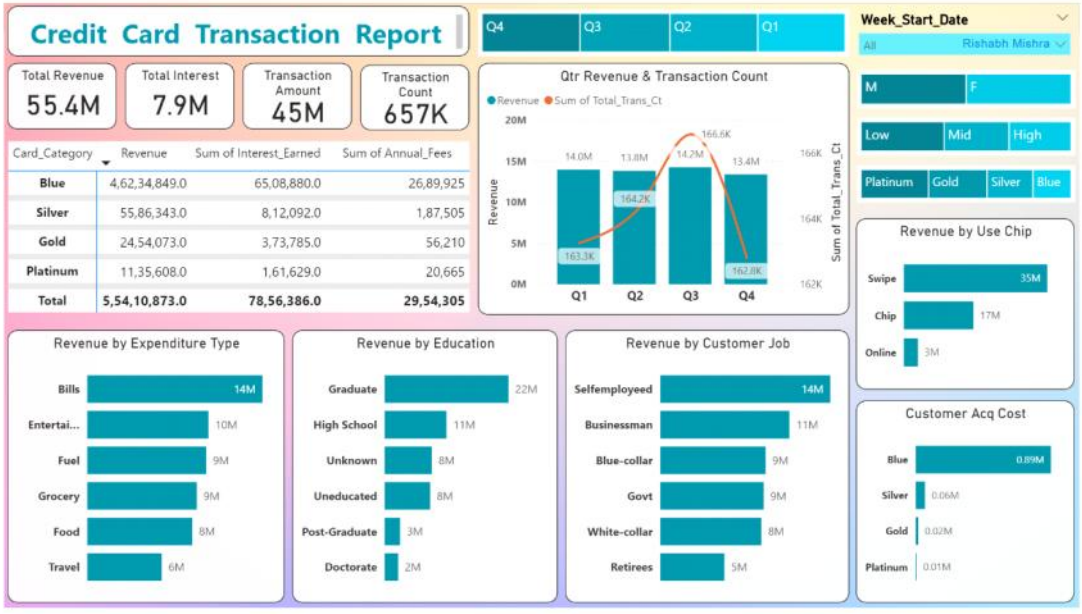
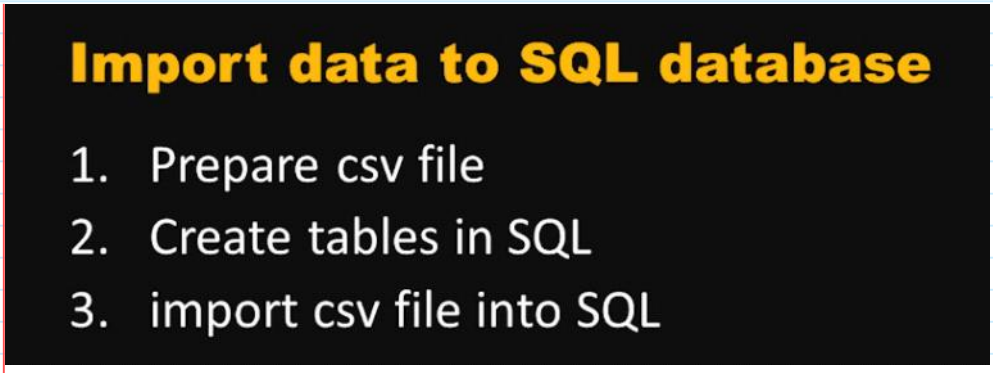
POWER BI project

08 September 2025 09:28

Credit Card Financial Dashboard



1st we will import the data from the sql database



First we need to have a data in sql then only we can import in the powerbi

When you work on a real world project we have data inside sql and we bring that from data pipelines or multiple pipelines

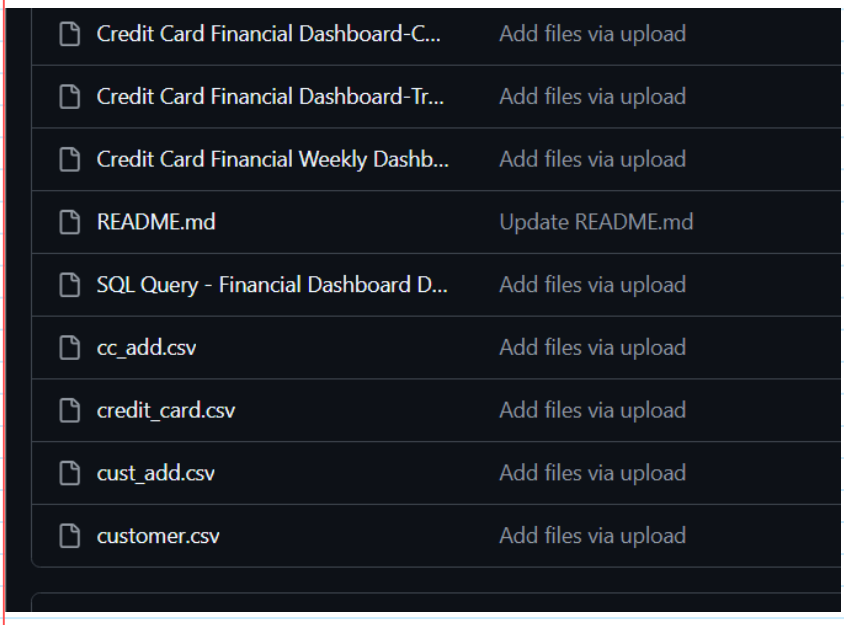
But right now we will do the above image instructions..

Once we get data in the sql we will connect it to power bi and then gen

Customer.csv - credit card customer dashboard

Creditcard.csv - credit card transaction dashboard

Remaining two .csv files are the additional files which will be used to update the data within the same dashboard.



Use sql to import

Created a database and before that established connection by win+ run - services.msc -> mysqlservices click to start service.

Opened it created a database named ccdb

Then used it

Then created 2 tables

```
);

CREATE TABLE cust_detail (
    Client_Num INT,
    Customer_Age INT,
    Gender VARCHAR(5),
    Dependent_Count INT,
    Education_Level VARCHAR(50),
    Marital_Status VARCHAR(20),
    State_cd VARCHAR(50),
    Zipcode VARCHAR(20),
    Car_Owner VARCHAR(5),
    House_Owner VARCHAR(5),
    Personal_Loan VARCHAR(5),
    Contact VARCHAR(50),
    Customer_Job VARCHAR(50),
    Income INT,
    Cust_Satisfaction_Score INT
);
```

```
CREATE TABLE cc_detail (
    Client_Num INT,
    Card_Category VARCHAR(20),
    Annual_Fees INT,
    Activation_30_Days INT,
    Customer_Acq_Cost INT,
    Week_Start_Date DATE,
    Week_Num VARCHAR(20),
    Qtr VARCHAR(10),
    current_year INT,
    Credit_Limit DECIMAL(10,2),
    Total_Revolving_Bal INT,
    Total_Trans_Amt INT,
    Total_Trans_Ct INT,
    Avg_Utilization_Ratio DECIMAL(10,3),
    Use_Chip VARCHAR(10),
    Exp_Type VARCHAR(50),
    Interest_Earned DECIMAL(10,3),
    Delinquent_Acc VARCHAR(5)
);
```

NOW

To get the data from csv we will use copy for postgresql but different in mysql

```
LOAD DATA LOCAL INFILE 'D:/Power BI proj/credit_card.csv'
INTO TABLE cc_detail
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

select * from cc_detail;

LOAD DATA LOCAL INFILE 'D:/Power BI proj/customer.csv'
INTO TABLE cust_detail
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
select * from cust_detail;
```

And before that you need to enable local infiles

What to do:

1. In the **Others** box (where you see `lastConnected=...` and `serverVersion=`), add this line on a **new line**:

```ini
OPT_LOCAL_INFILE=1
```

Example:

```ini
lastConnected=1757650470
serverVersion=
OPT_LOCAL_INFILE=1
```

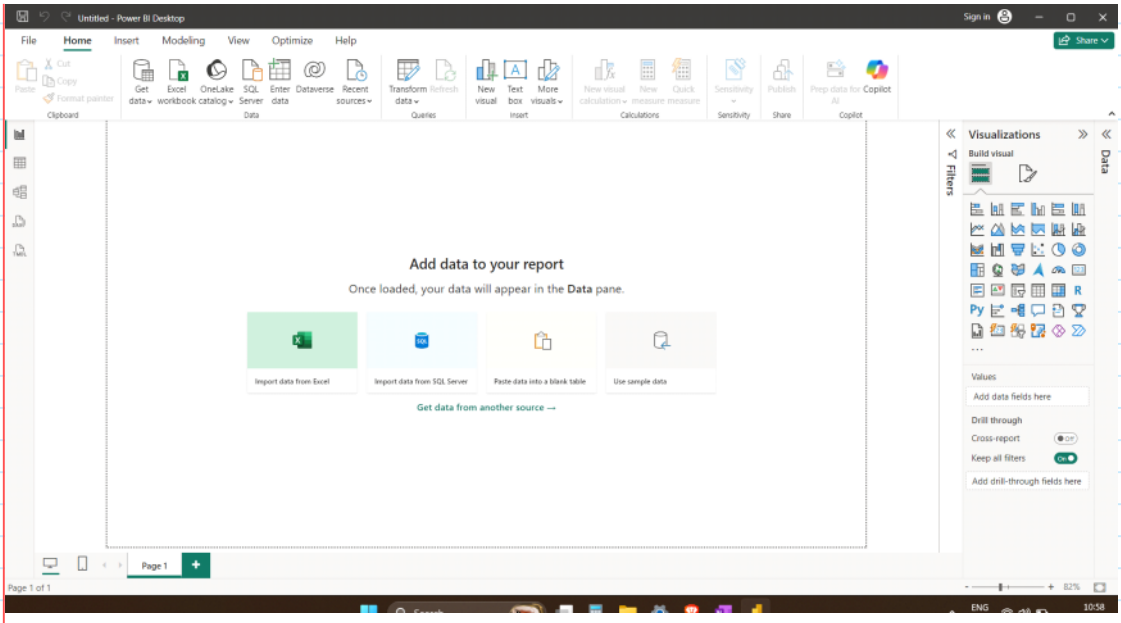
2. Click **Test Connection** → if it succeeds, click **OK** to save.

```sql
SHOW GLOBAL VARIABLES LIKE 'local_infile';
```

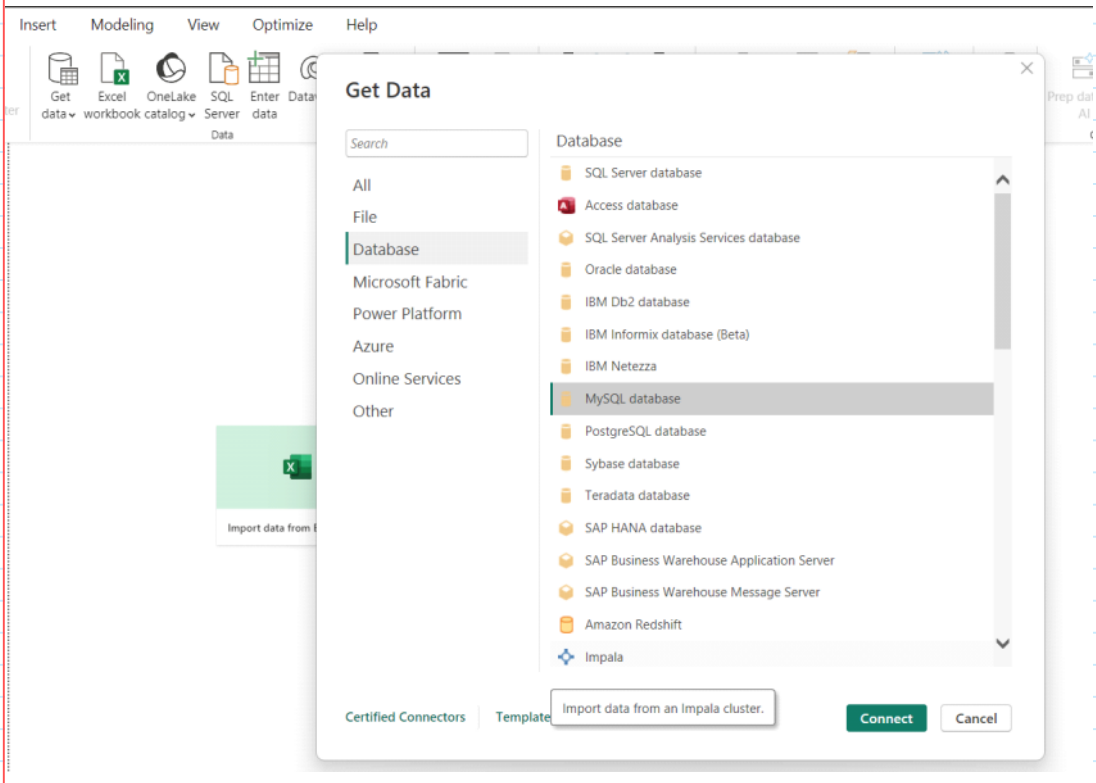
If it's `OFF`, turn it on:

```sql
SET GLOBAL local_infile = 1;
```

Now we need to setup the connections between powerbi and sql



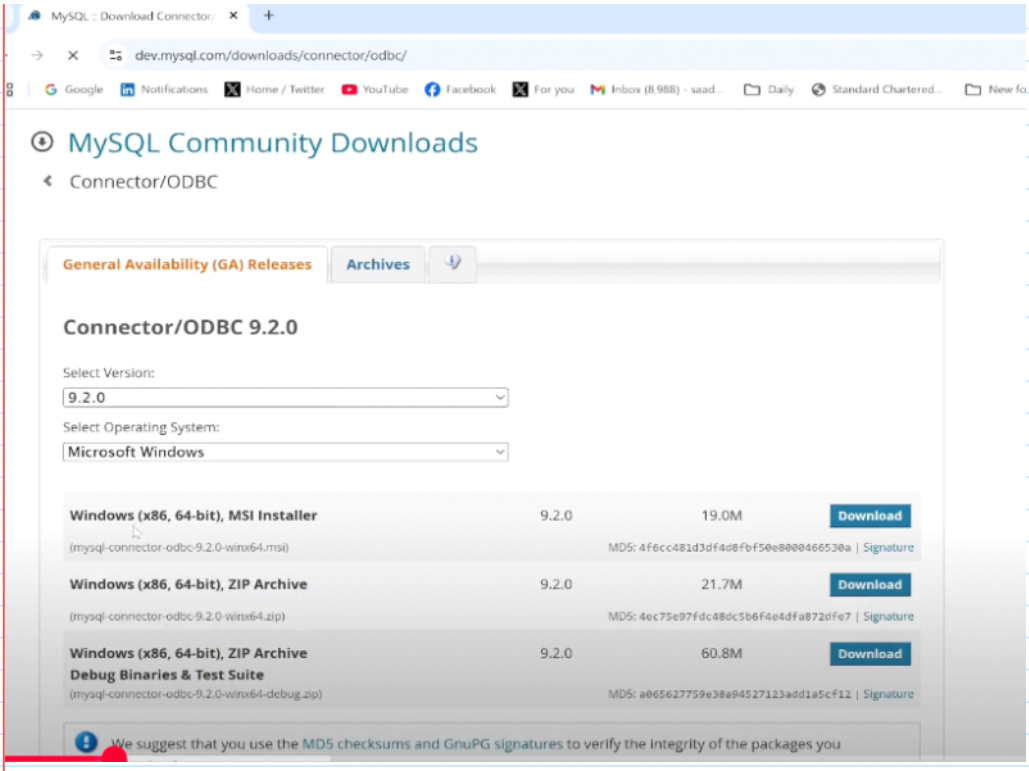
Power bi interface



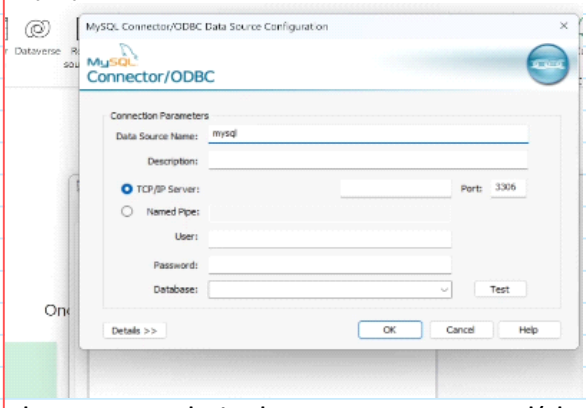
Click on get data and then connect it with mysql

Before that you need to install a connector to establish a connection between mysql and powerbi

Go to mysql odbc download and there select the version and install it



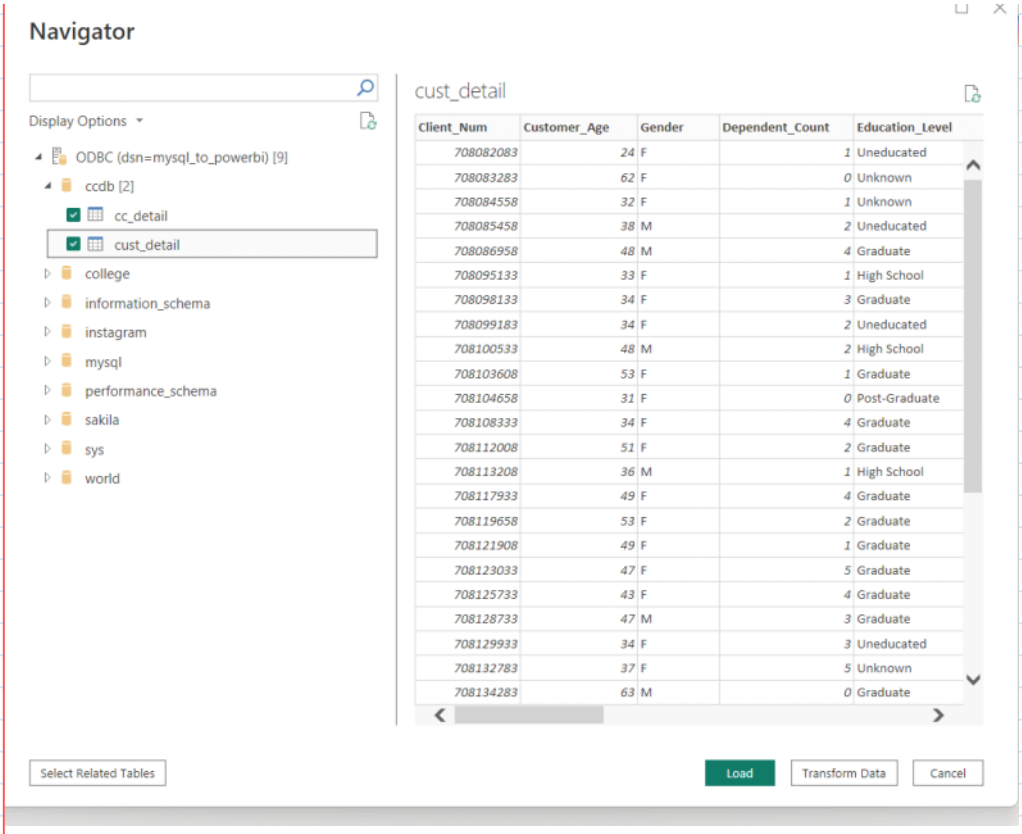
Go to windows -> odbc -> go to systemDSN -> click add -> click mysqlodbcansi ->



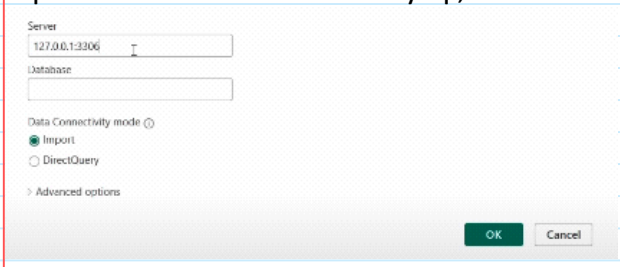
The password - is the master password(the one which we use while installing the sql) which was in mycase Arman123$
Then test connection (database same as u want to connect with powerbi)

Proceed to powerbi

And after that click get data -> other - > odbc -> provide the credentials then connect , it will show

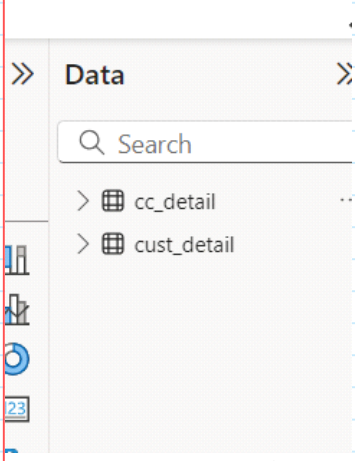


Or else
If powerbi is able to detect mysql, then->

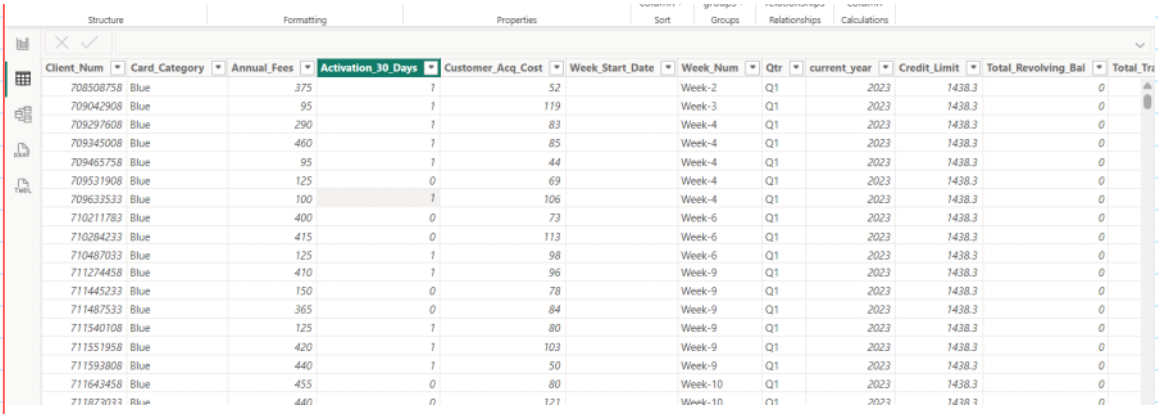


Then load

Now the data has been loaded



you can also get it from the table view in left side



Client_Num	Card_Category	Annual_Fees	Activation_30_Days	Customer_Acq_Cost	Week_Start_Date	Week_Num	Qtr	current_year	Credit_Limit	Total_Revolving_Bal
708508758	Blue	375	1	52		Week-2	Q1	2023	1438.3	0
709042908	Blue	95	1	119		Week-3	Q1	2023	1438.3	0
709297608	Blue	290	1	83		Week-4	Q1	2023	1438.3	0
709345008	Blue	460	1	85		Week-4	Q1	2023	1438.3	0
709465758	Blue	95	1	44		Week-4	Q1	2023	1438.3	0
709531908	Blue	125	0	69		Week-4	Q1	2023	1438.3	0
709633533	Blue	100	1	106		Week-4	Q1	2023	1438.3	0
710211783	Blue	400	0	73		Week-6	Q1	2023	1438.3	0
710284233	Blue	415	0	113		Week-6	Q1	2023	1438.3	0
710487033	Blue	125	1	98		Week-6	Q1	2023	1438.3	0
711274458	Blue	410	1	96		Week-9	Q1	2023	1438.3	0
711445233	Blue	150	0	78		Week-9	Q1	2023	1438.3	0
711487533	Blue	365	0	84		Week-9	Q1	2023	1438.3	0
711540108	Blue	125	1	80		Week-9	Q1	2023	1438.3	0
711551958	Blue	420	1	103		Week-9	Q1	2023	1438.3	0
711593808	Blue	440	1	50		Week-9	Q1	2023	1438.3	0
711643458	Blue	455	0	80		Week-10	Q1	2023	1438.3	0

Data processing and DAX queries

First save it -> mine is saved in powerbi folder in D:

So before data processing we need to data cleaning.. And we should prefer it in sql

,

Because if we apply too much filters/ apply many functions then the dashboard will slow down, as whenever we open powerbi it runs all its functions, and formulas first then only it shows the output.

So it is much preferrable that we do data cleaning in mysql

First loading into mysql - >

load data infile 'pathto csv' into table <tablename> fields terminated by ',' enclosed by ' " ' lines terminated by '\n' ignore 1 rows

Ignore 1 rows - > will ignore the header of csv

If you have MySQL, for example:

sql Copy code

```
LOAD DATA INFILE 'path/to/customer.csv'
INTO TABLE customer
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

Now your data is in SQL tables.

Data is loaded into sql from csv

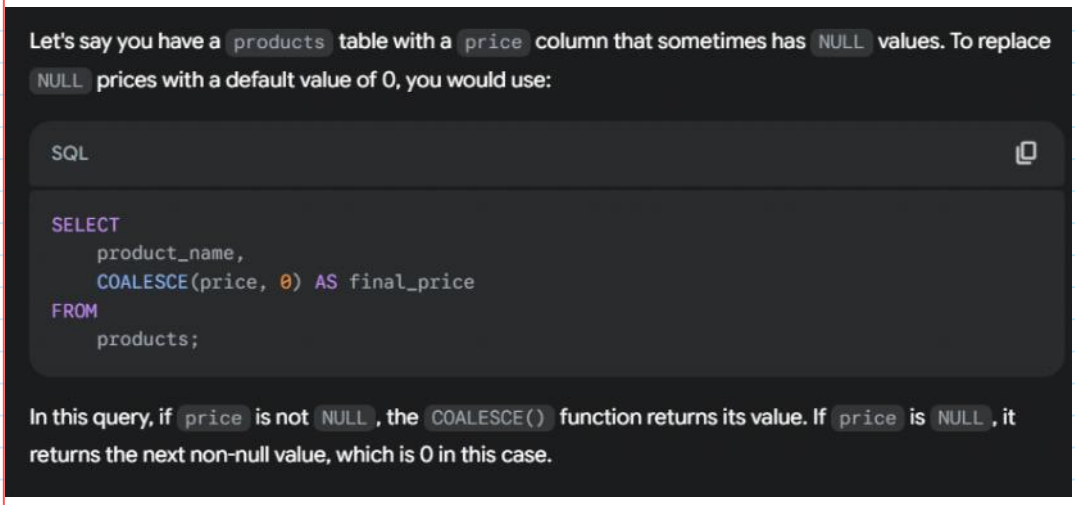
Explore the data

What needs cleaning?
Data needs cleaning for eg:

Check if any column has null values

Select
sum(case when Annual_Fees is null then 1 else 0 end) as
Missing_Annual_Fees,
Sum(case when Credit_Limit is null then 1 else 0 end) as missing_credit_limit
From credit_card_data;

Or



2.Remove Duplicates

Select Client_Num, Count(*) from credit_card_data
group by Client_Num having count(*) >1;

If duplicates exist then delete

Delete from credit_card_data where client_num in
(select client_num from credit_card_data group by client_num having
count(*) > 1);

2. Remove Duplicates

```sql
SELECT Client_Num, COUNT(*)
FROM credit_card_data
GROUP BY Client_Num
HAVING COUNT(*) > 1;
```

If duplicates exist, delete them:

```sql
DELETE FROM credit_card_data
WHERE Client_Num IN (
    SELECT Client_Num
    FROM credit_card_data
    GROUP BY Client_Num
    HAVING COUNT(*) > 1
);
```

3.Fix data types:

Make sure the numeric columns are numbers, dates are dates

3. Fix Data Types

Make sure numeric columns are numbers, dates are dates.

```sql
ALTER TABLE credit_card_data
ALTER COLUMN Week_Start_Date DATE;

ALTER TABLE credit_card_data
ALTER COLUMN Credit_Limit INT;
```

- ALTER TABLE credit_card_data ALTER COLUMN Week_Start_Date DATE; This statement changes the data type of the Week_Start_Date column to **DATE**. This is useful for ensuring that the values in this column are stored in a standardized date format, which allows for proper date-based operations like sorting, filtering by range, and date calculations.

- ALTER TABLE credit_card_data ALTER COLUMN Credit_Limit INT;
This statement changes the data type of the Credit_Limit column to **INT** (integer). This is appropriate for a column that should contain whole numbers, ensuring that the credit limit values are stored as a numeric type and can be used in mathematical calculations.

4. Standardize Categorical Values

For example, `Use Chip` column might have inconsistent values ("Chip Enabled", "chip enabled", "CHIP").

sql Copy code

```sql
UPDATE credit_card_data
SET [Use Chip] = 'Chip Enabled'
WHERE LOWER([Use Chip]) = 'chip enabled';
```

5. Handle Outliers

Check if values fall outside expected range. Example: Credit limit.

sql Copy code

```sql
SELECT MIN(Credit_Limit), MAX(Credit_Limit), AVG(Credit_Limit)
FROM credit_card_data;
```

✅ If you find `Credit_Limit = 0` or `Credit_Limit > 1,000,000`, you can cap or remove them.

6. Format Dates & Extract Features

Extract useful parts for analysis:

sql Copy code

```sql
ALTER TABLE credit_card_data ADD Month INT;
UPDATE credit_card_data
SET Month = MONTH(Week_Start_Date);
```

7. Derived / Calculated Columns

For example, utilization ratio = `Total_Revolving_Bal / Credit_Limit`.

sql Copy code

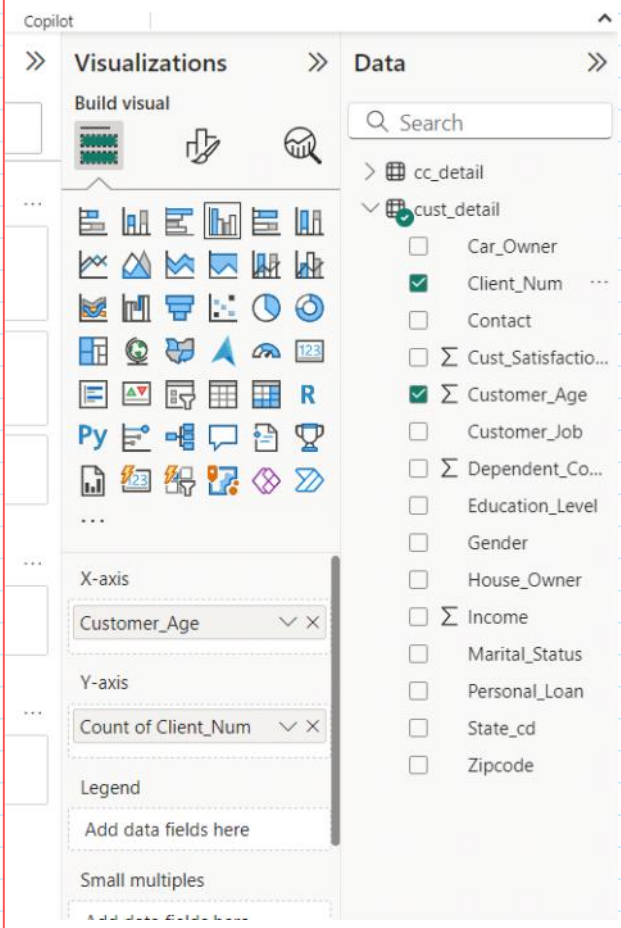
```sql
ALTER TABLE credit_card_data ADD Calc_Utilization_Ratio DECIMAL(5,2);

UPDATE credit_card_data
SET Calc_Utilization_Ratio = CAST(Total_Revolving_Bal AS FLOAT) / NULLIF(Credi
```

ALTER TABLE credit_card_data ADD Calc_Utilization_Ratio DECIMAL(5,2);

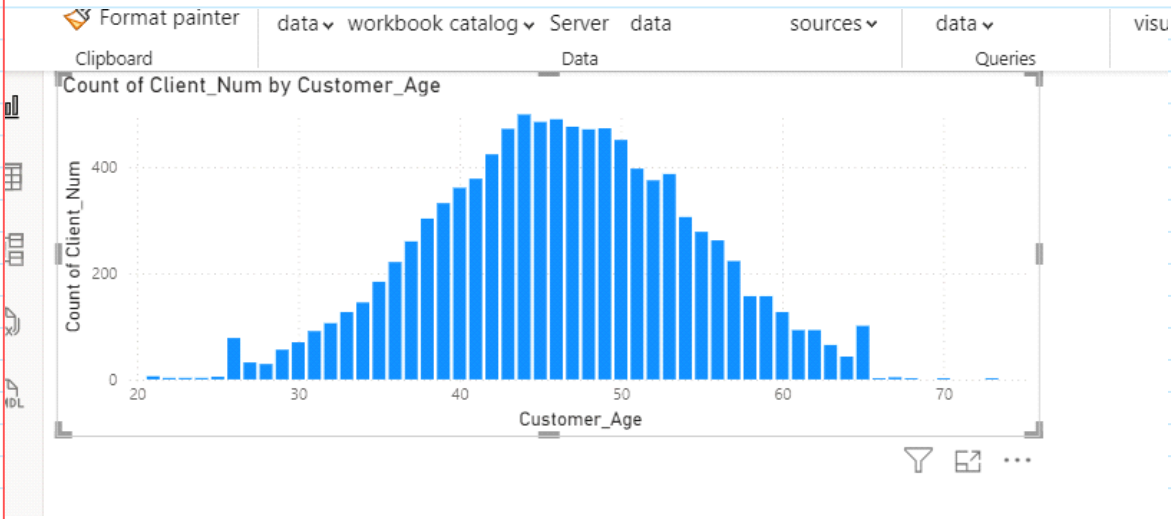
UPDATE credit_card_data
SET Calc_Utilization_Ratio = CAST(Total_Revolving_Bal AS FLOAT) /
NULLIF(Credit_Limit, 0);

After this move to the powerbi

For suppose If I want to make a chart based on the age

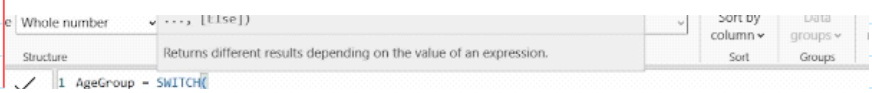


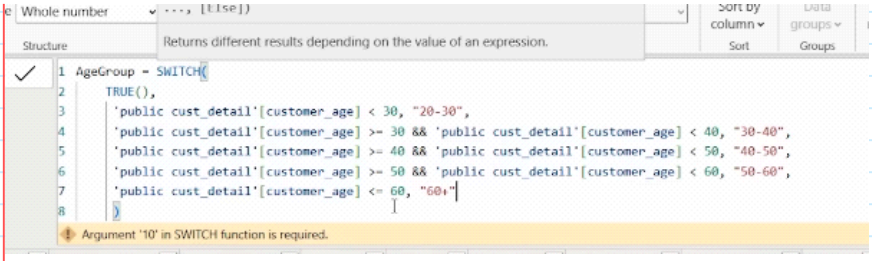
Dragged and dropped into the x and y axis



So if we wanted a group wise chart as the above chart doesn't make so much sense..

So a new column with agegroup is created and then in top function section



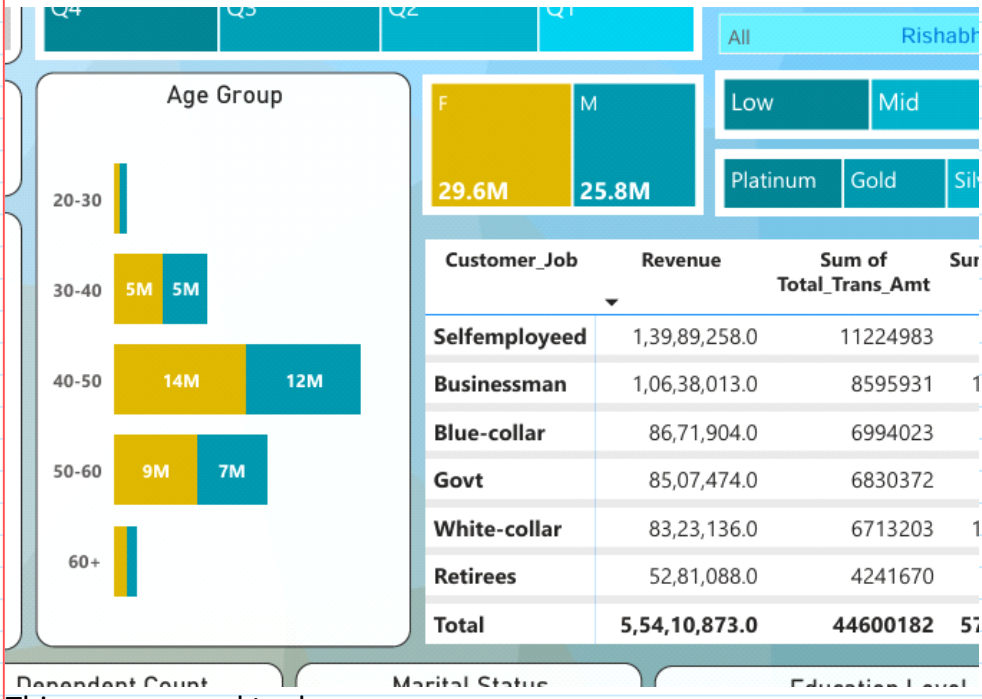


Also for safety purpose / default - "unknown"

Ok moving to the project

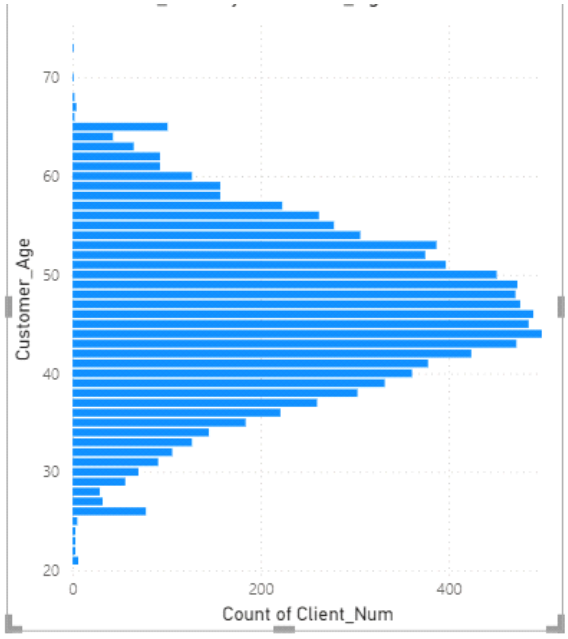
DAX queries,

Now first we need agegroup as



This one we need to do

Normally when we import bar chart with customer age as x axis and count of client as y axis

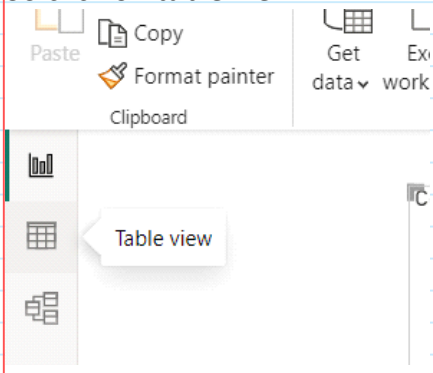


It shows like this which is not efficient or unclear

So we need to group the age so to get an estimate..

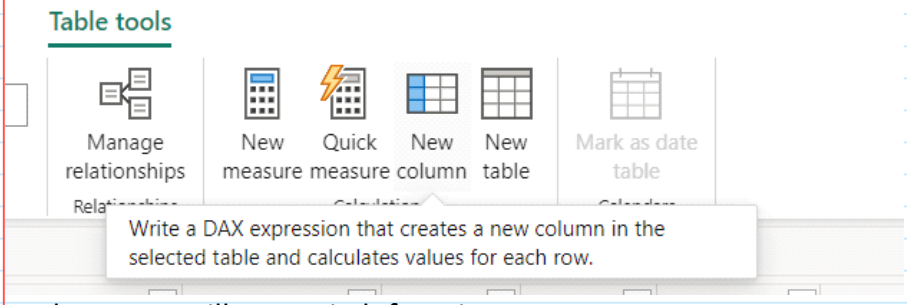
And same we will do with the income,

So click on table view



Click on customer detail,
Click on add new column

And then we need to write the function



And now we will use switch function
Which is useful in using multiple conditions

Syntax- switch(expression, value1 , result, value2 , result2,……,default)
-function: AgeGroup = Switch(
True(),

'public cust_detail'[customer_age]<30 , "20-30",
'public cust_detail'[customer_age]>=30 && 'public cust_detail'
[customer_age] <40,"30-40",
...

```
AgeGroup = SWITCH(
    TRUE(),
    'cust_detail'[Customer_Age]<30,"Below 30",
    'cust_detail'[Customer_Age]>=30 && 'cust_detail'[Customer_Age]<40, "30-40",
    'cust_detail'[Customer_Age]>=40 && 'cust_detail'[Customer_Age]<50, "40-50",
    'cust_detail'[Customer_Age]>=50 && 'cust_detail'[Customer_Age]<60, "50-60",
    'cust_detail'[Customer_Age]>60,"60+",
    "unknown"
)
```

And we can see that new column with the values has been created

	AgeGroup
1	50-60
2	30-40
2	50-60
3	40-50
5	40-50
2	50-60
2	50-60
3	50-60
1	40-50
1	40-50
1	40-50
3	50-60
3	50-60
3	50-60
3	30-40
3	50-60
3	40-50
3	40-50

Same method use - create income groups

```
IncomeGroup = SWITCH(
    TRUE(),
    'cust_detail'[Income]<35000,"Low",
    'cust_detail'[Income]>=35000 && 'cust_detail'[Income]<75000,"Med",
    'cust_detail'[Income]>=75000,"High",
    "unknown"
)
```

Marital_Status	State_cd	Zipcode	Car_Owner	House_Owner	Personal_Loan

mer_Job	Income	Cust_Satisfaction_Score	AgeGroup	IncomeGroup
ollar	37378	3	40-50	Med
nployeed	57111	3	40-50	Med
nployeed	1250	2	40-50	Low
-collar	64783	2	40-50	Med
es	48210	3	50-60	Med
nployeed	87414	2	40-50	High
nployeed	14975	1	40-50	Low
ssman	43175	3	40-50	Med
	112593	2	40-50	High
nployeed	39032	2	40-50	Med
	57930	2	30-40	Med
	76798	1	30-40	High
es	22553	1	40-50	Low

Go to cc detail and create new revenue col

```
1 Revenue = 'cc_detail'[Annual_Fees]+'cc_detail'[Total_Trans_Amt]+'cc_detail'[Interest_Earned]
```

rent_year	Credit_Limit	Total_Revolving_Bal	Total_Trans_Amt	Total_Trans_Ct	Avg_Utilization_Ratio	Use_Chip	Exp_Type	Interest_Earned	Delinquent_Acc	Revenue
2023	1438.3	0	692	16	0	Swipe	Bills	76.12	0	1143.12
2023	1438.3	0	2207	68	0	Swipe	Bills	154.49	0	2456.49
2023	1438.3	0	1585	33	0	Swipe	Bills	301.15	0	2176.15

So the card report is based on week so we will do it as:

So when we import date from the col we will see that the correct sorting is not in proper format as

Week 1 week10 11…14 week 2

But we want week1,week2…

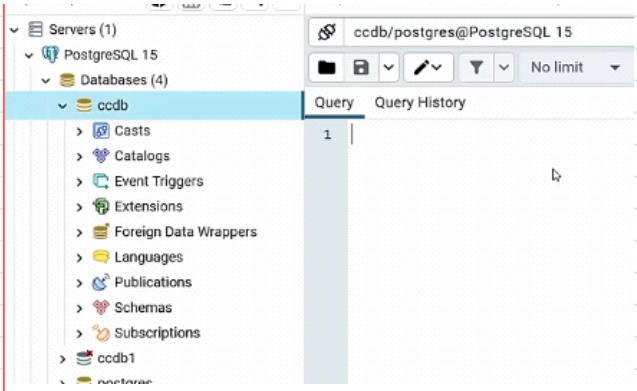
So new column we'll make

I failed to change dataset for date

Postgre sql - password Arman123$

Create database ccdb same as mysql ,

Then rightclick on Databases and refresh



Then click on the query after clicking on the ccdb database

Now currently the schemas -> table

Created the table using prev code before

Cc_detail , cust_detail..

Now import the data
Copy function

```
copy cc_detail
from 'D:\Power BI proj\credit_card.csv'
delimiter ','
csv header

select * from cc_detail;
```

Data Output | Messages | Notifications

Showing rows: 1 to 1000 | Page No: 1 of 11

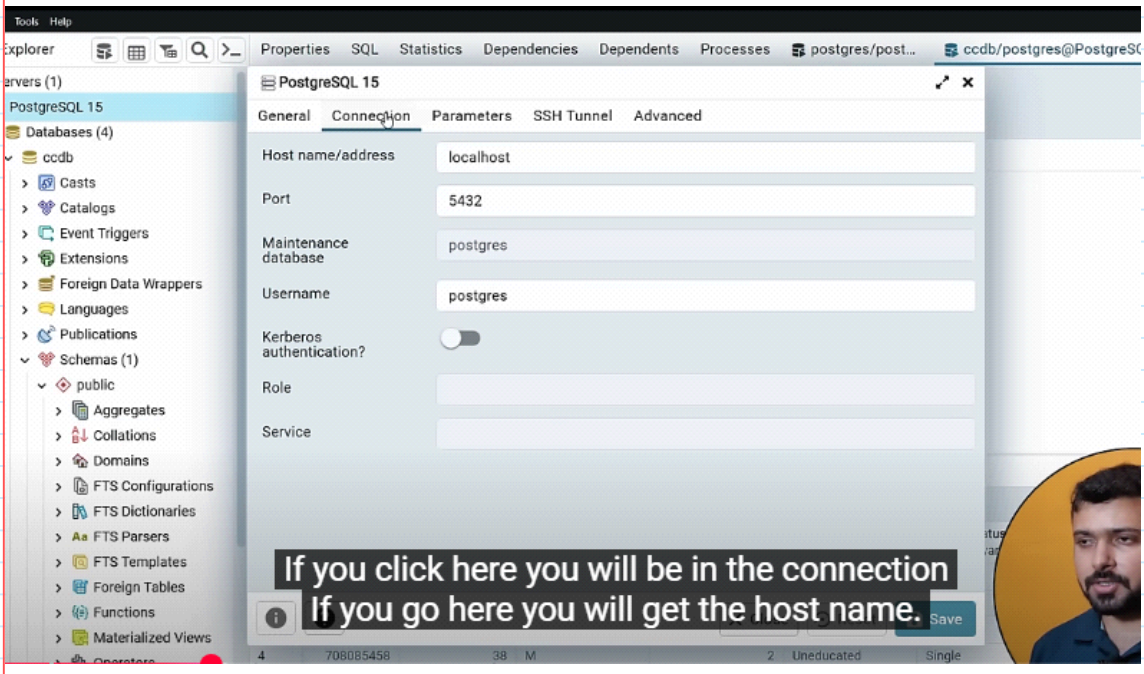
	client_num integer	card_category character varying (20)	annual_fees integer	activation_30_days integer	customer_acq_cost integer	week_start_date date
1	708082083	Blue	200	0	87	2023-01-01
2	708083283	Blue	445	1	108	2023-01-01
3	708084558	Blue	140	0	106	2023-01-01
4	708085458	Blue	250	1	150	2023-01-01
5	708086958	Blue	320	1	106	2023-01-01
6	708095133	Blue	100	0	94	2023-01-01
7	708098133	Blue	225	1	75	2023-01-01
8	708099183	Blue	400	1	75	2023-01-01

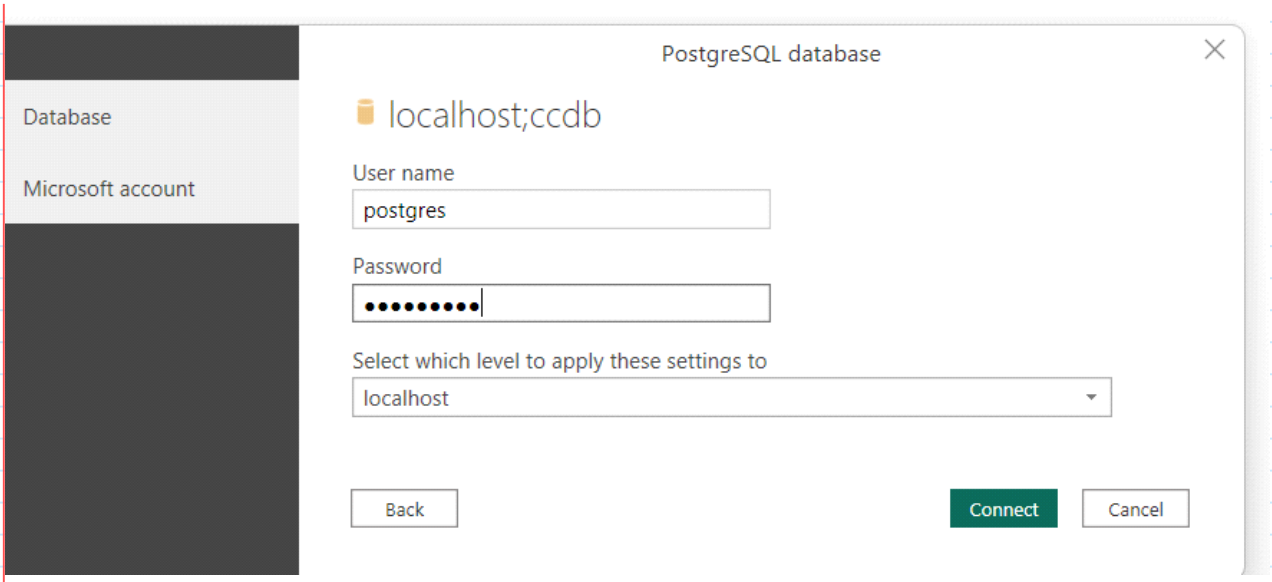
```sql
select * from cust_detail;
copy cust_detail
from 'D:\Power BI proj\customer.csv'
delimiter ','
csv header
```

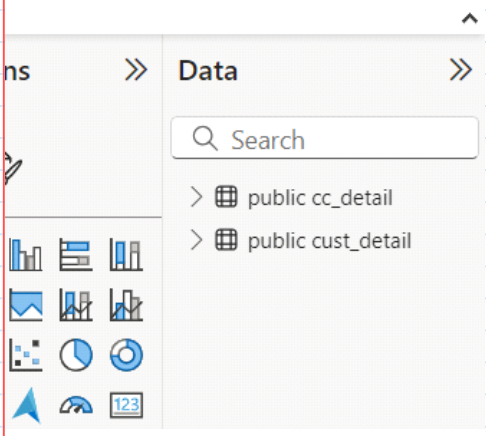
Data Output | Messages | Notifications

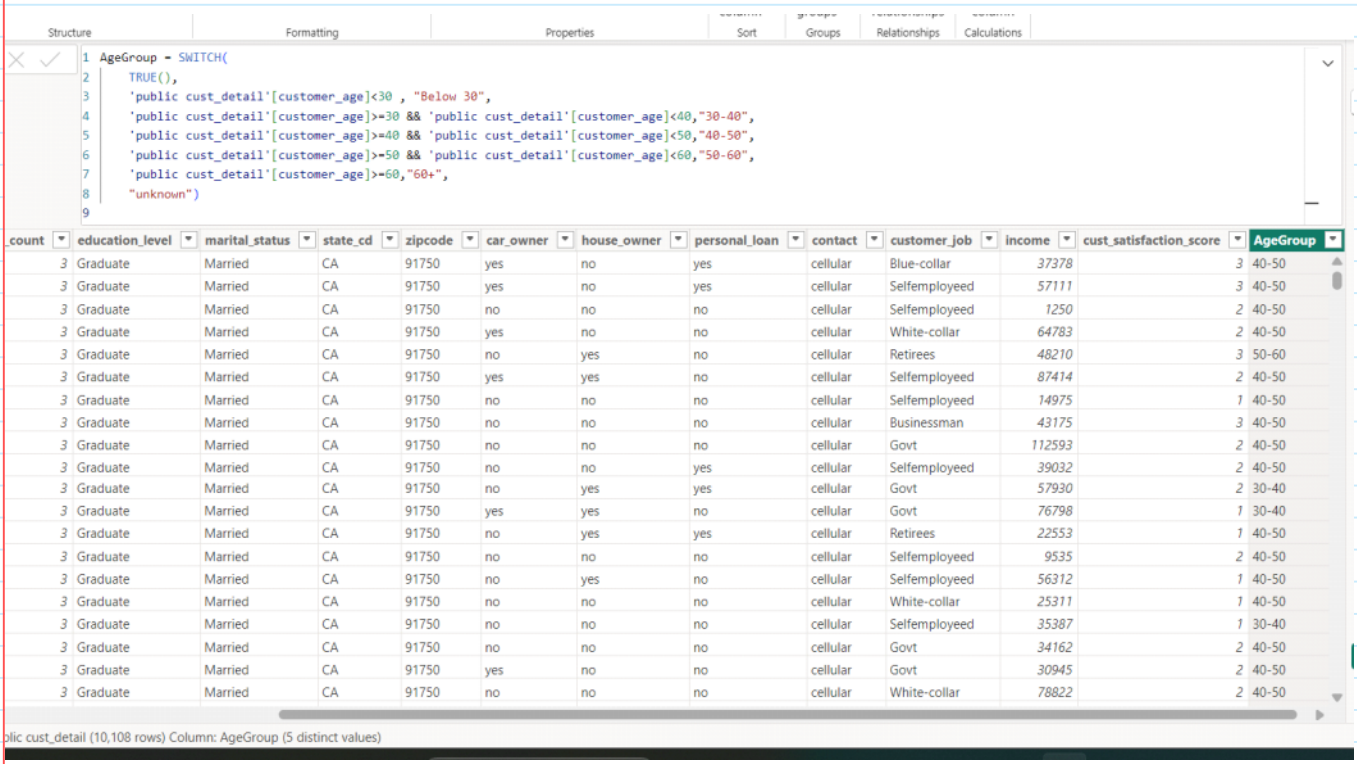
Showing rows: 1 to 1000 | Page No: 1 of 11

	client_num integer	customer_age integer	gender character varying (5)	dependent_count integer	education_level character varying (50)	marital_status character varying
1	708082083	24	F	1	Uneducated	Single
2	708083283	62	F	0	Unknown	Married
3	708084558	32	F	1	Unknown	Married
4	708085458	38	M	2	Uneducated	Single
5	708086958	48	M	4	Graduate	Single
6	708095133	33	F	1	High School	Single
7	708098133	34	F	3	Graduate	Single
8	708099183	34	F	2	Uneducated	Single



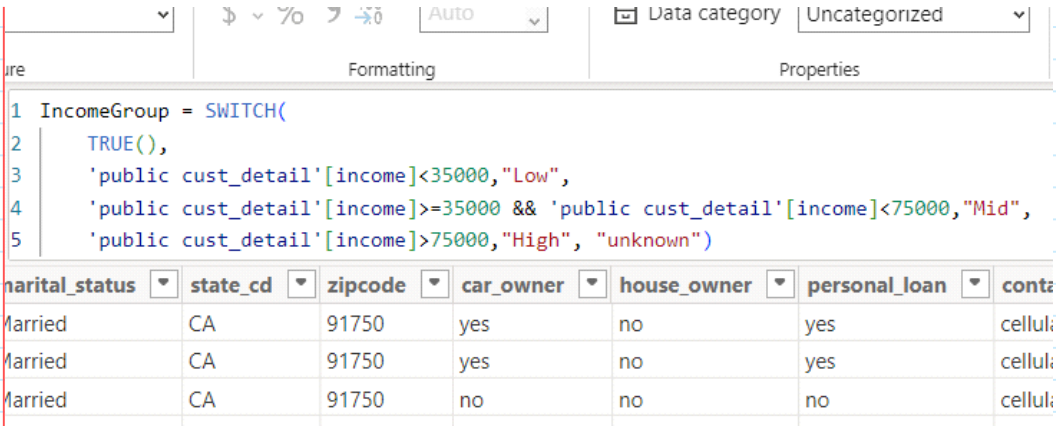




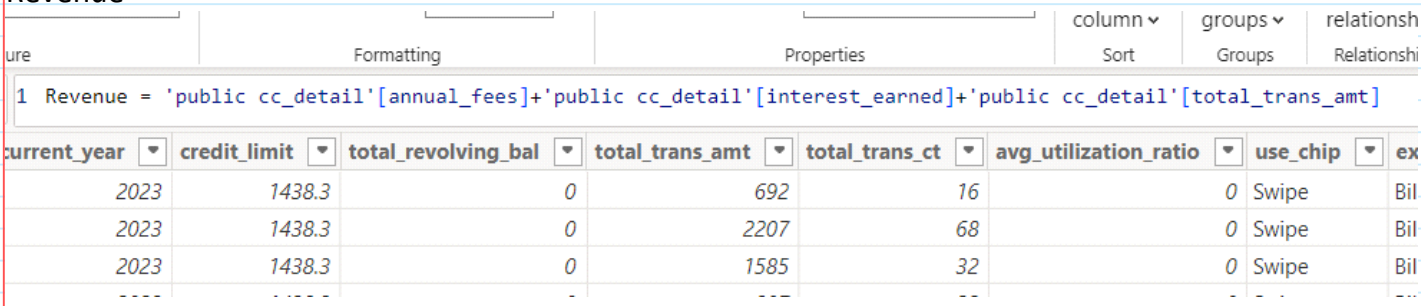


_count	education_level	marital_status	state_cd	zipcode	car_owner	house_owner	personal_loan	contact	customer_job	income	cust_satisfaction_score	AgeGroup
3	Graduate	Married	CA	91750	yes	no	yes	cellular	Blue-collar	37378	3	40-50
3	Graduate	Married	CA	91750	yes	no	yes	cellular	Selfemployeed	57111	3	40-50
3	Graduate	Married	CA	91750	no	no	no	cellular	Selfemployeed	1250	2	40-50
3	Graduate	Married	CA	91750	yes	no	no	cellular	White-collar	64783	2	40-50
3	Graduate	Married	CA	91750	no	yes	no	cellular	Retirees	48210	3	50-60
3	Graduate	Married	CA	91750	yes	yes	no	cellular	Selfemployeed	87414	2	40-50
3	Graduate	Married	CA	91750	no	no	no	cellular	Selfemployeed	14975	1	40-50
3	Graduate	Married	CA	91750	no	no	no	cellular	Businessman	43175	3	40-50
3	Graduate	Married	CA	91750	no	no	no	cellular	Govt	112593	2	40-50
3	Graduate	Married	CA	91750	no	no	yes	cellular	Selfemployeed	39032	2	40-50
3	Graduate	Married	CA	91750	no	yes	yes	cellular	Govt	57930	2	30-40
3	Graduate	Married	CA	91750	yes	yes	no	cellular	Govt	76798	1	30-40
3	Graduate	Married	CA	91750	no	yes	yes	cellular	Retirees	22553	1	40-50
3	Graduate	Married	CA	91750	no	no	no	cellular	Selfemployeed	9535	2	40-50
3	Graduate	Married	CA	91750	no	yes	no	cellular	Selfemployeed	56312	1	40-50
3	Graduate	Married	CA	91750	no	no	no	cellular	White-collar	25311	1	40-50
3	Graduate	Married	CA	91750	no	no	no	cellular	Selfemployeed	35387	1	30-40
3	Graduate	Married	CA	91750	no	no	no	cellular	Govt	34162	2	40-50
3	Graduate	Married	CA	91750	yes	no	no	cellular	Govt	30945	2	40-50
3	Graduate	Married	CA	91750	no	no	no	cellular	White-collar	78822	2	40-50

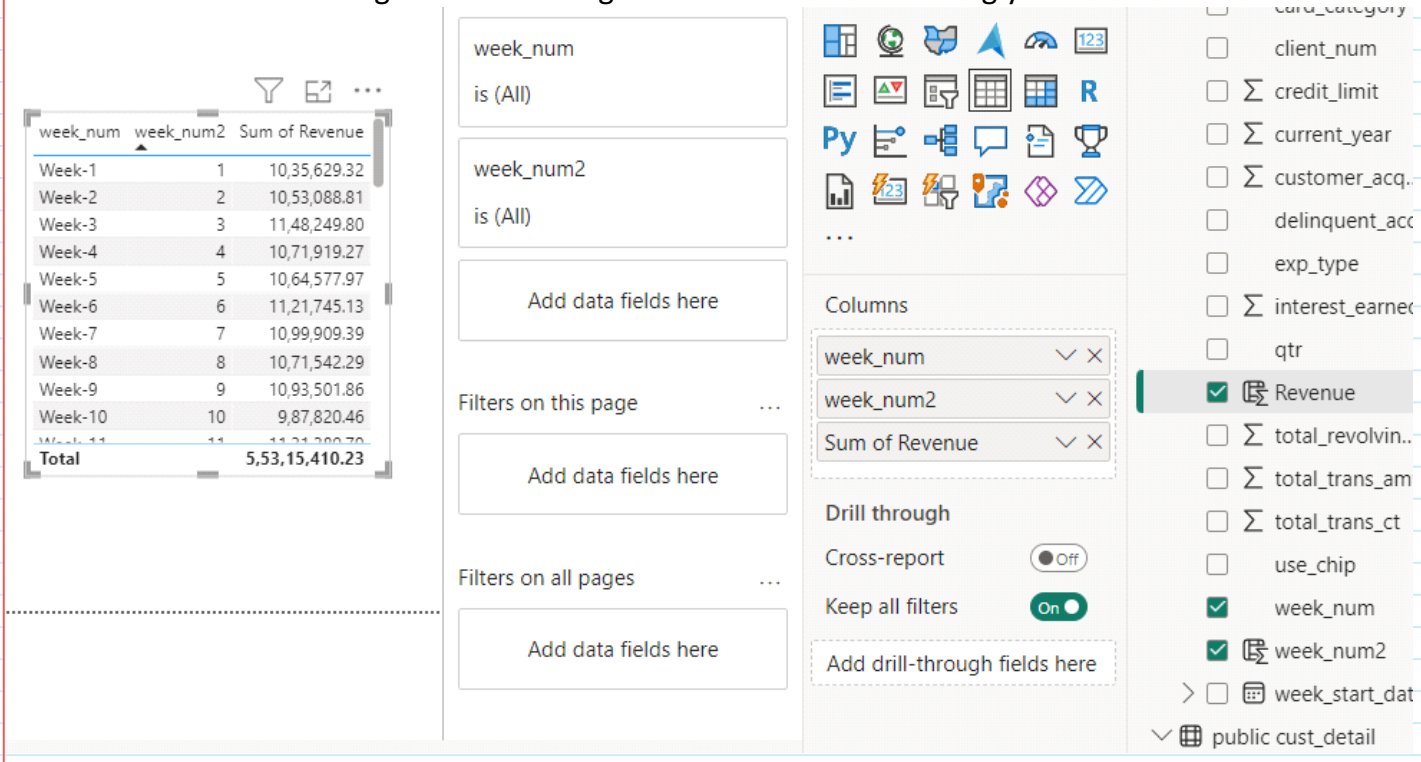
Same with income group



Revenue



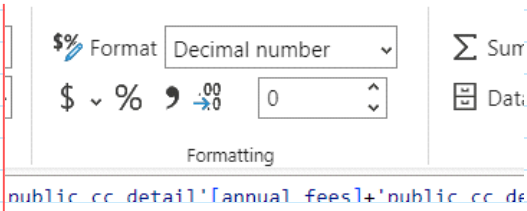
Now we sort the week using weeknum and generated revenue accordingly



Now we will calculate revenue by week on week

If we want to modify revenue column then click on the right side(revenue function button again)

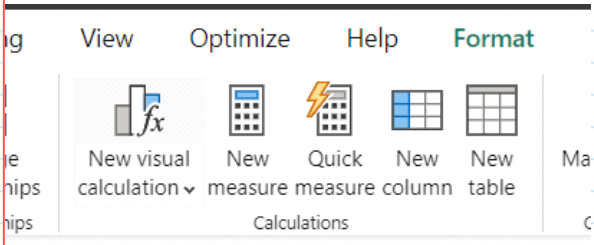
Click on upper arrow and now the decimal - 0



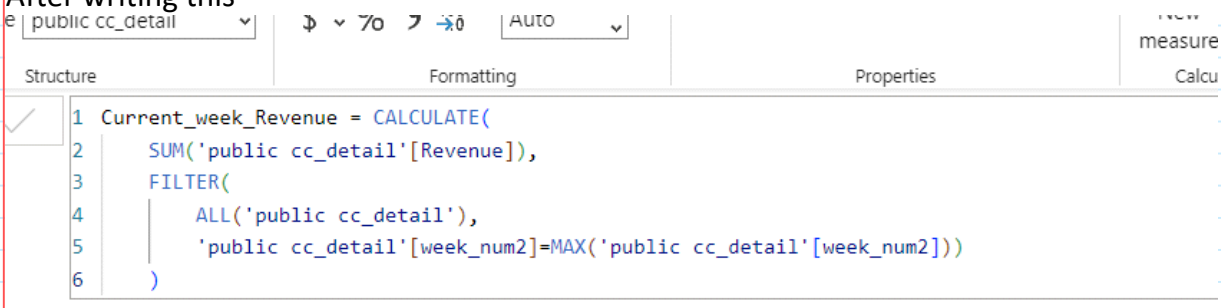
Now for current week revenue we will use a function

Click on public cc_detail on right hand side and then

Click on new measure



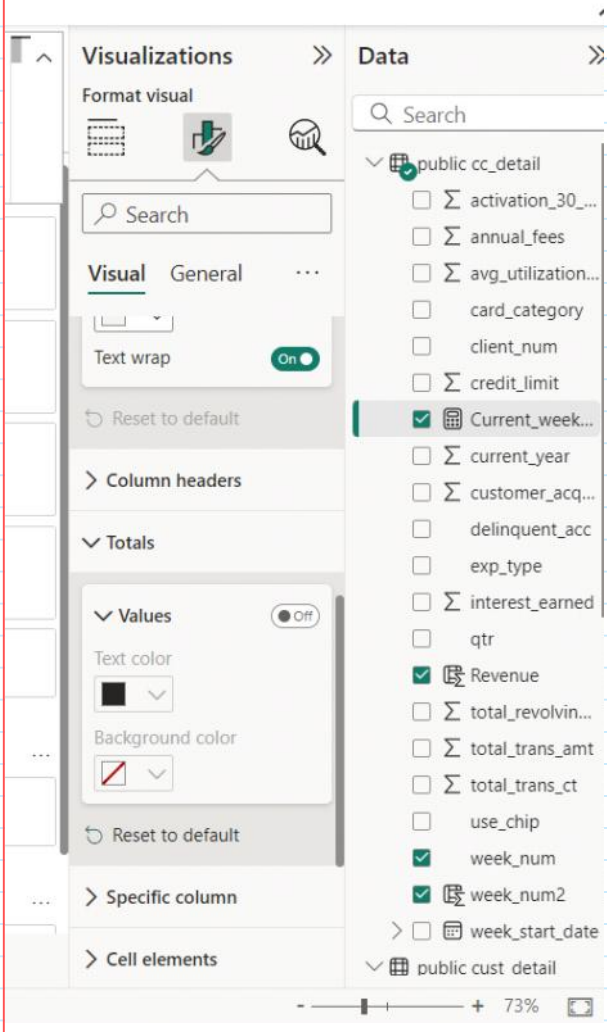
After writing this



You will realise that it will generate the same but it's a little different

week_num	week_num2	Sum of Revenue	Current_week_Revenue
Week-1	1	1035629	10,35,629.32
Week-2	2	1053089	10,53,088.81
Week-3	3	1148250	11,48,249.80
Week-4	4	1071919	10,71,919.27
Week-5	5	1064578	10,64,577.97
Week-6	6	1121745	11,21,745.13
Week-7	7	1099909	10,99,909.39
Week-8	8	1071542	10,71,542.29
Week-9	9	1093502	10,93,501.86
Week-10	10	987820	9,87,820.46
Total		55315410	9,33,134.43

We can remove the total from visualization



Now we will also be needing for the previous week..

So that by taking out the differences we can extract the data insight (change in every week)

cture | Formatting | Properties

```
1  Previous_week_Revenue = CALCULATE(
2      SUM('public cc_detail'[Revenue]),
3      FILTER(
4          ALL('public cc_detail'),
5          'public cc_detail'[week_num2]=MAX('public cc_detail'[week_num2])-1)
6      )
```

Now the visual change will be:-

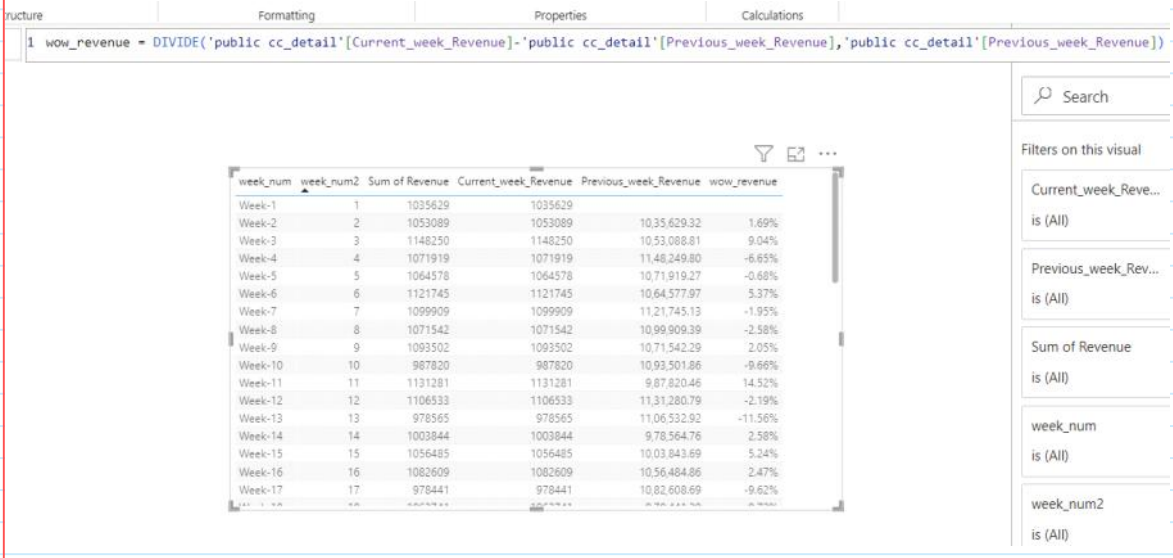
week_num	week_num2	Sum of Revenue	Current_week_Revenue	Previous_week_Revenue
Week-1	1	1035629	1035629	
Week-2	2	1053089	1053089	10,35,629.32
Week-3	3	1148250	1148250	10,53,088.81
Week-4	4	1071919	1071919	11,48,249.80
Week-5	5	1064578	1064578	10,71,919.27
Week-6	6	1121745	1121745	10,64,577.97
Week-7	7	1099909	1099909	11,21,745.13
Week-8	8	1071542	1071542	10,99,909.39
Week-9	9	1093502	1093502	10,71,542.29
Week-10	10	987820	987820	10,93,501.86
Week-11	11	1131281	1131281	9,87,820.46
Week-12	12	1106533	1106533	11,31,280.79
Week-13	13	978565	978565	11,06,532.92
Week-14	14	1003844	1003844	9,78,564.76
Week-15	15	1056485	1056485	10,03,843.69
Week-16	16	1082609	1082609	10,56,484.86
Week-17	17	978441	978441	10,82,608.69

It resulted in shifting one value..

The point of doing this is that we can watch the differece

Current value = final_value - previous_value / prev value

Why prev val- ? So that we can know the change percentage..



week_num	week_num2	Sum of Revenue	Current_week_Revenue	Previous_week_Revenue	wow_revenue
Week-1	1	1035629	1035629		
Week-2	2	1053089	1053089	10,35,629.32	1.69%
Week-3	3	1148250	1148250	10,53,088.81	9.04%
Week-4	4	1071919	1071919	11,48,249.80	-6.65%
Week-5	5	1064578	1064578	10,71,919.27	-0.68%
Week-6	6	1121745	1121745	10,64,577.97	5.37%
Week-7	7	1099909	1099909	11,21,745.13	-1.95%
Week-8	8	1071542	1071542	10,99,909.39	-2.58%
Week-9	9	1093502	1093502	10,71,542.29	2.05%
Week-10	10	987820	987820	10,93,501.86	-9.66%
Week-11	11	1131281	1131281	9,87,820.46	14.52%
Week-12	12	1106533	1106533	11,31,280.79	-2.19%
Week-13	13	978565	978565	11,06,532.92	-11.56%
Week-14	14	1003844	1003844	9,78,564.76	2.58%
Week-15	15	1056485	1056485	10,03,843.69	5.24%
Week-16	16	1082609	1082609	10,56,484.86	2.47%
Week-17	17	978441	978441	10,82,608.69	-9.62%

Now lets change the order of prev week and current week for better understanding

week_num	week_num2	Sum of Revenue	Previous_week_Revenue	Current_week_Revenue	wow_revenue
Week-1	1	1035629		1035629	
Week-2	2	1053089	10,35,629.32	1053089	1.69%
Week-3	3	1148250	10,53,088.81	1148250	9.04%
Week-4	4	1071919	11,48,249.80	1071919	-6.65%
Week-5	5	1064578	10,71,919.27	1064578	-0.68%
Week-6	6	1121745	10,64,577.97	1121745	5.37%
Week-7	7	1099909	11,21,745.13	1099909	-1.95%
Week-8	8	1071542	10,99,909.39	1071542	-2.58%
Week-9	9	1093502	10,71,542.29	1093502	2.05%
Week-10	10	987820	10,93,501.86	987820	-9.66%
Week-11	11	1131281	9,87,820.46	1131281	14.52%
Week-12	12	1106533	11,31,280.79	1106533	-2.19%
Week-13	13	978565	11,06,532.92	978565	-11.56%
Week-14	14	1003844	9,78,564.76	1003844	2.58%
Week-15	15	1056485	10,03,843.69	1056485	5.24%
Week-16	16	1082609	10,56,484.86	1082609	2.47%
Week-17	17	978441	10,82,608.69	978441	-9.62%

Now if you notice then prev val was 10325 and current 1053 so increase - 1.7% approx

And after removing the unnecessary vals like week_num and sumofrev

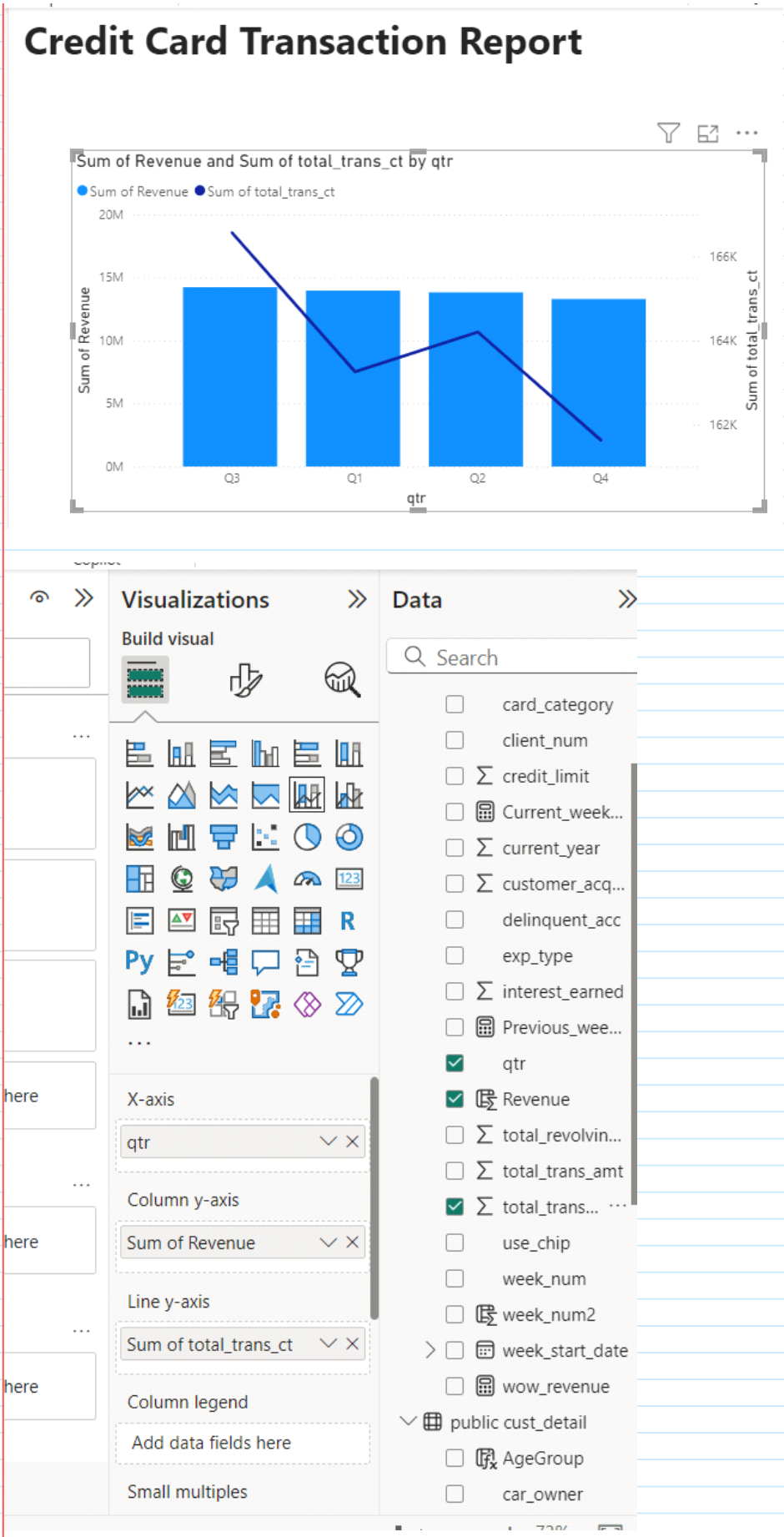
week_num2	Previous_week_Revenue	Current_week_Revenue	wow_revenue
1		1035629	
2	10,35,629.32	1053089	1.69%
3	10,53,088.81	1148250	9.04%
4	11,48,249.80	1071919	-6.65%
5	10,71,919.27	1064578	-0.68%
6	10,64,577.97	1121745	5.37%
7	11,21,745.13	1099909	-1.95%
8	10,99,909.39	1071542	-2.58%
9	10,71,542.29	1093502	2.05%
10	10,93,501.86	987820	-9.66%
11	9,87,820.46	1131281	14.52%
12	11,31,280.79	1106533	-2.19%
13	11,06,532.92	978565	-11.56%
14	9,78,564.76	1003844	2.58%
15	10,03,843.69	1056485	5.24%
16	10,56,484.86	1082609	2.47%
17	10,82,608.69	978441	-9.62%

Now next step :

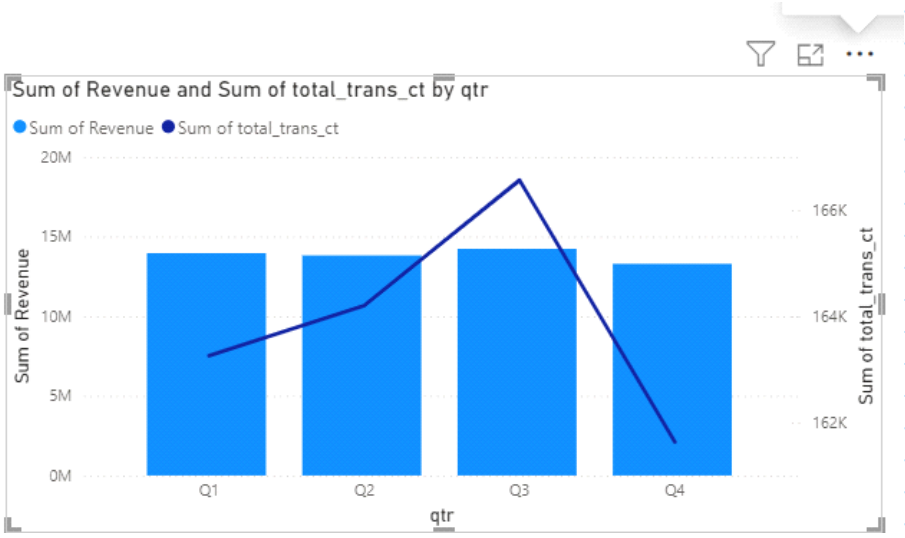
Design Dashboard and Insights

Create a new sheet and
name the dashboard
credit card transaction report increased size to the 28

After that we will make our first chart
(line and stacked chart)

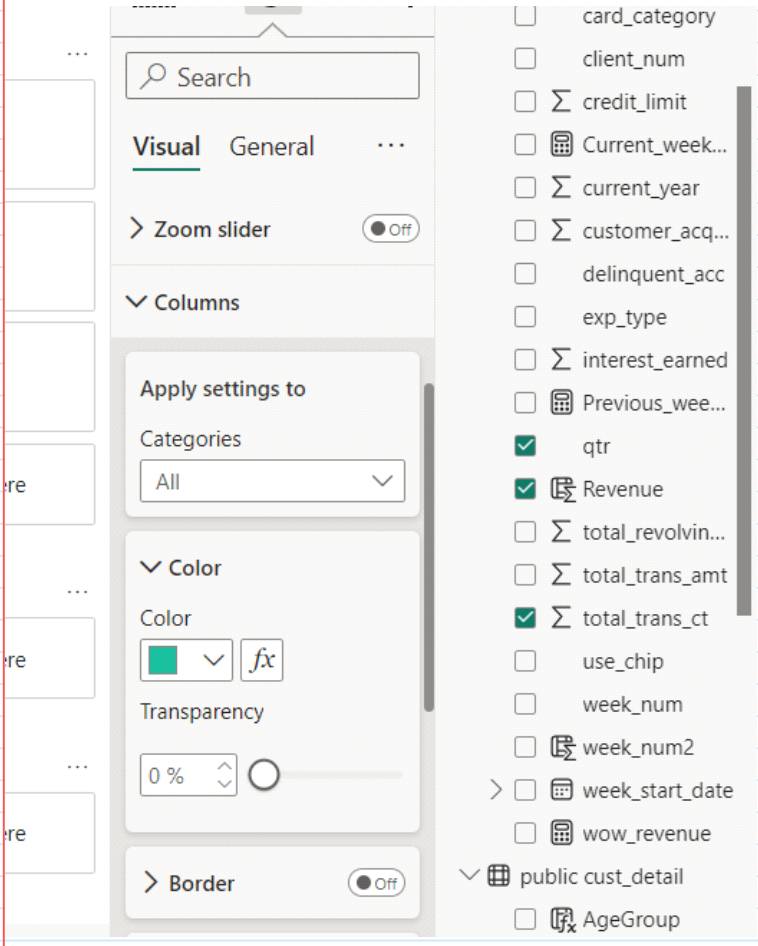


We can notice that the above qtr is not sorted we can sort it by going to the 3 dots



Lets do basic formatting

Go to the format visuals - go to columns and then change the color

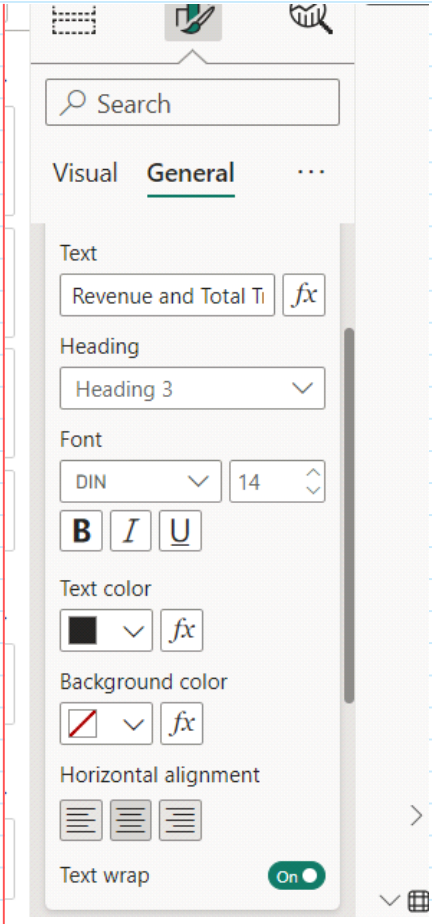


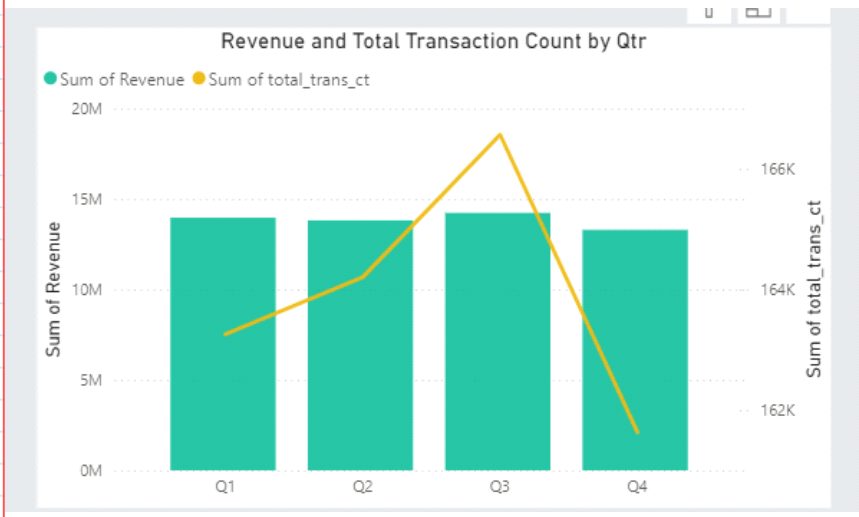
Now also change the line color

After that to remove the below label Qtr

Click on table -> x axis -> title off

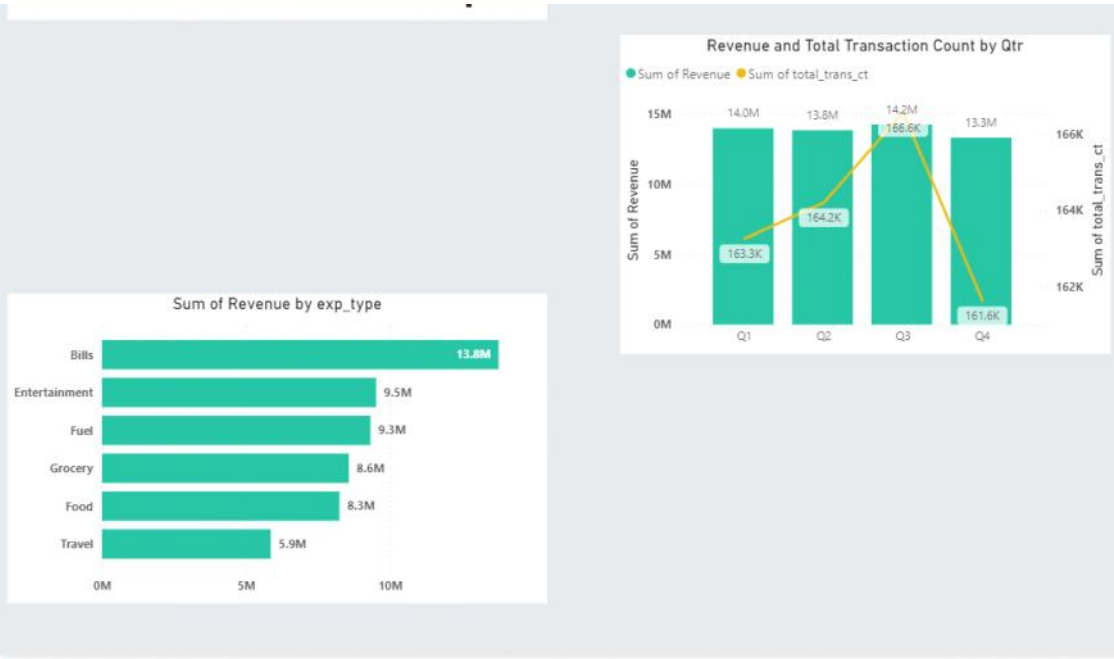
Legend is necessary

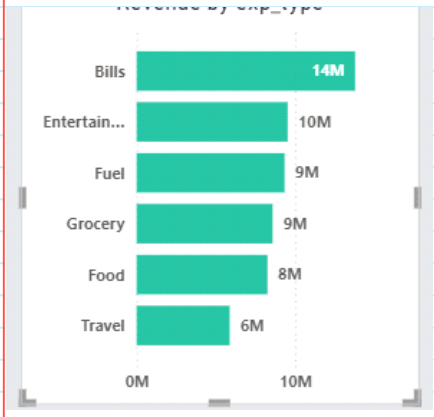




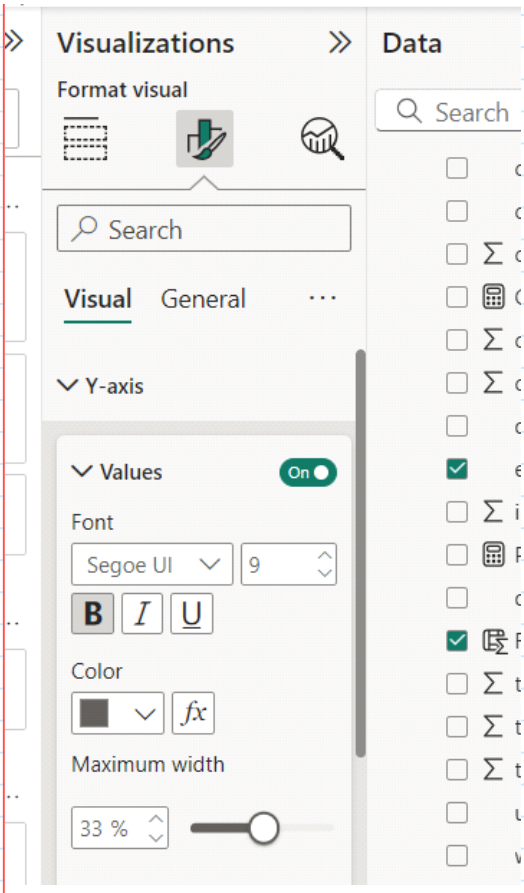
Now apply some basic formats that on - datalabels for y axis , values on etc etc

Now its easy to copy paste the format , create another bar chart and then Click on the above chart and then click on format painter and then click on the new chart. Format will be copied.



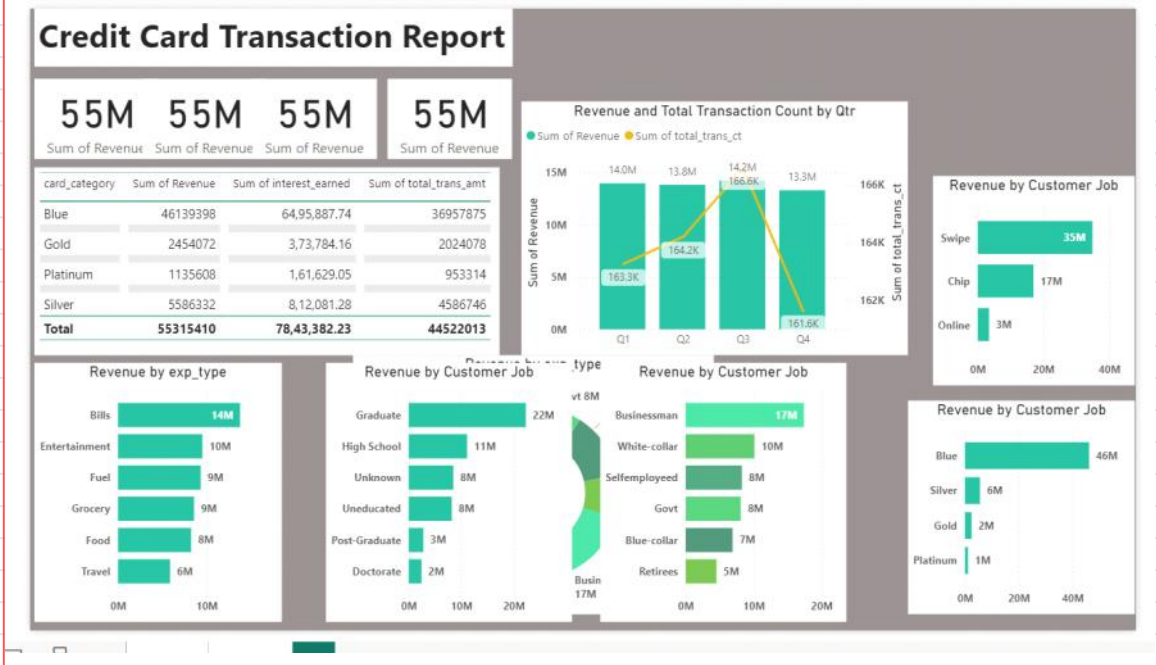


Here you can see that entertainment is not being fully exposed so for that go to y axis



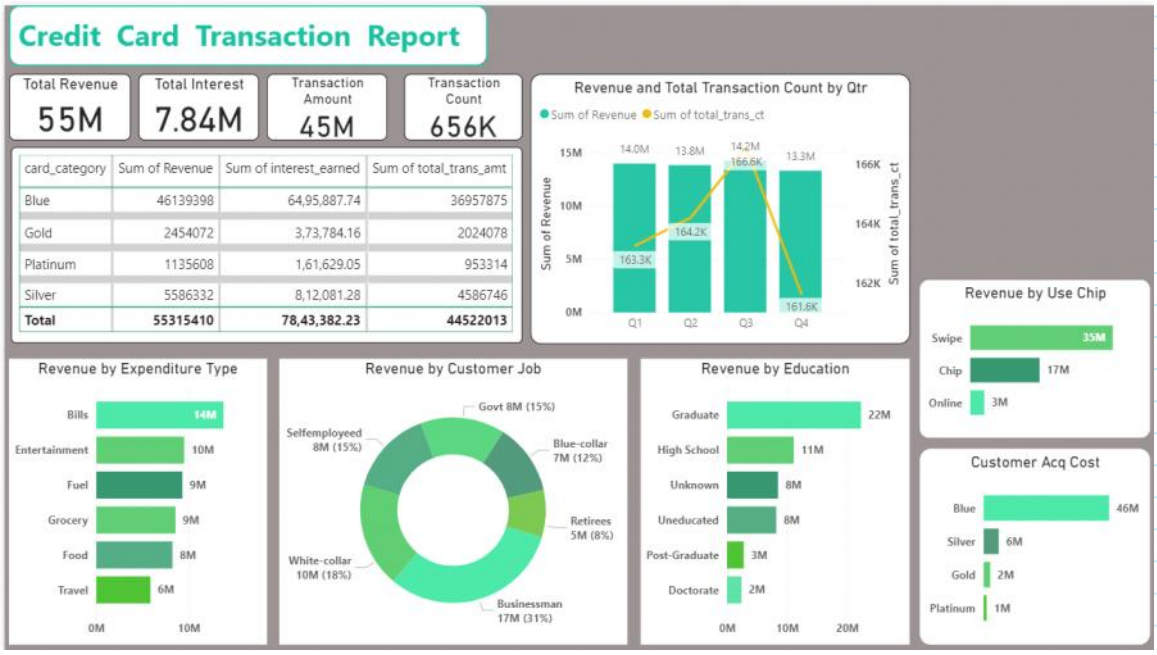
Increase the maximum width

Copy paste for others also



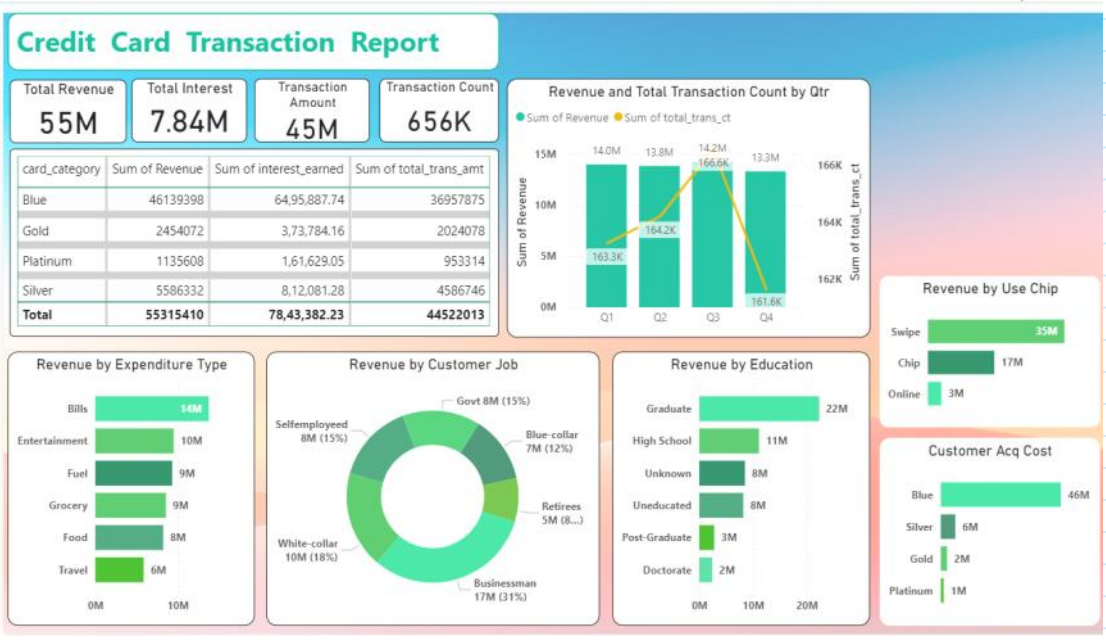
Now applying border (from format your visual-> general -> effects -> visual border)

And removing x axis value we achieve some thing like below img:

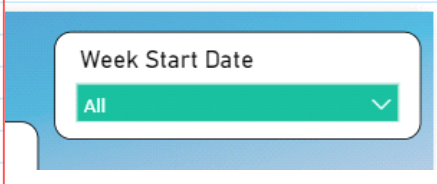


Now to change the background

Click outer region and choose image



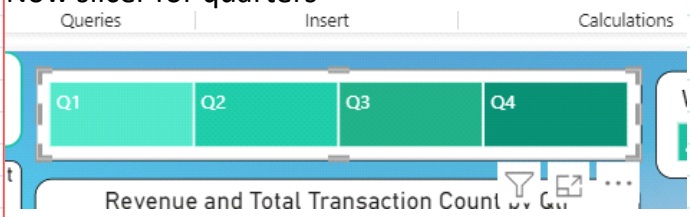
Now we will add slicer for week start date



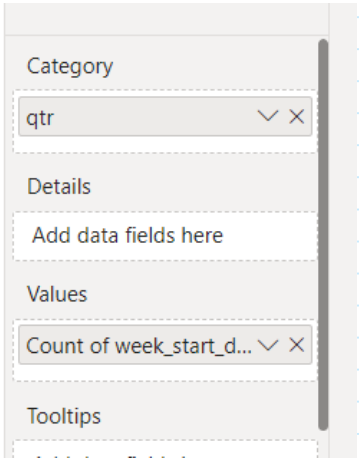
Now we will use tree map

It is a chart which takes shapes based on area

Now slicer for quarters

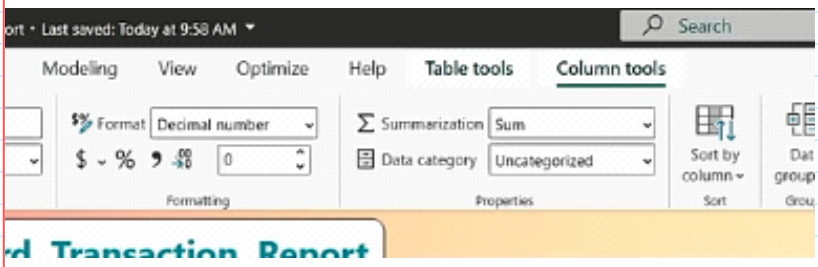


You can select multiple by holding ctrl



Upto this we have achieved

Now we want gender slicers

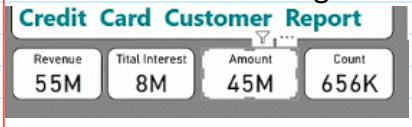


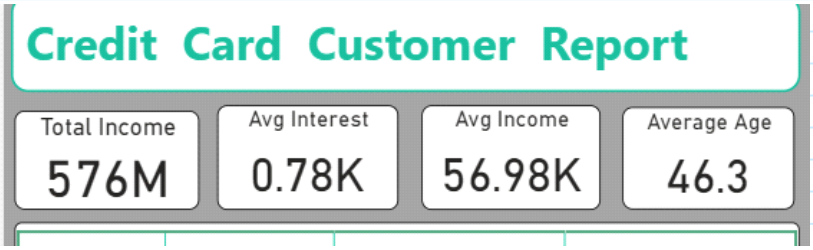
Next dashboard will be credit card customer report

Ctrl + hold and click selecting the charts..

And then ctrl + v into the new page and then sync

After that we will change the 2 cards i.e. Amount and Count

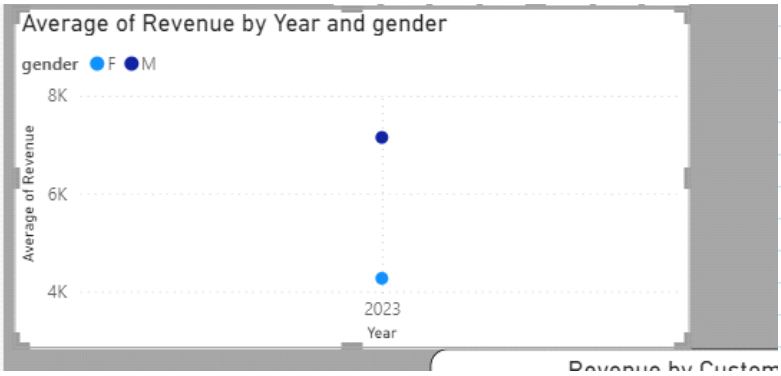


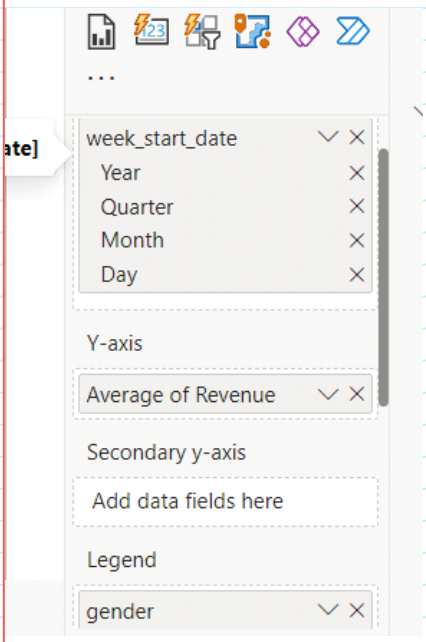


Current

First weekly report we will need a line chart.

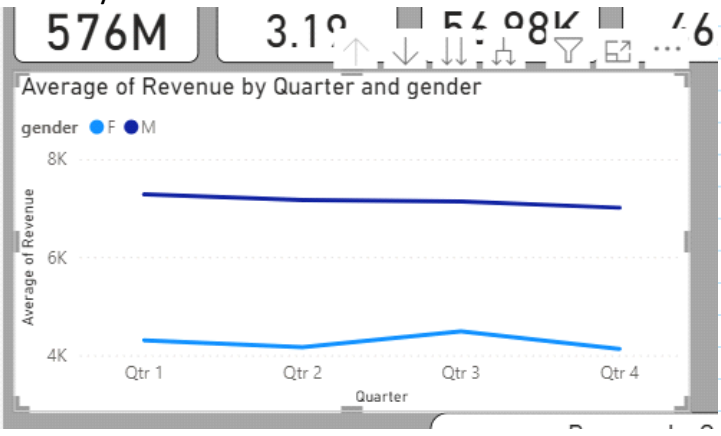
Line chart 1:





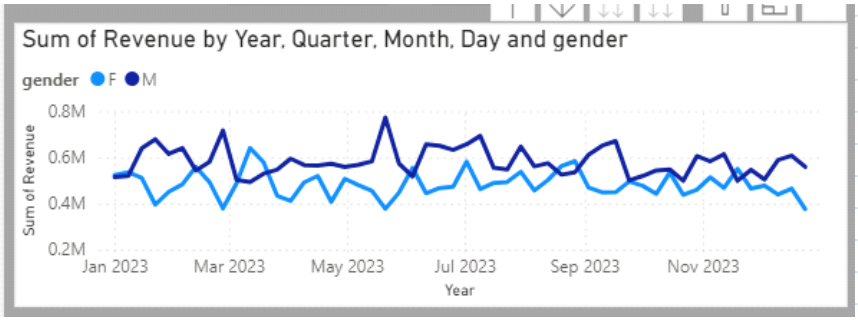
X - axis -> Year , Quarter , Month, Day

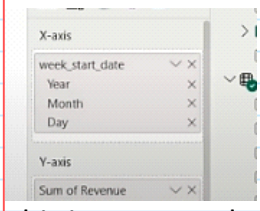
Removed year



And since we also need week no.

So we add weekno. Into the x axis



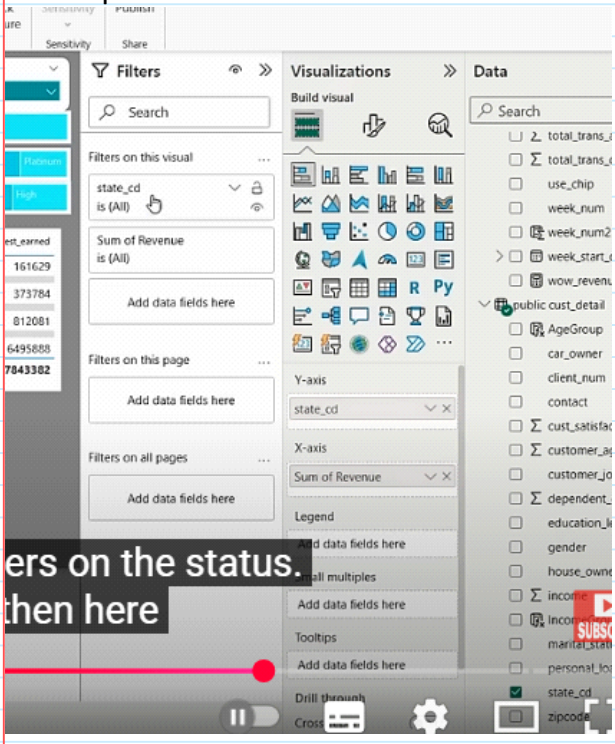


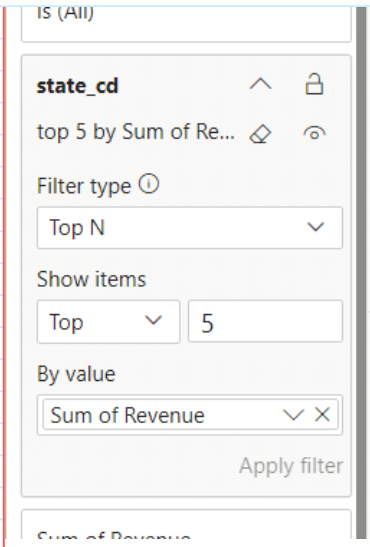
This is generated automatically since the data type is date

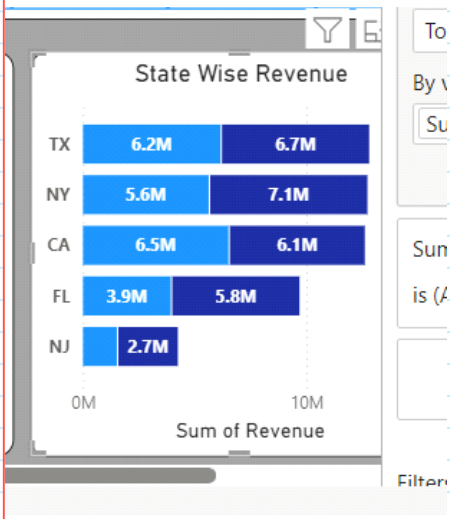
Remove title of year ,

Create a treemap for male and female

Filter options -->







Now the dashboard has been made

Now we will update the data and see how the values/ insights get change

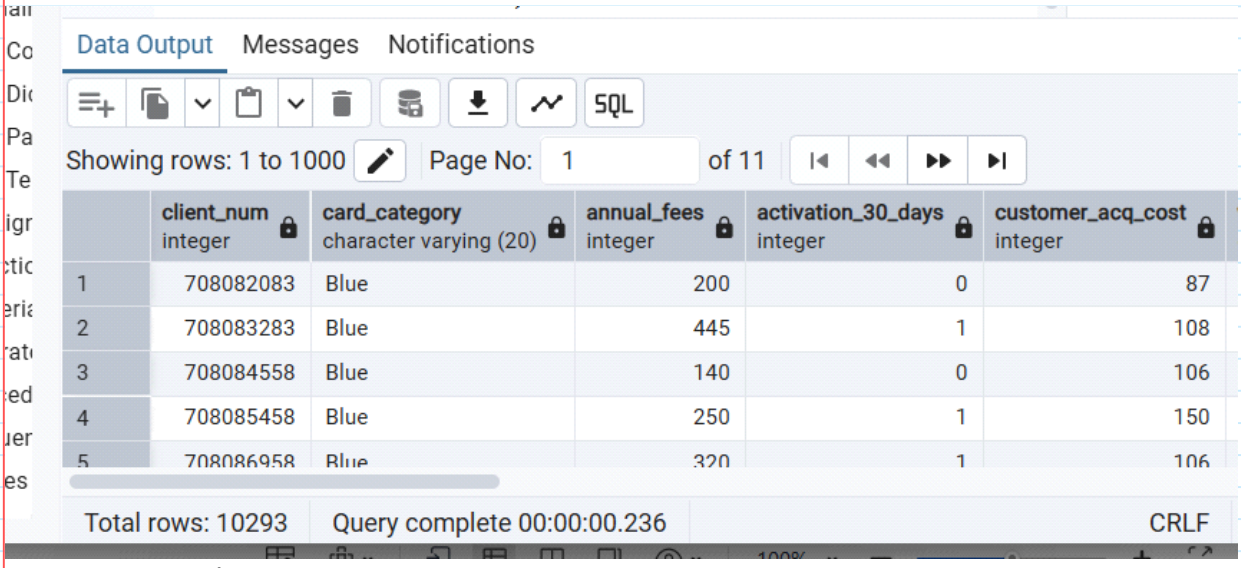
We will add the data using sql and the result will be shown in the powerbi

New csv filess - col orders should be same

We will need to copy paste the same into it just like we did before

```
4
5  COPY cc_detail
6  FROM 'D:\cc_add.csv'
7  DELIMITER ','
8  CSV HEADER;
```

And the data will be added automatically



10118 previously

Now if we move to power bi we see that the last data is 2023 dec 24
And if we click refresh then we will notice change

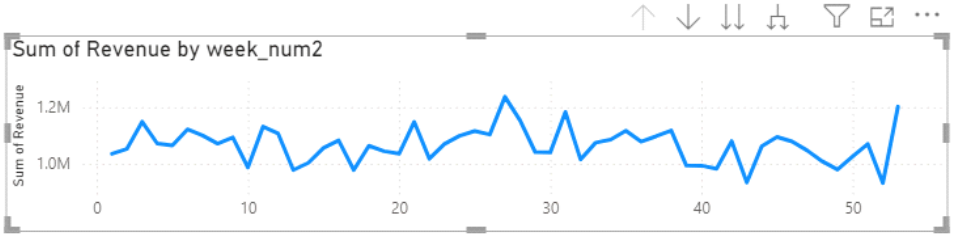
After:

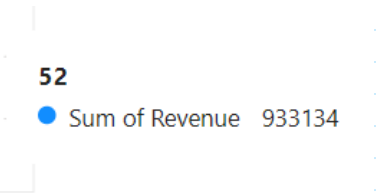
As we have updated the powerbi

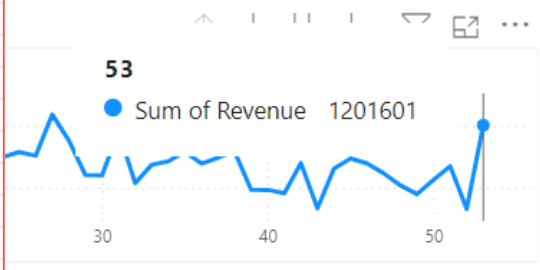
And earlier we set up the week on week revenue we can see that the data has been updated

week_num2	Previous_week_Revenue	Current_week_Revenue	wow_revenue
53	9,33,134.43	1201601	28.77%
52	10,70,439.10	933134	-12.83%
51	10,26,549.11	1070439	4.28%
50	9,80,152.37	1026549	4.73%
49	10,08,776.60	980152	-2.84%

Also we can get that week on week revenue through line chart

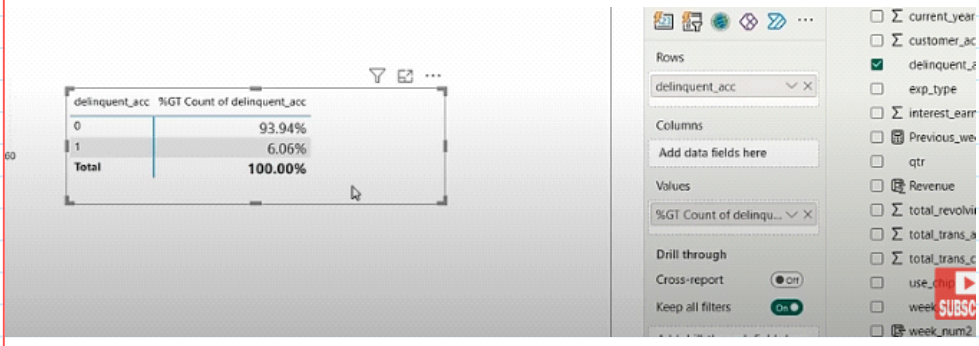


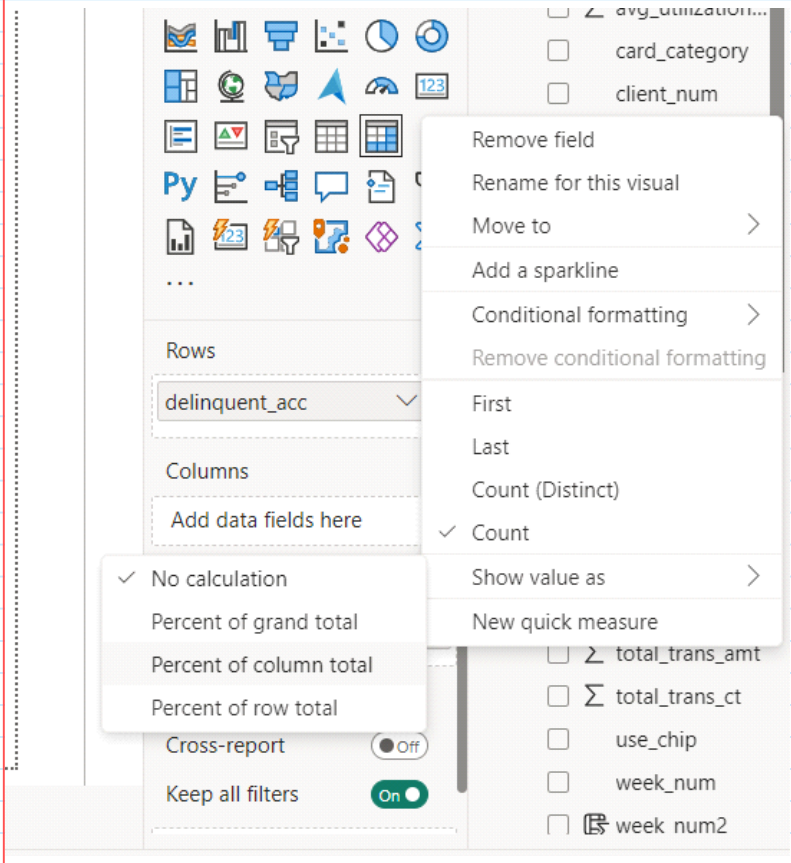




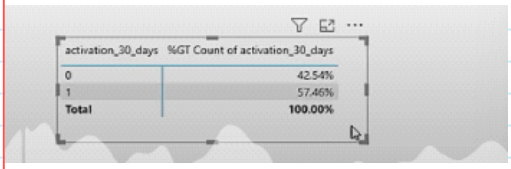
Overview YTD:

- Overall revenue is 57M
- Total interest is 8M
- Total transaction amount is 46M
- Male customers are contributing more in revenue 31M, female 26M
- Blue & Silver credit card are contributing to 93% of overall transactions
- TX, NY & CA is contributing to 68%
- Overall Activation rate is 57.5%
- Overall Delinquent rate is 6.06%





Within 30 days how many client activated their card how many didn't



delinquent_acc	Blue-collar	Businessman	Govt	Retirees	Selfemployeed	White-collar	**Total**
0	14.75%	17.81%	14.00%	9.16%	23.87%	14.35%	**93.94%**
1	0.85%	0.99%	1.10%	0.61%	1.66%	0.85%	**6.06%**
Total	**15.60%**	**18.80%**	**15.10%**	**9.77%**	**25.53%**	**15.19%**	**100.00%**

Added the jobs in col section

And we notice that self employed has most delinquent rate