CREDIT CARD ANALYSIS

The Queries Are Divided Into Three Parts

1. Beginner- Page (3-12)
2. Intermediate- Page (13-26)
3. Advanced- Page (27-42)

These are some of the Questions Answered Using MYSQL

1. Total Customers
2. Average Revenue Per Customer
3. Gender wise Monthly Revenue
4. Running Total by Weeks
5. Revenue Moving Averages for 4 Weeks
6. Top 5 Most Positive and Negative Weeks by Revenue
7. Monthly Revenue and Month on Month Change
8. Three Month Simple Moving Average
9. Quarter on Quarter Change %
10. Compare Average Transaction Amounts

BEGINNER QUERIES

Total Customers

```
SELECT
    COUNT(Client_Num) AS 'Total_Customers'
FROM
    creditcard_detail;
```

Total_Customers
10108

Revenue

```
SELECT
    CONCAT('$ ',
            ROUND(SUM(Annual_Fees) + SUM(Interest_Earned),
                    2)) AS 'Revenue'
FROM
    creditcard_detail;
```

Revenue
$ 10793397.23

Above Average Acquisition Cost by Card Category

```sql
SELECT
    Card_Category 'Card Category',
    AVG(Customer_Acq_Cost) AS 'AVG ACQ'
FROM
    creditcard_detail
GROUP BY 1
HAVING AVG(Customer_Acq_Cost) > (SELECT
        AVG(Customer_Acq_Cost)
    FROM
        creditcard_detail);
```

Card Category	AVG ACQ
Blue	96.3722
Platinum	98.0597

Average Revenue Per Customer

```
SELECT
    ROUND(SUM(Annual_Fees + Interest_Earned) / COUNT(*),
          2) AS 'AVG Customer Revenue'
FROM
    creditcard_detail;
```

AVG Customer Revenue
1067.81

Customer Acquisition Cost (Minimum, Average,Maximum)

Min Customer_ACQ_Cost	AVG Customer_ACQ_Cost	Max Customer_ACQ_Cost
40	96.25	172

```
SELECT
    MIN(Customer_Acq_Cost) AS 'Min Customer_ACQ_Cost',
    ROUND(AVG(Customer_Acq_Cost), 2) AS 'AVG Customer_ACQ_Cost',
    MAX(Customer_Acq_Cost) AS 'Max Customer_ACQ_Cost'
from creditcard_detail;
```

Averages by Card Category

```sql
SELECT
    Card_Category as 'Card Category',
    COUNT(Client_Num) AS 'Total Customers',
    round(AVG(Customer_Acq_Cost), 2) AS 'AVG ACQ',
    round(AVG(Interest_Earned), 2) AS 'AVG Interest',
    round(AVG(Credit_Limit), 2) AS 'AVG Limit',
    round(AVG(Annual_Fees), 2) AS 'AVG Fees',
    round(AVG(Avg_Utilization_Ratio), 2) AS 'AVG Uitilization Ratio'
FROM
    creditcard_detail
GROUP BY 1;
```

Card Category	Total Customers	AVG ACQ	AVG Interest	AVG Limit	AVG Fees	AVG Uitilization Ratio
Blue	9214	96.37	705	7285.66	291.47	0.29
Platinum	67	98.06	2412.37	16455.13	308.43	0.15
Silver	639	95.23	1270.86	23391.64	293.44	0.08
Gold	188	93.31	1988.21	21857.84	298.99	0.12

Total customers by Card Category

```sql
SELECT
    Card_Category, COUNT(Card_Category) AS 'TotalCustomers'
FROM
    creditcard_detail
GROUP BY 1
ORDER BY 2 DESC;
```

Card_Category	TotalCustomers
Blue	9214
Silver	639
Gold	188
Platinum	67

AVG Age by Card Category

Card_Category	AVG_Age
Blue	46
Platinum	47
Silver	47
Gold	48

```sql
SELECT
    Card_Category AS 'Card_Category',
    ROUND(AVG(Customer_Age)) AS 'AVG_Age'
FROM
    creditcard_detail AS ccd
        JOIN
    customers_details AS cud ON ccd.Client_Num = cud.Client_Num
GROUP BY 1;
```

Male & Female Count by Card Category

```sql
SELECT
    Card_Category,
    COUNT(CASE WHEN Gender = 'M' THEN 0 END) AS 'Male_Customers',
    COUNT(CASE WHEN Gender = 'F' THEN 1 END) AS 'Female_Customers'
FROM
    creditcard_detail AS ccd
        JOIN
    customers_details AS cud ON ccd.Client_Num = cud.Client_Num
GROUP BY
    Card_Category;
```

Card_Category	Male_Customers	Female_Customers
Blue	3829	5385
Platinum	33	34
Silver	274	365
Gold	92	96

Male & Female Count

Gender	Count
F	5880
M	4228

```sql
SELECT
    Gender,
    COUNT(Gender) as 'Count'
FROM
    customers_details
GROUP BY 1;
```

Average Satisfaction Score by Card Category

```sql
SELECT
    Card_Category AS 'card_category',
    AVG(Cust_Satisfaction_Score) AS 'AVG satisfaction_score'
FROM
    customers_details AS cud
        JOIN
    creditcard_detail AS ccd ON cud.Client_Num = ccd.Client_Num
GROUP BY 1
ORDER BY 2 DESC;
```

card_category	AVG satisfaction_score
Silver	3.2175
Blue	3.1937
Gold	3.0426
Platinum	2.7164

Average Transaction Amount

avg transaction amount
4405

```sql
SELECT
    ROUND(AVG(Total_Trans_Amt)) AS 'avg transaction amount'
FROM
    creditcard_detail;
```

Average Transaction Amount by Card Category

```sql
SELECT
    Card_Category,
    ROUND(AVG(Total_Trans_Amt)) AS 'avg transaction amount'
FROM
    creditcard_detail
GROUP BY 1
ORDER BY 2 DESC;
```

Card_Category	avg transaction amount
Platinum	14229
Gold	10766
Silver	7178
Blue	4011

Card Uses by Expense Type

Exp Type	Card_Use
Bills	2970
Entertainment	1988
Fuel	1759
Grocery	1502
Food	1187
Travel	702

```sql
SELECT
    `Exp Type`, COUNT(Client_Num) AS 'Card_Use'
FROM
    creditcard_detail
GROUP BY 1
ORDER BY 2 DESC;
```

Average Income

```
SELECT
    round(avg(Income), 2) as 'AVG Income'
FROM
    customers_details;
```

AVG Income
56976.10

Income by (Minimum, Average, Maximum)

Min Income	AVG Income	Max Income
1250	56976.10	239791

```
SELECT
    MIN(Income) AS 'Min Income',
    ROUND(AVG(Income), 2) AS 'AVG Income',
    MAX(Income) AS 'Max Income'
FROM
    customers_details;
```

INTERMEDIATE QUERIES

Total Delinquent Accounts and Rate

```sql
SELECT
    COUNT(CASE
        WHEN Delinquent_Acc = 1 THEN 1
    END) AS 'Total Delinquents',
    ROUND(COUNT(CASE
                WHEN Delinquent_Acc = 1 THEN 1
            END) / COUNT(*) * 100,
            2) AS 'Delinquency Rate'
FROM
    creditcard_detail;
```

Total Delinquents	Delinquency Rate
614	6.07

Total Delinquent Accounts by Card Category

Card_Category	Delinquent_Accounts
Blue	559
Silver	39
Gold	12
Platinum	4

```sql
SELECT
    Card_Category,
    COUNT(Delinquent_Acc) AS 'Delinquent_Accounts'
FROM
    creditcard_detail
WHERE
    Delinquent_Acc = 1
GROUP BY 1;
```

Gender wise Delinquent Count and Rate

```sql
SELECT
    gender AS 'Gender',
    COUNT(case when Delinquent_Acc = 1 then 1 end) AS 'Delinquent Count',
    count(case when Delinquent_Acc = 1 then 1 end) / count(*) * 100 as 'Delinquency Rate'
FROM
    customers_details AS cud
        JOIN
    creditcard_detail AS ccd ON cud.Client_Num = ccd.Client_Num
GROUP BY 1;
```

Gender	Delinquent Count	Delinquency Rate
F	356	6.0544
M	258	6.1022

Male & Female Count

Gender	Count
F	5880
M	4228

```sql
SELECT
    Gender,
    COUNT(Gender) as 'Count'
FROM
    customers_details
GROUP BY 1;
```

Delinquent Count and Rate by customer jobs

```sql
SELECT
    cud.Customer_Job AS 'Customer Job',
    count(*) as 'Total Customers',
    COUNT(case when Delinquent_Acc = 1 then 1 end) AS 'Delinquent Count',
    round(count(case when Delinquent_Acc = 1 then 1 end) / count(*) * 100, 2) as 'Delinquent Rate'
FROM
    customers_details AS cud
        JOIN
    creditcard_detail AS ccd ON cud.Client_Num = ccd.Client_Num
GROUP BY 1
order by 2 desc;
```

Customer Job	Total Customers	Delinquent Count	Delinquent Rate
Selfemployeed	2575	167	6.49
Businessman	1901	101	5.31
Blue-collar	1579	87	5.51
White-collar	1542	85	5.51
Govt	1525	113	7.41
Retirees	986	61	6.19

Customer count by Income Group

```sql
SELECT
    CASE
        WHEN income BETWEEN 1250 AND 25000 THEN 'Low Income'
        WHEN income BETWEEN 25001 AND 50000 THEN 'Mid Income'
        WHEN income BETWEEN 50001 AND 75000 THEN 'High Income'
        ELSE 'Very High'
    END AS 'Income_Group',
    COUNT(Income) AS 'Customer Count'
FROM
    creditcard_detail AS ccd
        JOIN
    customers_details AS cud ON ccd.Client_Num = cud.Client_Num
GROUP BY 1
ORDER BY 2 DESC;
```

Income_Group	Customer Count
Low Income	2804
Mid Income	2719
Very High	2613
High Income	1972

AVG Transaction Amt by Card Category

```sql
SELECT
    Card_Category AS 'card category',
    COUNT(Card_Category) AS 'total customers',
    round(AVG(Annual_Fees), 2) AS 'AVG fees'
FROM
    creditcard_detail
GROUP BY Card_Category
ORDER BY 3 DESC;
```

card category	total customers	AVG fees
Platinum	67	308.43
Gold	188	298.99
Silver	639	293.44
Blue	9214	291.47

Above Average Acquisition Cost by Card Category using Sub-Query

```
SELECT
    Card_Category AS 'Card Category',
    AVG(Customer_Acq_Cost) AS 'AVG ACQ',
    (SELECT
            AVG(Customer_Acq_Cost)
        FROM
            creditcard_detail) AS 'Overall AVG ACQ'
FROM
    creditcard_detail
GROUP BY Card_Category
HAVING AVG(Customer_Acq_Cost) > (SELECT
        AVG(Customer_Acq_Cost)
    FROM
        creditcard_detail);
```

Card Category	AVG ACQ	Overall AVG ACQ
Blue	96.3722	96.2541
Platinum	98.0597	96.2541

```sql
SELECT
    Gender,
    COUNT(CASE
        WHEN Gender = 'M' THEN 1
        ELSE 0
    END) AS 'total_customers',
    ROUND(AVG(CASE
                WHEN Gender = 'M' THEN Annual_Fees
                ELSE Annual_Fees
            END)) AS 'AVG_annual_fees',
    ROUND(AVG(CASE
                WHEN Gender = 'M' THEN Interest_Earned
                ELSE Interest_Earned
            END)) AS 'AVG_interest_earned',
    ROUND(AVG(CASE
                WHEN Gender = 'M' THEN Customer_Acq_Cost
                ELSE Customer_Acq_Cost
            END)) AS 'AVG_ACQ_cost',
    CAST(AVG(CASE
            WHEN Gender = 'M' THEN Avg_Utilization_Ratio
            ELSE Avg_Utilization_Ratio
        END)
        AS DECIMAL (3 , 2 )) 'AVG_utili_ratio'
FROM
    creditcard_detail AS ccd
        JOIN
    customers_details AS cud ON ccd.Client_Num = cud.Client_Num GROUP BY 1;
```

Gender wise Total Customers & AVG (Annual Fees, Interest Earned, Acquisition Cost, Utilization ratio)

Gender	total_custc	AVG_annual_fee	AVG_interest_ear	AVG_ACQ_cost	AVG_utili_ratio
F	5880	293	615	95	0.26
M	4228	290	1000	99	0.30

Profitability Analysis
(e.g., Profit Margin)

```sql
SELECT
    CONCAT('Q',
           QUARTER(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y'))) AS Quarter,
    YEAR(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) AS Year,
    ROUND(SUM(Annual_Fees + Interest_Earned), 2) AS Revenue,
    ROUND(SUM(Customer_Acq_Cost), 2) AS Total_Cost,
    ROUND(SUM(Annual_Fees + Interest_Earned) - SUM(Customer_Acq_Cost),
           2) AS Profit,
    ROUND(((SUM(Annual_Fees + Interest_Earned) - SUM(Customer_Acq_Cost)) / SUM(Annual_Fees + Interest_Earned)) * 100,
           2) AS Profit_Margin_Percentage
FROM
    creditcard_detail
GROUP BY Year , Quarter
ORDER BY Year , Quarter;
```

Quarter	Year	Revenue	Total_Cost	Profit	Profit_Margin_Percentage
Q1	2023	2713774.77	242346	2471428.77	91.07
Q2	2023	2685265.72	243889	2441376.72	90.92
Q3	2023	2785779.69	244555	2541224.69	91.22
Q4	2023	2608577.05	242146	2366431.05	90.72

Average Utilization Ratio and Average Credit Limit by Age Group

```sql
SELECT
    CASE
        WHEN Customer_Age BETWEEN 18 AND 25 THEN '18-25'
        WHEN Customer_Age BETWEEN 26 AND 35 THEN '26-35'
        WHEN Customer_Age BETWEEN 36 AND 45 THEN '36-45'
        WHEN Customer_Age BETWEEN 46 AND 60 THEN '46-60'
        WHEN Customer_Age > 60 THEN '60+'
        ELSE 'Unknown'
    END AS Age_Group,
    round(AVG(Avg_Utilization_Ratio), 3) AS avg_utilization_ratio,
    round(AVG(Credit_Limit), 2) AS avg_credit_limit
FROM creditcard_detail ccd
JOIN customers_details cud ON ccd.Client_Num = cud.Client_Num
GROUP BY Age_Group
ORDER BY Age_Group;
```

Age_Group	avg_utilization_ratio	avg_credit_limit
18-25	0.268	10117.45
26-35	0.265	8537.32
36-45	0.279	8617.1
46-60	0.273	8697.61
60+	0.279	8186.71

Card Category wise Monthly Revenue

```sql
SELECT
    Card_Category,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'January' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS January,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'February' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS February,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'March' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS March,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'April' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS April,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'May' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS May,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'June' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS June,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'July' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS July,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'August' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS August,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'September' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS September,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'October' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS October,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'November' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS November,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'December' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS December
FROM
    creditcard_detail
GROUP BY
    Card_Category;
```

Card_Category	January	February	March	April	May	June	July	August	September	October	November	December
Blue	859894.22	727340.42	697291.49	846287.7	704968.23	734614.42	919862.16	742161.42	713559.72	856591.15	707073.22	671878.59
Platinum	30321.62	21294.35	8977.94	18751.23	15735.75	26670.47	13150.82	8250.89	11448.29	16589.08	9319.15	1784.46
Silver	108704.55	70760.68	67739.71	101528.28	65655.95	71053.37	120440.39	64174.76	74432.63	93556.33	89878.52	71661.11
Gold	52358.78	27728.22	41362.79	43824.16	33793.46	22382.7	54101.42	28713.92	35483.27	37341.72	18047.87	34855.85

Top 3 States by Most Customers In Each Card Category

```sql
with cte as(
SELECT
    cd.Card_Category AS 'category',
    COUNT(cd.Client_Num) AS 'Total_Customers',
    cust.state_cd AS 'state'
FROM
    creditcard_detail AS cd
        JOIN
    customers_details AS cust ON cd.client_Num = cust.Client_Num
GROUP BY 3 , 1
ORDER BY 2 DESC),
cte1 as(
select category,
Total_Customers,
state,
rank() over(partition by category order by `Total_Customers` desc) as 'rnk' from cte)
SELECT
    *
FROM
    cte1
WHERE
    rnk IN (1 , 2, 3);
```

category	Total_Customers	state	rnk
Blue	2266	CA	1
Blue	2190	TX	2
Blue	2071	NY	3
Gold	48	NY	1
Gold	43	TX	2
Gold	43	CA	2
Platinum	18	CA	1
Platinum	14	FL	2
Platinum	11	TX	3
Platinum	11	NY	3
Silver	150	TX	1
Silver	141	CA	2
Silver	140	NY	3

Top 10 Client by Total Transaction Amount

```sql
with cte as (
SELECT
    Client_Num, Card_Category, Total_Trans_Amt
FROM
    creditcard_detail
ORDER BY 3 DESC),
cte1 as (
select *, rank() over(order by Total_Trans_Amt desc) as rnk from cte)
SELECT
    *
FROM
    cte1
WHERE
    rnk BETWEEN 1 AND 10;
```

Client_Num	Card_Category	Total_Trans_Amt	rnk
718140783	Platinum	18484	1
717642633	Platinum	17995	2
801036033	Platinum	17744	3
716004258	Platinum	17634	4
713758758	Gold	17628	5
712503408	Platinum	17498	6
778428108	Platinum	17437	7
721220583	Platinum	17390	8
756658083	Gold	17350	9
713965683	Platinum	17258	10

Gender wise Monthly Revenue using view

```sql
create view Monthly_Revenue as
SELECT
    Card_Category,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'January' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS January,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'February' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS February,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'March' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS March,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'April' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS April,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'May' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS May,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'June' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS June,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'July' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS July,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'August' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS August,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'September' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS September,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'October' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS October,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'November' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS November,
    ROUND(SUM(CASE WHEN MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) = 'December' THEN Annual_Fees + Interest_Earned ELSE 0 END), 2) AS December
FROM
    creditcard_detail
GROUP BY
    Card_Category;
-- use view
select * from monthly_revenue;
```

Gender	January	February	March	April	May	June	July	August	September	October	November	December
F	515689.38	401711.71	456960.06	496490.33	371618.29	411388.56	554331.31	437428.75	394402.75	502687.67	414652.63	379700.75
M	535589.79	445411.96	358411.87	513901.04	448535.1	443332.4	553223.48	405872.24	440521.16	501390.61	409666.13	400479.26

```sql
SELECT
    `Exp Type`,
    ROUND(AVG(Total_Trans_Amt), 2) AS 'AVG_Amt_Exp_Type',
    (SELECT
            ROUND(AVG(Total_Trans_Amt), 2)
        FROM
            creditcard_detail
        WHERE
            Card_Category LIKE 'blue%'
                AND `use chip` LIKE 'online%') AS 'AVG_Trans_Amt_Condition',
    (SELECT
            ROUND(AVG(Total_Trans_Amt), 2)
        FROM
            creditcard_detail) AS 'AVG_Trans_Amt_Overall'
FROM
    creditcard_detail
WHERE
    `Use Chip` LIKE 'Online%'
        AND Card_Category LIKE 'blue%'
GROUP BY 1
HAVING AVG(Total_Trans_Amt) > AVG_Trans_Amt_Condition
ORDER BY 2 DESC;
```

Compare AVG Transaction AMT for 'Blue' Card Category with Chip for 'Online' Transactions Using Sub-Query

Exp Type	AVG Amt Exp Type	AVG Trans Amt Condition	AVG Trans Amt Overall
Travel	7650.09	4222.51	4404.63
Food	5334.60	4222.51	4404.63

ADVANCED QUERIES

```sql
with cte1 as (
SELECT
    `Exp Type` as 'Exp Type', round(AVG(Total_Trans_Amt), 2) AS 'AVG Amt Exp Type'
FROM
    creditcard_detail
WHERE
    `Use Chip` LIKE 'online%'
        AND Card_Category LIKE 'blue%'
GROUP BY 1),
cte2 as (
select round(AVG(Total_Trans_Amt), 2) AS 'AVG Trans Amt Condition'
FROM
    creditcard_detail
WHERE
    `Use Chip` LIKE 'online%'
        AND Card_Category LIKE 'blue%'),
cte3 as (
SELECT
    round(AVG(Total_Trans_Amt), 2) AS 'AVG Trans Amt Overall'
FROM
    creditcard_detail)
SELECT
    `Exp Type`, `AVG Amt Exp Type`, `AVG Trans Amt Condition`, `AVG Trans Amt Overall`
FROM
    cte1
        JOIN
    cte2
        JOIN
    cte3
HAVING `AVG Amt Exp Type` > `AVG Trans Amt Condition`
ORDER BY 2 DESC;
```

Compare AVG Transaction AMT for 'Blue' Card Category with Chip for 'Online' Transactions, using CTEs

Exp Type	AVG Amt Exp Type	AVG Trans Amt Condition	AVG Trans Amt Overall
Travel	7650.09	4222.51	4404.63
Food	5334.60	4222.51	4404.63

AVG Annual Fees by State Compared to AVG Annual Fees

```sql
with cte as (
SELECT
    state_cd AS 'State',
    COUNT(*) AS 'Total Customers',
    ROUND(AVG(Annual_Fees)) AS 'AVG Fees State',
    ROUND((SELECT
                AVG(Annual_Fees)
            FROM
                creditcard_detail)) AS 'AVG Annual Fees'
FROM
    creditcard_detail AS ccd
        JOIN
    customers_details AS cud ON ccd.Client_Num = cud.Client_Num
GROUP BY 1)
SELECT
    *, concat('$ ', (`AVG Fees State` - `AVG Annual Fees`)) AS 'Diff'
FROM
    cte;
```

State	Total Customers	AVG Fees State	AVG Annual Fees	Diff
FL	1711	292	292	$ 0
NJ	716	294	292	$ 2
NY	2270	290	292	$ -2
TX	2394	293	292	$ 1
CA	2468	291	292	$ -1
MO	20	267	292	$ -25
MA	28	288	292	$ -4
IA	47	298	292	$ 6
AK	31	296	292	$ 4
MI	63	294	292	$ 2
GA	16	283	292	$ -9
CT	14	284	292	$ -8
IL	58	298	292	$ 6
VA	27	266	292	$ -26
UT	27	305	292	$ 13
HI	9	301	292	$ 9
AZ	10	329	292	$ 37
WA	30	272	292	$ -20
NV	56	303	292	$ 11
CO	9	315	292	$ 23
MN	22	315	292	$ 23
AR	11	322	292	$ 30
PA	18	328	292	$ 36
OR	5	290	292	$ -2
OH	11	331	292	$ 39
NM	12	303	292	$ 11
SC	17	339	292	$ 47
NE	8	386	292	$ 94

Credit Card Activated before 30 days by Card Category

```sql
with cte as (
SELECT
    Card_Category,
    COUNT(Client_Num) as 'total_customers',
    COUNT(CASE
        WHEN Activation_30_Days = 1 THEN 1
    END) AS 'activate_30_days'
FROM
    creditcard_detail
GROUP BY 1)
SELECT
    Card_Category,
    total_customers,
    activate_30_days,
    CONCAT(ROUND((activate_30_days / total_customers) * 100),
            ' %') AS 'percent'
FROM
    cte;
```

Card_Category	total_custc	activate_30_days	percent
Blue	9214	5270	57 %
Platinum	67	39	58 %
Silver	639	394	62 %
Gold	188	106	56 %

Week on Week Change

```sql
with cte as (
SELECT
    WEEK(STR_TO_DATE(week_start_date, '%d-%m-%Y')) AS 'Week_No',
    ROUND(SUM(Annual_Fees) + SUM(Interest_Earned)) AS 'Revenue'
FROM
    creditcard_detail AS ccd
        JOIN
    customers_details AS cud ON ccd.Client_Num = cud.Client_Num
GROUP BY 1
ORDER BY 1),
cte1 as (
select *, coalesce(lag(Revenue) over(), 0) as 'Previous_Week' from cte)
SELECT
    week_no,
    revenue,
    COALESCE(CONCAT(ROUND(((revenue - previous_week) / previous_week) * 100),
                    '%'),
            0) AS 'WoW_change'
FROM
    cte1;
```

week_no	revenue	WoW_change
1	199862	0
2	208350	4%
3	224883	8%
4	202684	-10%
5	215500	6%
6	222878	3%
7	209153	-6%
8	203451	-3%
9	211641	4%
10	194740	-8%
11	215556	11%
12	216452	0%
13	188624	-13%
14	194431	3%
15	205506	6%
16	215236	5%
17	193514	-10%
18	201705	4%
19	210374	4%
20	198706	-6%
21	216612	9%

Running Total by Week

```sql
with cte as (
SELECT
    WEEK(STR_TO_DATE(week_start_date, '%d-%m-%Y')) AS 'Week_No',
    ROUND(SUM(Annual_Fees) + SUM(Interest_Earned)) AS 'Revenue'
FROM
    creditcard_detail AS ccd
        JOIN
    customers_details AS cud ON ccd.Client_Num = cud.Client_Num
GROUP BY 1
ORDER BY 1)
SELECT
    week_no,
    sum(revenue) over(order by week_no) AS 'Cumulative_Revenue'
FROM
    cte group by 1;
```

week_no	Cumulative_Revenue
1	199862
2	408212
3	633095
4	835779
5	1051279
6	1274157
7	1483310
8	1686761
9	1898402
10	2093142
11	2308698
12	2525150
13	2713774
14	2908205
15	3113711
16	3328947
17	3522461
18	3724166
19	3934540
20	4133246
21	4349858

Revenue Moving Averages for 4 Weeks

```sql
with cte as (
SELECT
    WEEK(STR_TO_DATE(week_start_date, '%d-%m-%Y')) AS 'Week_No',
    ROUND(SUM(Annual_Fees) + SUM(Interest_Earned)) AS 'Revenue'
FROM
    creditcard_detail AS ccd
        JOIN
    customers_details AS cud ON ccd.Client_Num = cud.Client_Num
GROUP BY 1
ORDER BY 1)
select week_no,
round(avg(revenue)
over(order by week_no rows between 3 preceding and current row))
as Moving_AVG
from
    cte;
```

week_no	Moving_AVG
1	199862
2	204106
3	211032
4	208945
5	212854
6	216486
7	212554
8	212746
9	211781
10	204746
11	206347
12	209597
13	203843
14	203766
15	201253
16	200949
17	202172
18	203990
19	205207
20	201075
21	206849

Creating and Using Temp Table

```sql
-- create Age Group temporary table
create temporary table Temp_Age_Group (
SELECT
    *,
    CASE
        WHEN Customer_Age BETWEEN 21 AND 35 THEN '21-35'
        WHEN Customer_Age BETWEEN 36 AND 49 THEN '36-49'
        WHEN Customer_Age BETWEEN 50 AND 60 THEN '50-60'
        ELSE '60+'
    END AS 'age_group'
FROM
    customers_details
);

-- create temporary table for Revenue
create temporary table temp_revenue (
SELECT
    *, ROUND(Annual_Fees + interest_earned, 2) AS 'Revenue'
FROM
    creditcard_detail );
```

```sql
-- Revenue by Age Group
SELECT
    age_group,
    COUNT(revenue) AS 'total_customers',
    ROUND(SUM(revenue), 2) AS 'Revenue'
FROM
    temp_revenue AS tr
        JOIN
    temp_age_group AS tag ON tr.Client_Num = tag.client_num
GROUP BY 1
ORDER BY 2 DESC;
```

age_group	total_custc	Revenue
36-49	5645	5943114.17
50-60	3120	3586981.5
21-35	938	921013.85
60+	405	342287.71

Delinquent Count by Age Group Using Temp Table

```sql
with cte as(
SELECT
    tage.age_group AS 'age_group',
    COUNT(trev.delinquent_acc) AS 'total_customers',
    count(case
    when trev.delinquent_acc = 1 then 1 end) as 'total_delinquent'
FROM
    temp_age_group AS tage
        JOIN
    temp_revenue AS trev ON trev.Client_Num = tage.client_num
GROUP BY 1)
select *,
concat(round(total_delinquent/total_customers*100, 2), '%') as '%_of_total' from cte;
```

age_group	total_custc	total_delinquent	%_of_total
21-35	938	49	5.22%
60+	405	22	5.43%
36-49	5645	358	6.34%
50-60	3120	185	5.93%

Create Stored Procedure

```
-- create stored procedure to search Credit Card details
DELIMITER $$
CREATE PROCEDURE credit_details(IN client_n TEXT)
BEGIN
    SET @sql_query = CONCAT('
        SELECT
            ccd.Client_Num,
            ccd.Card_Category,
            ccd.Total_Trans_Amt,
            ccd.Annual_Fees,
            ccd.Credit_limit,
            cud.Customer_Age,
            cud.Gender,
            cud.Income,
            cud.state_cd
        FROM creditcard_detail AS ccd
        JOIN customers_details AS cud ON ccd.Client_Num = cud.Client_Num
        WHERE ccd.Client_Num IN (', client_n, ')');

    -- Execute the dynamically constructed query
    PREPARE stmt FROM @sql_query;
    EXECUTE stmt;
    DEALLOCATE PREPARE stmt;
END$$
DELIMITER ;
```

```
-- drop stored procedure
drop procedure if exists credit_details;
```

```
-- call the procedure with integer IDs
CALL credit_details('708082083, 708083283, 708084558');
```

Client_Num	Card_Category	Total_Trans_Amt	Annual_Fees	Credit_limit	Customer_Age	Gender	Income	state_cd
708082083	Blue	15149	200	3544	24	F	202326	FL
708083283	Blue	992	445	3421	62	F	5225	NJ
708084558	Blue	1447	140	8258	32	F	14235	NJ

Month on Month Revenue Change

```sql
with cte as (
SELECT
    MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) AS 'Months',
    ROUND(SUM(Annual_Fees + Interest_Earned), 2) AS 'Revenue'
FROM
    creditcard_detail
GROUP BY 1),
cte1 as (
select *, lag(revenue) over() as 'pre_rev' from cte)
SELECT
    months,
    revenue,
    CONCAT(ROUND(COALESCE((revenue - pre_rev) / revenue * 100, 0),
                2),
           '%') AS 'MoM Change'
FROM
    cte1;
```

months	revenue	MoM Change
January	1051279.17	0%
February	847123.67	-24.1%
March	815371.93	-3.89%
April	1010391.37	19.3%
May	820153.39	-23.2%
June	854720.96	4.04%
July	1107554.79	22.83%
August	843300.99	-31.34%
September	834923.91	-1%
October	1004078.28	16.85%
November	824318.76	-21.81%
December	780180.01	-5.66%

Three Month Simple Moving Average

```sql
with cte as
(SELECT
    MONTHNAME(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) as 'Months',
    ROUND(SUM(Annual_Fees + Interest_Earned)) as 'Revenue'
FROM
    creditcard_detail
GROUP BY 1 )
select Months,
round((sum(revenue) over(rows between 2 preceding and current row)) / 3, 2)
as 'Moving AVG'
from
cte;
```

Months	Moving AVG
January	350426.33
February	632801
March	904591.67
April	890962.33
May	881972
June	895088.33
July	927476.33
August	935192.33
September	928593.33
October	894101
November	887773.67
December	869525.67

Quarterly Cumulative Sum

```sql
with cte as (
SELECT
    CONCAT('Q',
            QUARTER(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y'))) AS 'Quarters',
    ROUND(SUM(Annual_Fees + Interest_Earned), 2) AS 'Revenue'
FROM
    creditcard_detail
GROUP BY 1)
select *,
sum(revenue) over(order by quarters) as 'Cumulative Sum'
from
    cte;
```

Quarters	Revenue	Cumulative Sum
Q1	2713774.77	2713774.77
Q2	2685265.72	5399040.49
Q3	2785779.69	8184820.18
Q4	2608577.05	10793397.23

Quarterly Cumulative Sum using Sub-query

```sql
SELECT
    Year,
    Quarter,
    Revenue,
    ROUND(SUM(Revenue) OVER (
        PARTITION BY Year
        ORDER BY Quarter
    ), 2) AS Cumulative_Revenue
FROM (
    SELECT
        YEAR(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y')) AS Year,
        CONCAT('Q', QUARTER(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y'))) AS Quarter,
        ROUND(SUM(Annual_Fees + Interest_Earned), 2) AS Revenue
    FROM
        creditcard_detail
    GROUP BY
        Year, Quarter
) AS aggregated_data
ORDER BY
    Year, Quarter;
```

Year	Quarter	Revenue	Cumulative_Revenue
2023	Q1	2713774.77	2713774.77
2023	Q2	2685265.72	5399040.49
2023	Q3	2785779.69	8184820.18
2023	Q4	2608577.05	10793397.23

```sql
with cte as (
SELECT
    CONCAT('Q',
           QUARTER(STR_TO_DATE(Week_Start_Date, '%d-%m-%Y'))) AS Quarter,
           ROUND(SUM(Annual_Fees + Interest_Earned), 2) AS Revenue
FROM
    creditcard_detail group by quarter),
cte1 as(
select *, coalesce(lag(Revenue) over(), 0) as 'pre_rev' from cte),
cte2 as(
SELECT
    quarter,
    revenue,
    CONCAT(COALESCE(ROUND((revenue - pre_rev) / pre_rev * 100, 2),
                    0),
           '%') AS 'QoQ Change'
FROM
    cte1)
select * from cte2;
```

Quarter on Quarter Change

quarter	revenue	QoQ Change
Q1	2713774.77	0%
Q2	2685265.72	-1.05%
Q3	2785779.69	3.74%
Q4	2608577.05	-6.36%

```sql
with cte1 as (
SELECT
    WEEK(STR_TO_DATE(week_start_date, '%d-%m-%Y')) AS 'Week_No',
    ROUND(SUM(Annual_Fees) + SUM(Interest_Earned)) AS 'Revenue',
    coalesce(lag(ROUND(SUM(Annual_Fees) + SUM(Interest_Earned)))
    over(order by WEEK(STR_TO_DATE(week_start_date, '%d-%m-%Y'))),
    0) as 'Pre_rev'
FROM
    creditcard_detail
GROUP BY 1),
cte2 as(
SELECT
    *,
    COALESCE(CONCAT(ROUND((((revenue - pre_rev) / pre_rev) * 100),
                          2),
                    '%'),
             0) AS 'WoW_Change', rank()
             over(order by round((((revenue - pre_rev) / pre_rev) * 100), 2))
as 'rnk'
from cte1)
SELECT
    Week_No, Revenue, WoW_Change
FROM
    cte2
WHERE
    rnk BETWEEN 48 AND 52
        OR rnk BETWEEN 2 AND 6
ORDER BY 3 DESC;
```

Top 5 Most Positive and Negative Weeks by Revenue

Week_No	Revenue	WoW_Change
31	231640	9.58%
21	216612	9.01%
27	240096	8.81%
44	214504	15.31%
11	215556	10.69%
13	188624	-12.86%
43	186031	-11.9%
32	205027	-11.49%
39	196802	-11.15%
22	194461	-10.23%

THANK YOU