Data Visualization in Power BI

-By Sangini Phansekar

Project Name:-

CREDIT CARD FINANCIAL DASHBOARD

Step1:-

Project Overview

To develop a comprehensive Credit Card Weekly Dashboard that provides real time insights into key performance metrics and trends, enabling stakeholders to monitor and analyze credit card operations effectively. (Here I am tracking weekly performance as per data given)

Step2:-

Import data to SQL database

1. Prepare CSV file :- Here I used two CSV files one is credit_card.csv with 18 columns, 10108 rows and second one customer.csv with 15 columns and 10108 rows.
2. Create tables in SQL:-

```
-- SQL Query to create and import data from csv files:

-- 0. Create a database
CREATE DATABASE ccdb;

-- 1. Create cc_detail table

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),
```

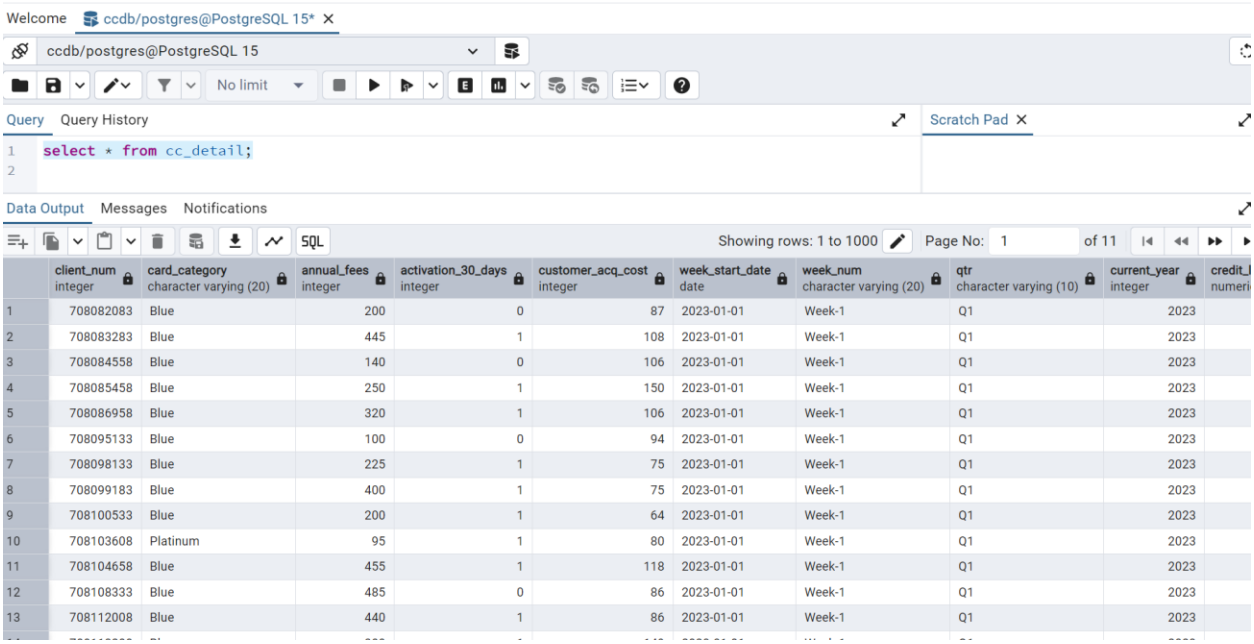
```
    Exp_Type VARCHAR(50),
    Interest_Earned DECIMAL(10,3),
    Delinquent_Acc VARCHAR(5)
);
```

-- 2. Create cc_detail table

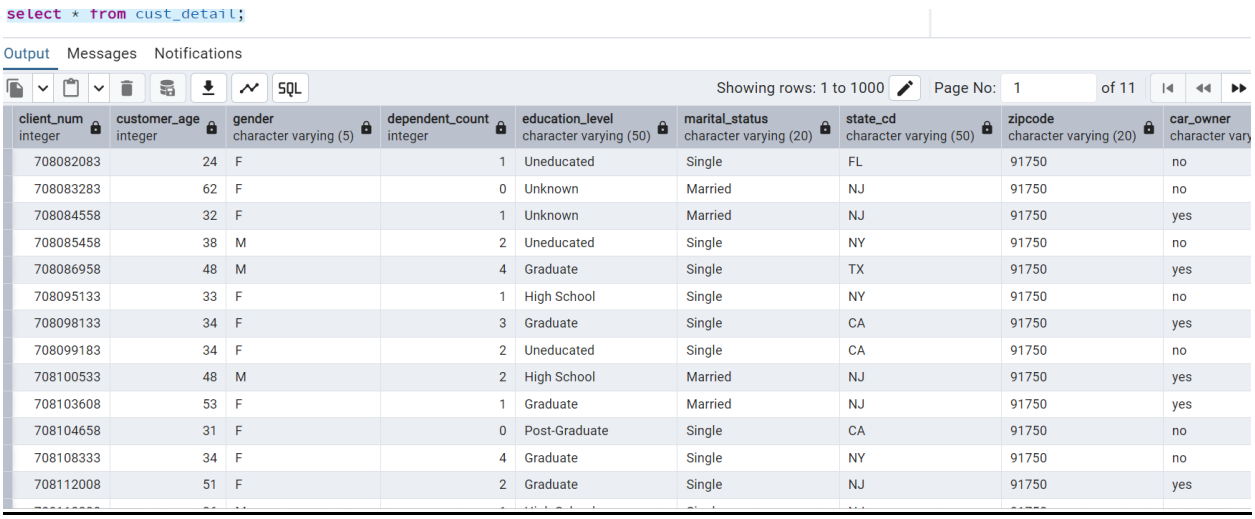
```
CREATE TABLE cust_detail (
    Client_Num INT,
    Customer_Age INT,
    Gender VARCHAR(5),
    Dependent_Count INT,
    Education_Level VARCHAR(50),
    Marital_Status VARCHAR(20),
    State_cd VARCHAR(50),
    Zipcode VARCHAR(20),
    Car_Owner VARCHAR(5),
    House_Owner VARCHAR(5),
    Personal_Loan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

-- 3. Copy csv data into SQL

```
-- copy cc_detail table

COPY cc_detail
FROM 'D:\credit_card.csv'
DELIMITER ','
CSV HEADER;


-- copy cust_detail table

COPY cust_detail
FROM 'D:\customer.csv'
DELIMITER ','
CSV HEADER;
```



	client_num integer	card_category character varying (20)	annual_fees integer	activation_30_days integer	customer_acq_cost integer	week_start_date date	week_num character varying (20)	qtr character varying (10)	current_year integer	credit_l numeri
1	708082083	Blue	200	0	87	2023-01-01	Week-1	Q1	2023	
2	708083283	Blue	445	1	108	2023-01-01	Week-1	Q1	2023	
3	708084558	Blue	140	0	106	2023-01-01	Week-1	Q1	2023	
4	708085458	Blue	250	1	150	2023-01-01	Week-1	Q1	2023	
5	708086958	Blue	320	1	106	2023-01-01	Week-1	Q1	2023	
6	708095133	Blue	100	0	94	2023-01-01	Week-1	Q1	2023	
7	708098133	Blue	225	1	75	2023-01-01	Week-1	Q1	2023	
8	708099183	Blue	400	1	75	2023-01-01	Week-1	Q1	2023	
9	708100533	Blue	200	1	64	2023-01-01	Week-1	Q1	2023	
10	708103608	Platinum	95	1	80	2023-01-01	Week-1	Q1	2023	
11	708104658	Blue	455	1	118	2023-01-01	Week-1	Q1	2023	
12	708108333	Blue	485	0	86	2023-01-01	Week-1	Q1	2023	
13	708112008	Blue	440	1	86	2023-01-01	Week-1	Q1	2023	



client_num integer	customer_age integer	gender character varying (5)	dependent_count integer	education_level character varying (50)	marital_status character varying (20)	state_cd character varying (50)	zipcode character varying (20)	car_owner character vary
708082083	24	F	1	Uneducated	Single	FL	91750	no
708083283	62	F	0	Unknown	Married	NJ	91750	no
708084558	32	F	1	Unknown	Married	NJ	91750	yes
708085458	38	M	2	Uneducated	Single	NY	91750	no
708086958	48	M	4	Graduate	Single	TX	91750	yes
708095133	33	F	1	High School	Single	NY	91750	no
708098133	34	F	3	Graduate	Single	CA	91750	yes
708099183	34	F	2	Uneducated	Single	CA	91750	no
708100533	48	M	2	High School	Married	NJ	91750	yes
708103608	53	F	1	Graduate	Married	NJ	91750	yes
708104658	31	F	0	Post-Graduate	Single	CA	91750	no
708108333	34	F	4	Graduate	Single	NY	91750	no
708112008	51	F	2	Graduate	Single	NJ	91750	yes

Step 3: Importing data from SQL database to Power BI

1. Get Data from Postegres SQL and Save

Step 4: Data Processing & DAX Queries

Applied some queries to make some columns

Queries applied on Credit Card transactions table

1. Revenue = 'public cc_detail'[annual_fees] + 'public cc_detail'[total_trans_amt] + 'public cc_detail'[interest_earned]

2. Current_week_Revenue = CALCULATE(

SUM('public cc_detail'[Revenue]),

FILTER(

ALL('public cc_detail'),

'public cc_detail'[week_num2] = MAX('public cc_detail'[week_num2])))

3. week_num2 = WEEKNUM('public cc_detail'[week_start_date])

4. wow_revenue = DIVIDE(([Current_week_Revenue]-[Previous_week_Revenue]),[Previous_week_Revenue])

Queries Applied on Customer table

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",

'public cust_detail'[customer_age] >= 40 && 'public cust_detail'[customer_age] < 50 ,"40-50",

'public cust_detail'[customer_age] >= 50 && 'public cust_detail'[customer_age] < 60 ,"50-60",

'public cust_detail'[customer_age] >= 60,"60+",

"unkown"

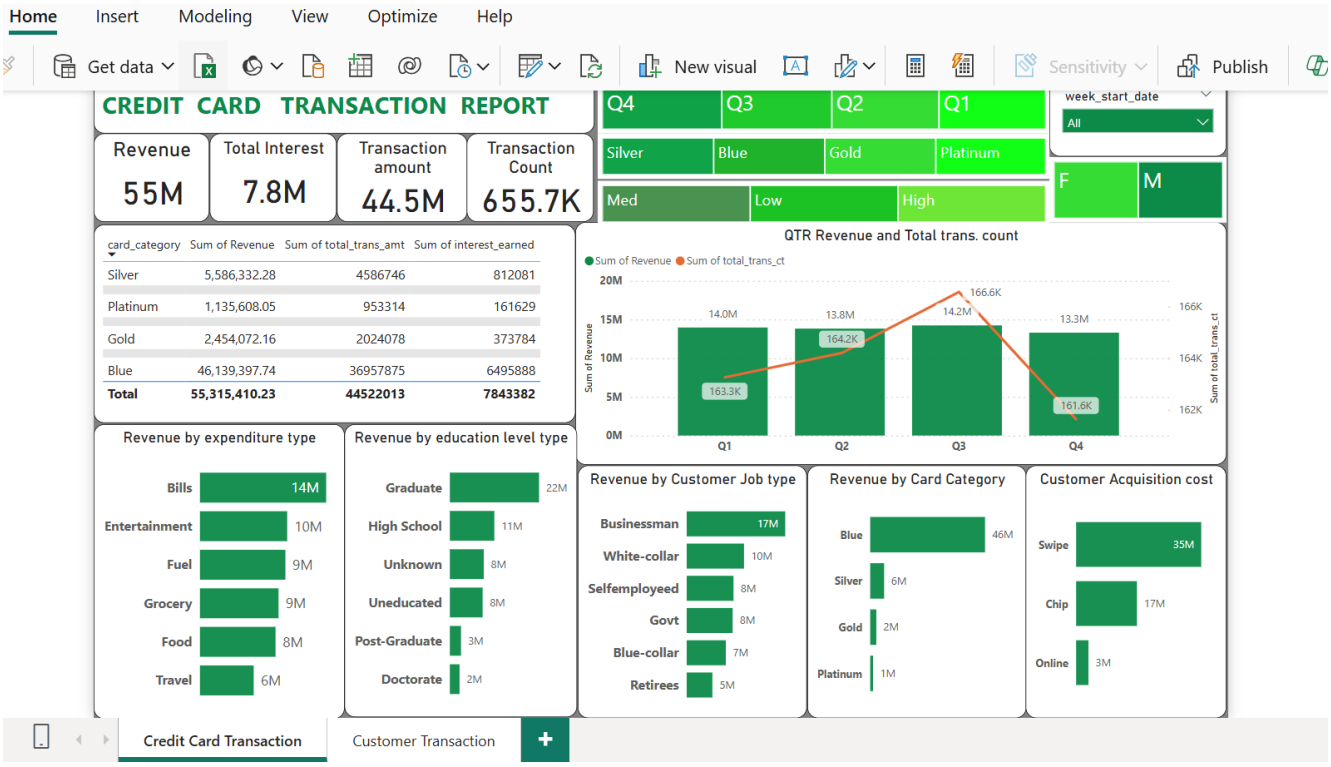
)

```
IncomeGroup = switch(
    True(),
'public cust_detail'[income] < 35000, "Low",
```

```
        'public cust_detail'[income] >=35000 && 'public cust_detail'[income] < 70000
    ,"Med",
        'public cust_detail'[income] >= 70000, "High",
     "unkown" )
```

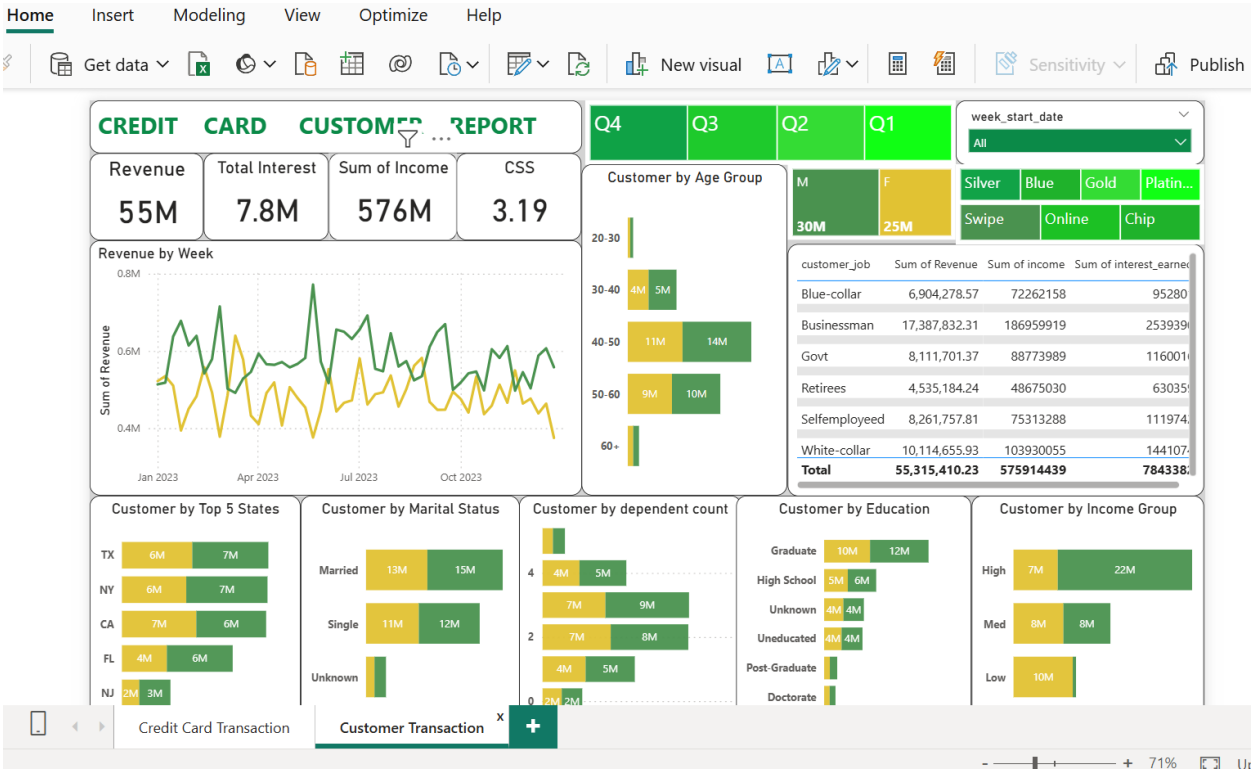
DASHBOARD DESIGNING: -

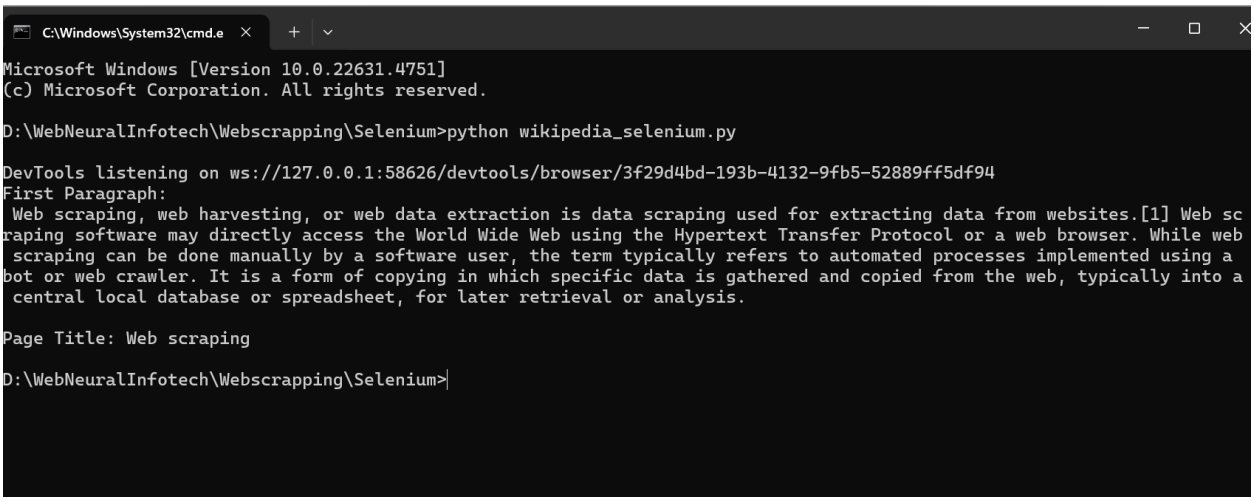
1. Slicer for weekly filters.
2. Cards to show KPI like Revenue, Total Transaction amount etc.
3. Line & Stacked column chart – Revenue & Total transation
4. Stacked Bar Chart- Revenue by Expenditure type, by Education level type, by Customer Job type, by Card Category type
5. Table – Card Type, Revenue, Total Interest, Total Transaction amount
6. Treemaps for QTR filter, CARD filter, gender filter and Income filter



1. Slicer for weekly filters.
2. Cards to show KPI like Revenue, Total Interest etc.
3. Line & Stacked column chart – revenue by weel and Sum of Revenue
4. Stacked Bar Chart- Customer by top 5 states, Marital Status by education etc.
5. Table – Customer Job, Sum of Interest etc.
6. Treemaps for QTR filter, Card filter etc.

In this Dashboard I have used treemap for Revenue by gender and align on every chart alignment.




C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.
D:\WebNeuralInfotech\Webscrapping\Selenium>python wikipedia_selenium.py
DevTools listening on ws://127.0.0.1:58626/devtools/browser/3f29d4bd-193b-4132-9fb5-52889ff5df94
First Paragraph:
Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites.[1] Web scraping software may directly access the World Wide Web using the Hypertext Transfer Protocol or a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.
Page Title: Web scraping
D:\WebNeuralInfotech\Webscrapping\Selenium>