1) Import Data from SQL Database

- Gathered two files related to the UseCase -

1)Customer credit-card details

2) Customer Details

2) Data Processing and DAX

- As the data is clean so we don't need data processing in it.

- We are going to use DAX Query to add more features to our data for better visualization and results.

A] Columns

1) Age\_Group: Make the group of ages for better visualization and results. Age\_Group ranging from 20-30,30-40,40-50,50,60,60+

Example: Age\_Group = SWITCH(TRUE(),'credit cust\_detail'[Customer\_Age]<=30,"20-30",

'credit cust\_detail'[Customer\_Age]>=30 && 'credit cust\_detail'[Customer\_Age]<40,"30-40",

'credit cust\_detail'[Customer\_Age]>=40 && 'credit cust\_detail'[Customer\_Age]<50,"40-50",

'credit cust\_detail'[Customer\_Age]>=50 && 'credit cust\_detail'[Customer\_Age]<60,"50-60",

'credit cust\_detail'[Customer\_Age]>=60,"60+",

"UNKNOWN")

2) Income\_Group: Make the group of Incomes for better visualization and results. Income\_Group ranging from 35000,35000-70000,70000

Example: Income\_Group = SWITCH(

TRUE(),

'credit cust\_detail'[Income]<35000,"Low",

'credit cust\_detail'[Income]>=35000 && 'credit cust\_detail'[Income]<70000,"Med",

'credit cust\_detail'[Income]>=70000,"High",

"UNKNOWN"

)

3) Revenue: Revenue can differ based on different criterias/

Bussiness understanding. To calculate the revenue following columns sum are considered:

1) Annual Fees 2) Total Transaction Amount 3) Interest Rate

Example: Revenue = 'credit cc\_details'[Annual\_Fees]+'credit cc\_details'[Total\_Trans\_Amt]+'credit cc\_details'[Interest\_Earned]

4) Week Num2: New column which will provide the week number which will be sorted properly. And it will help to track week wise revenue. We will create this column using function WEEKNUM()

Example: week\_num2 = WEEKNUM('credit cc\_details'[Weeks\_start\_date])

B] Measures

1) Current\_Week\_Revenue:Computes the total revenue for the current week, ignoring any existing filters on the table.

Steps:

-SUM('credit cc\_details'[Revenue]): Adds up the revenue values from the Revenue column.

-FILTER(ALL('credit cc\_details'), 'credit cc\_details'[week\_num2] = MAX('credit cc\_details'[week\_num2])):

\*ALL('credit cc\_details'): Removes existing filters from the credit cc\_details table.

\*FILTER: Filters the data to include only rows where week\_num2 matches the maximum week\_num2 value.

-MAX('credit cc\_details'[week\_num2]): Finds the maximum value of week\_num2, representing the current week.

2) Previous\_Week\_Revenue: Computes the sum of the Revenue for the week immediately before the current week.

Steps:

- CALCULATE: Adjusts the context to compute the total revenue.

- SUM('credit cc\_details'[Revenue]): Adds up the Revenue values.

- FILTER: Applies a filter to consider all rows where week\_num2 equals the previous week.

- ALL('credit cc\_details'): Ignores existing filters on the table.

- MAX('credit cc\_details'[week\_num2]) - 1: Determines the previous week number by subtracting 1 from the maximum week number in the current context.

3)Week\_on\_week\_revenue: Calculates the week-over-week (WoW) percentage change in revenue.

Steps:

-Current\_Week\_Revenue - Previous\_Week\_Revenue: Computes the difference in revenue between the current week and the previous week.

-DIVIDE(...): Safely divides the difference by the previous week's revenue to get the percentage change, handling division by zero.