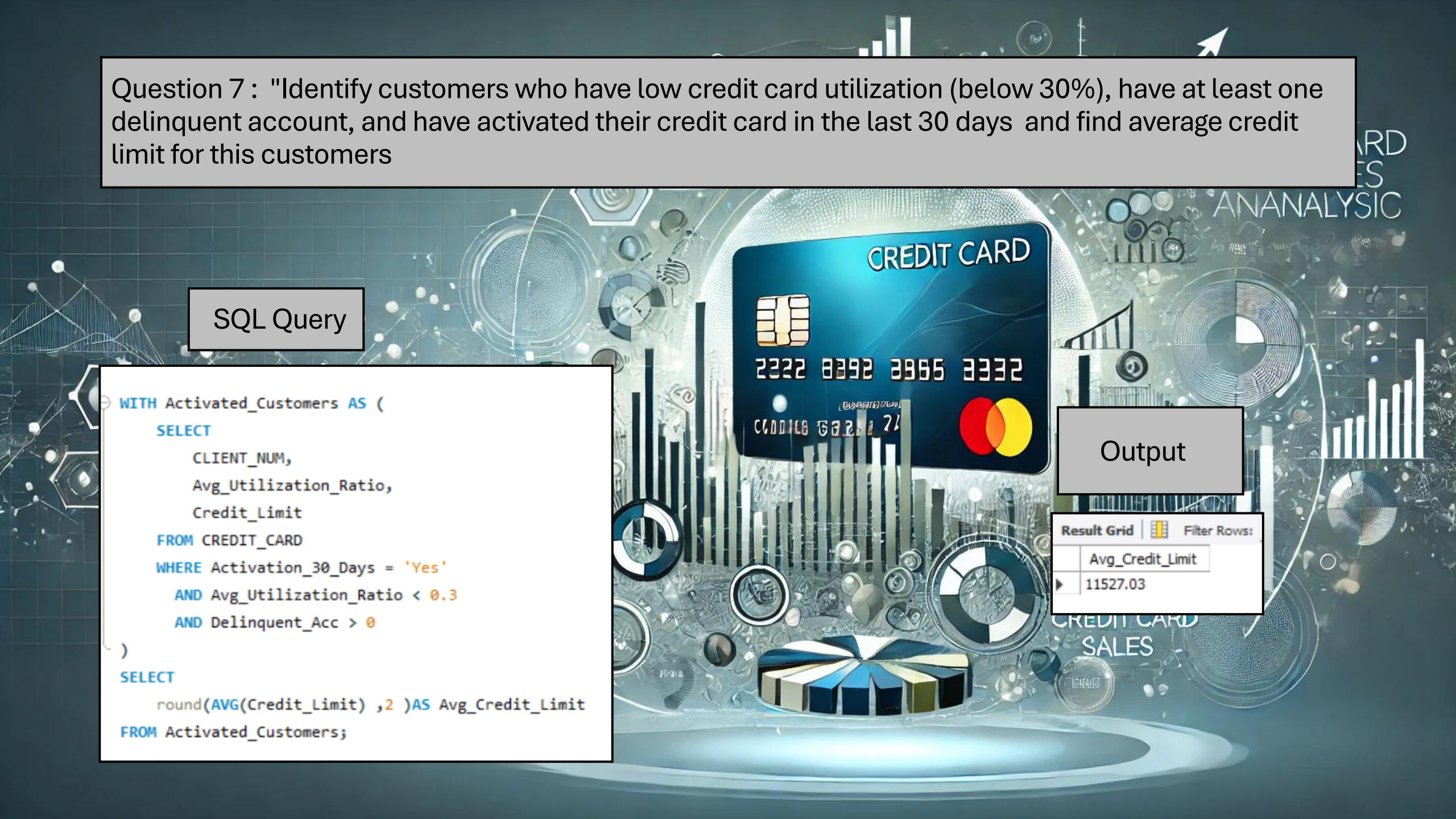
Result Grid | Filter Rows: | Export:

CLIENT_NUM	Week_Start_Date	Rolling_Avg_Utilization
708082083	01-01-2023	0.469
708083283	01-01-2023	0.736
708084558	01-01-2023	0.214
708085458	01-01-2023	0
708086958	01-01-2023	0.239
708095133	01-01-2023	0.055
708098133	01-01-2023	0.5
708099183	01-01-2023	0.327
708100533	01-01-2023	0.85
708103608	01-01-2023	0.212
708104658	01-01-2023	0.619

Question 7 : "Identify customers who have low credit card utilization (below 30%), have at least one delinquent account, and have activated their credit card in the last 30 days and find average credit limit for this customers

SQL Query

```sql
WITH Activated_Customers AS (
    SELECT
        CLIENT_NUM,
        Avg_Utilization_Ratio,
        Credit_Limit
    FROM CREDIT_CARD
    WHERE Activation_30_Days = 'Yes'
      AND Avg_Utilization_Ratio < 0.3
      AND Delinquent_Acc > 0
)
SELECT
    round(AVG(Credit_Limit) ,2 )AS Avg_Credit_Limit
FROM Activated_Customers;
```



Output

Result Grid | Filter Rows:

Avg_Credit_Limit
11527.03

Question 8 : Identify customers with significant transaction amounts who also have delinquent accounts.

SQL Query

```sql
WITH Ranked_Customers AS (
    SELECT
        CLIENT_NUM,
        Total_Trans_Amt,
        PERCENT_RANK() OVER (ORDER BY Total_Trans_Amt DESC) AS Percent_Rank_
    FROM CREDIT_CARD
)
SELECT
    CLIENT_NUM,
    Total_Trans_Amt
FROM Ranked_Customers
WHERE Percent_Rank_ >= 0.9 AND CLIENT_NUM IN (
    SELECT CLIENT_NUM FROM CREDIT_CARD WHERE Delinquent_Acc > 0
);
```

Output

Result Grid | Filter Rows:

CLIENT_NUM	Total_Trans_Amt
708162558	1355
708288108	647
708295533	1215
708296883	1178
708300483	704
708434733	1249
708892833	725
708895158	1371
709701558	1460
709707033	1424
710573808	1334
710574558	1403
711204483	781
711318033	1304
711711483	1433
711927108	1196

Question 9 : calculate the total interest earned by customers whose acquisition cost is above the median value

SQL Query

```sql
WITH RANKING AS (
    SELECT
        CLIENT_NUM,
        CUSTOMER_ACQ_COST,
        DENSE_RANK() OVER (ORDER BY CUSTOMER_ACQ_COST DESC, CLIENT_NUM DESC) AS DESC_,
        DENSE_RANK() OVER (ORDER BY CUSTOMER_ACQ_COST ASC, CLIENT_NUM ASC) AS ASC_
    FROM CREDIT_CARD
),
MedianCost AS (
    SELECT
        CAST(AVG(CUSTOMER_ACQ_COST) AS SIGNED) AS MEDIAN
    FROM RANKING
    WHERE DESC_ = ASC_ OR ASC_ + 1 = DESC_ OR ASC_ - 1 = ASC_
)
SELECT
    CLIENT_NUM,
    SUM(Interest_Earned) AS TOTAL_INTEREST
FROM CREDIT_CARD
WHERE CUSTOMER_ACQ_COST > (SELECT MEDIAN FROM MedianCost)
GROUP BY CLIENT_NUM
```

Output

CLIENT_NUM	TOTAL_INTEREST
708083283	69.44
708084558	202.58
708085458	236.4
708086958	1004.87
708104658	732
708113208	525
708128733	1303.64
708134283	311.43
708139833	441.37
708147108	277.08
708152358	721.71
708153558	634.88
708158133	350.08
708160008	1927.12
708161133	466.92
708163758	2355.3

Question 10 : Identify the percentage of delinquent accounts in each quarter

SQL Query

```sql
WITH Delinquent_Accounts AS (
    SELECT
        Qtr,
        COUNT(Delinquent_Acc) AS Total_Delinquent_Acc
    FROM CREDIT_CARD
    WHERE Delinquent_Acc > 0
    GROUP BY Qtr
),
Total_Accounts AS (
    SELECT
        Qtr,
        COUNT(*) AS Total_Acc
    FROM CREDIT_CARD
    GROUP BY Qtr
)
SELECT
    D.Qtr,
    D.Total_Delinquent_Acc,
    ROUND((D.Total_Delinquent_Acc * 100.0 / T.Total_Acc), 2)  AS Delinquent_Percentage
FROM Delinquent_Accounts D
JOIN Total_Accounts T ON D.Qtr = T.Qtr;
```

Output

Result Grid | Filter Rows: | Export:

Qtr	Total_Delinquent_Acc	Delinquent_Percentage
Q1	181	7.14
Q2	136	5.36
Q3	157	6.19
Q4	140	5.59

Question 11 : For each Card Category, calculate the delinquency rate (number of delinquent accounts divided by total accounts).

SQL Query

```
SELECT CARD_CATEGORY  ,
COUNT(DELINQUENT_ACC)  * 100 / (SELECT COUNT(*) FROM CREDIT_CARD)  AS DELIQUENT_PEERCENTAGE
FROM CREDIT_CARD
WHERE DELINQUENT_ACC > 0
GROUP BY CARD_CATEGORY ;
```

Output

Result Grid | Filter Rows:

CARD_CATEGORY	DELIQUENT_PEERCENTAGE
Blue	5.5303
Silver	0.3858
Gold	0.1187
Platinum	0.0396

Question 12 : Identify customers who have a high credit limit but low transaction volume, and calculate the total interest earned for this segment.

SQL Query

```sql
WITH High_Limit_Low_Volume AS (
    SELECT
        CLIENT_NUM,
        CREDIT_LIMIT,
        SUM(Total_Trans_Vol) AS Total_Trans_Vol,
        SUM(Interest_Earned) AS Total_Interest_Earned
    FROM CREDIT_CARD
    GROUP BY CLIENT_NUM, CREDIT_LIMIT
),
Overall_Avg_Volume AS (
    SELECT
        AVG(Total_Trans_Vol) AS Overall_Avg_Trans_Vol,
        AVG(CREDIT_LIMIT) AS AVG_LIMIT
    FROM CREDIT_CARD
)
SELECT
    H.CLIENT_NUM,
    H.CREDIT_LIMIT,
    H.Total_Trans_Vol,
    H.Total_Interest_Earned,
    CASE
        WHEN H.Total_Trans_Vol < O.Overall_Avg_Trans_Vol AND H.CREDIT_LIMIT > O.AVG_LIMIT
        THEN 'Target for Increased Spending'
        ELSE 'No Action Needed'
    END AS Marketing_Strategy
FROM High_Limit_Low_Volume H
CROSS JOIN Overall_Avg_Volume O
WHERE H.Total_Trans_Vol < O.Overall_Avg_Trans_Vol AND H.CREDIT_LIMIT > O.AVG_LIMIT ;
```

Output

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

CLIENT_NUM	CREDIT_LIMIT	Total_Trans_Vol	Total_Interest_Earned	Marketing_Strategy
708095133	33304	29	275.12	Target for Increased Spending
708112008	23510	35	465.08	Target for Increased Spending
708119658	12836	53	554.18	Target for Increased Spending
708121908	22917	45	265.85	Target for Increased Spending
708152358	18512	57	721.71	Target for Increased Spending
708155733	14734	23	296.47	Target for Increased Spending
708177333	11261	26	243.18	Target for Increased Spending
708193008	14480	43	387.84	Target for Increased Spending
708195633	34516	54	488	Target for Increased Spending
708197358	30543	63	622.72	Target for Increased Spending
708217758	14035	30	385.44	Target for Increased Spending
708223383	11410	38	793.44	Target for Increased Spending
708243483	14938	51	427.12	Target for Increased Spending
708292833	27389	60	247.86	Target for Increased Spending
708296883	11749	27	259.16	Target for Increased Spending
708316008	10031	29	394.24	Target for Increased Spending