Internship Project Report

Financial Year Data in Cement Factory Data

Company: Celebal Technologies

Role: Power BI Intern

Branch: B.Tech CSE with Spl. in Business Analytics and Optimization and B.Tech CSE with Spl. in E- Commerce, Retail and automation

Team Members:

Student's Name	SAP ID
Monika Kushwaha	500068753
Indrakshee Roy Choudhury	500069104
Rima Giri	500069082
Vinayak Jain	500067449

Faculty Mentor:

Dr. Hitesh Kumar Sharma

Assistant Professor

Department of Informatics

Team Lead (Celebal Technologies):

Pulkit Pareek

ABSTRACT

We have created Power BI dashboards showing the visualizations created from the given dataset We have performed data cleaning and further used DAX commands to create new measures using which we have plotted different visualizations as per the given project Description.

Project Description:

Problem statements
1. Total Quantity Sold
2. Quantity Sold Last Month
3. Quantity Sold Last Year
4. Average Quantity Sold per Day
5. Quantity Sold in "Germany” and "Dell" & "HP"
6. Quantity Sold Two Months Ago
7. Quantity Sold Two Quarters Ago
8. Last Month Profit
9. Profit 2 Month Ago
10. Last Month Cumulative Profit
11. Cumulative Profit 2 Month Ago
12. Total Profit running total in Date
13. Apply Currency Conversion in it
14. How to apply the Fiscal Year

Contents

INTRODUCTION

Microsoft PowerBI:

Connect to and visualize any data using the unified, scalable platform for self-service and enterprise business intelligence (BI) that’s easy to use and helps you gain deeper data insight. Power BI helps to bridge the gap between data and decision making.

Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. Your data may be an Excel spreadsheet, or a collection of cloud-based and on-premises hybrid data warehouses. Power BI lets you easily connect to your data sources, visualize, and discover what's important, and share that with anyone or everyone you want.

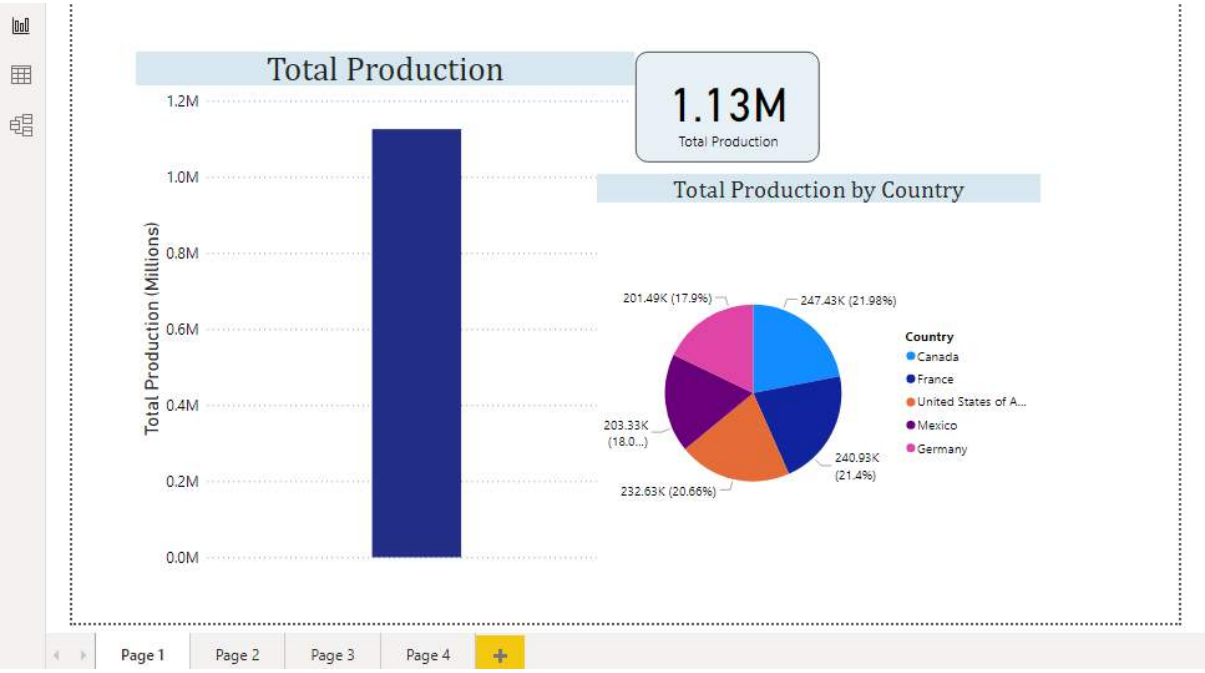
Power BI consists of several elements that all work together, starting with these three basics:

- A Windows desktop application called Power BI Desktop.
- An online SaaS (Software as a Service) service called the Power BI service.
- Power BI mobile apps for Windows, iOS, and Android devices.

Power BI Desktop is a free application you install on your local computer that lets you connect to, transform, and visualize your data. With Power BI Desktop, you can connect to multiple different sources of data, and combine them (often called modelling) into a data model. This data model lets you build visuals, and collections of visuals you can share as reports, with other people inside your organization. Most users who work on business intelligence projects use Power BI Desktop to create reports, and then use the Power BI service to share their reports with others.

Report No. 1

Task - **Total Quantity Sold**



Description-

This visualization shows the total quantity sold alongside the total quantity sold country wise in a financial year by the cement company.

Here, we have used three charts: Clustered column chart for Total Production, Pie chart for Quantity sold country wise and a score card to show the details of the same.

In the pie chart, five different colours have been used to represent different countries and a legend for the same is also shown.

DAX commands used:

Total Production = CALCULATE (SUM (financials [Units Sold]))

We have used the function **SUM** here, which adds all the numbers in a column.

For Total production country wise we have just used the two measures ‘Country’ and ‘Total Production’ from the financial table.

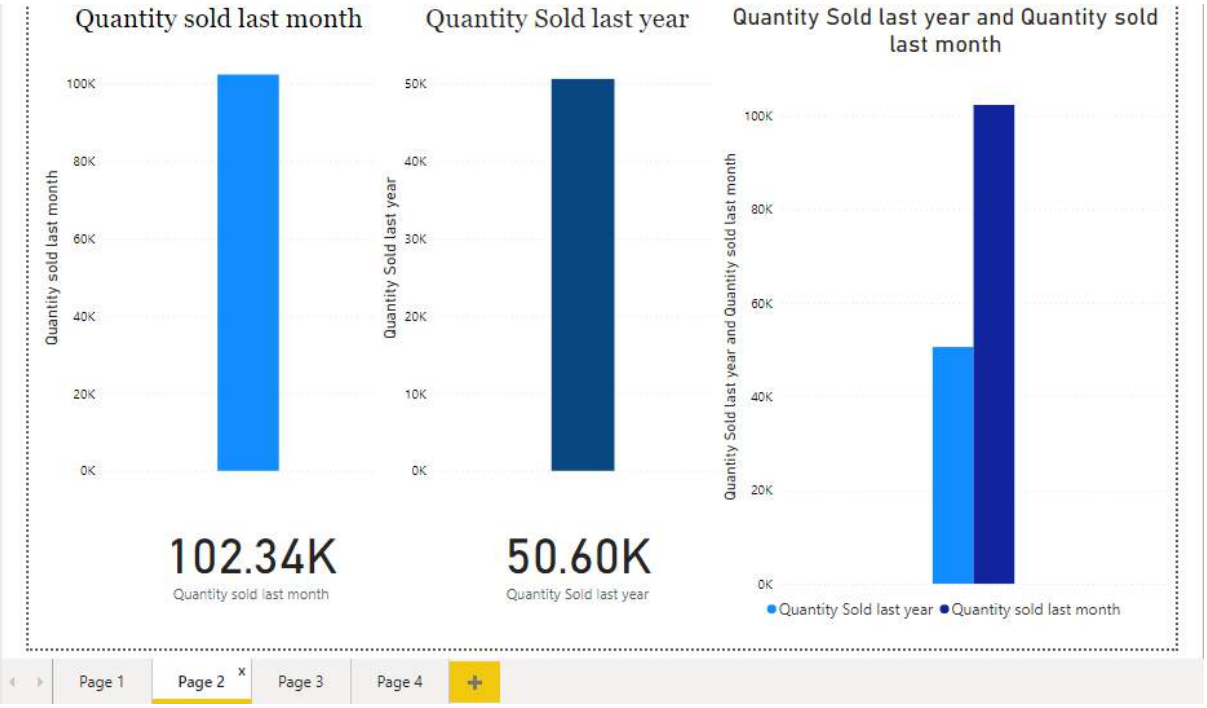
Result-

Total Quantity = **1,125,806.00**

Report No. 2

Task-

1. **Quantity Sold Last Month**
2. **Quantity Sold Last Year**



Description-

This visualization shows the total quantity sold last month and the total quantity sold last year alongside the comparison of both the measures from the financial year data of the cement company.

Here, we have used three clustered bar charts and two score cards to show the visualization in a better way. We can see that in the comparison chart we have used two different colours to represent the quantity sold last month and the quantity sold last year, and it tells us that the quantity sold last month was comparatively way higher than the total quantity sold in the previous year. This shows that the cement company has had a tremendous growth in case of sales from last year till now.

DAX commands used-

Quantity sold last month = CALCULATE (SUM (financials [Units Sold]), DATESINPERIOD (financials [Date], MAX (financials [Date]), -1, MONTH))

Quantity Sold last year = CALCULATE (SUM (financials [Units Sold]), DATESINPERIOD (financials [Date], MAX (financials [Year]), -1, YEAR))

Here, we have used three functions,

- **SUM:** This function adds all the numbers in a column.

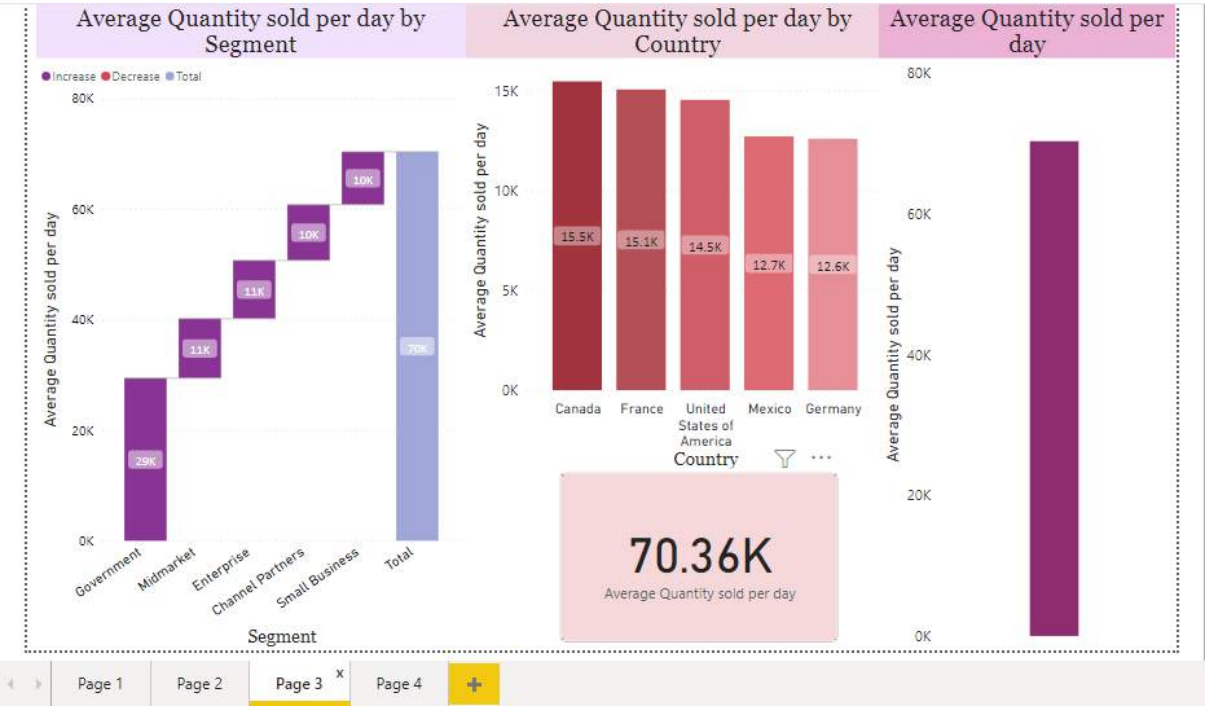
- **DATESINPERIOD**: This function returns a table that contains a column of dates that begins with a specified start date and continues for the specified number and type of date intervals.
- **MAX:** This function returns the largest value in a column, or between two scalar expressions.

Result-

Quantity sold last month = **102,336.00**
Quantity sold last month = **50,601.00**

Report No. 3

Task- **Average Quantity Sold per Day**



Description-

The above visualization shows the average quantity sold per day in total alongside the average quantity sold per day with respect to segment and country from the financial data of the cement company.

Here, we have used three different charts for better understanding

- Waterfall chart to show the average sales per day with respect to different segments the cement company deals in.
- Stacked column chart to show the average sales per day with respect to different countries in which the cement company functions.
- Clustered column chart for average sales per day in total.

We see that country wise Canada has the maximum average sales per day followed by France in the second place while segment wise Government sector has the most average sales in a single day. We have also used different colours for improved understanding. In the waterfall chart for segment wise average sales per day we have denoted increase by purple, decrease by red and total by light blue. Similarly, in country wise Average sales per day we have used a gradient colour method to denote the maximum to minimum sales.

DAX commands used-

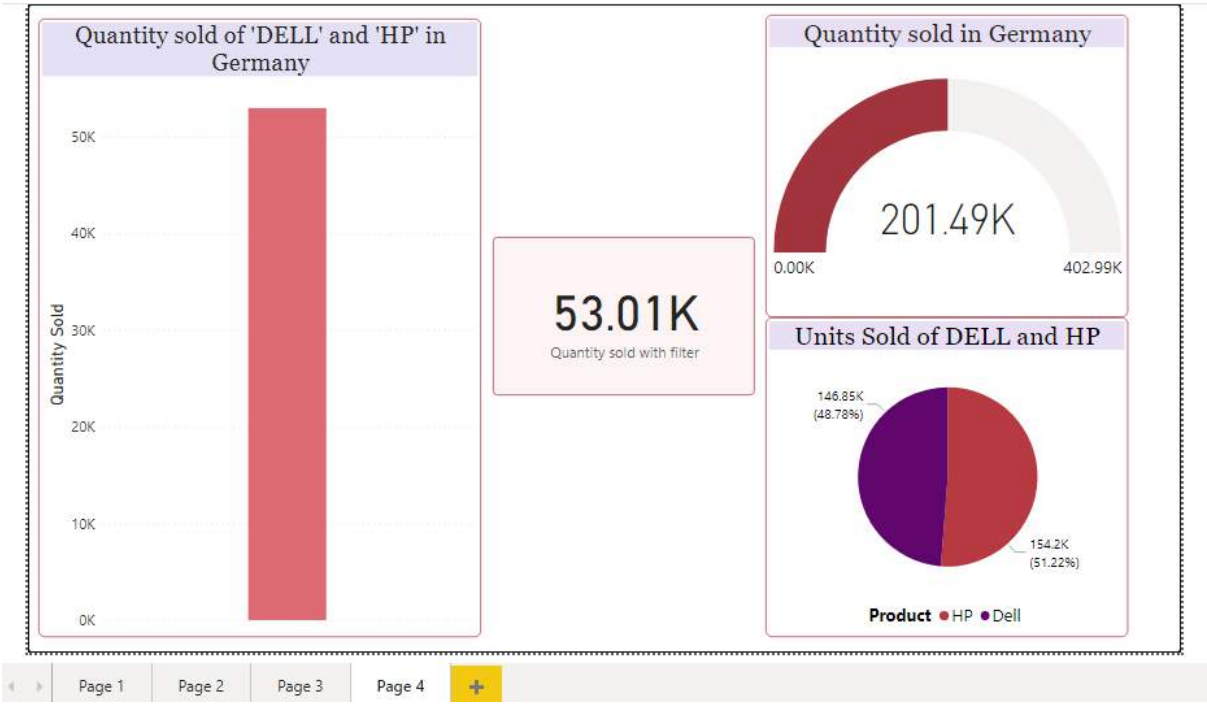
Average Quantity sold per day = AVERAGEX (VALUES (financials [Date]), financials [Total Production])

We have used the **AVERAGEX** function here, which calculates the average (arithmetic mean) of a set of expressions evaluated over a table.

Result- Average Quantity sold per day = **70,362.88**

Report No. 4

Task- **Quantity Sold in "Germany" and "Dell" & "HP"**



Description-

The above visualization shows the quantity of units sold in Germany of 'Dell' and 'HP' by the cement company.

Here, we have used three different charts for better understanding

- Stacked column chart to show the units sold in Germany of 'Dell' and 'HP'.
- A Gauge chart to show the quantity sold in the country 'Germany'
- A Pie chart to show the number of units sold of 'Dell' and 'HP'

We have also used different colours in the pie chart to represent the units sold of dell and units sold of HP respectively and a legend for the same is shown in the above visualization. The visualization also gives out a very important information, that is, from the Gauge chart it is evident that out of the total sales, Germany comprises of almost half the total sales of the cement company.

DAX commands used-

Quantity sold with filter = CALCULATE (SUM ('financials'[Units Sold]),'financials'[Country] IN {"Germany"}, financials [Product] IN {"Dell","HP"})

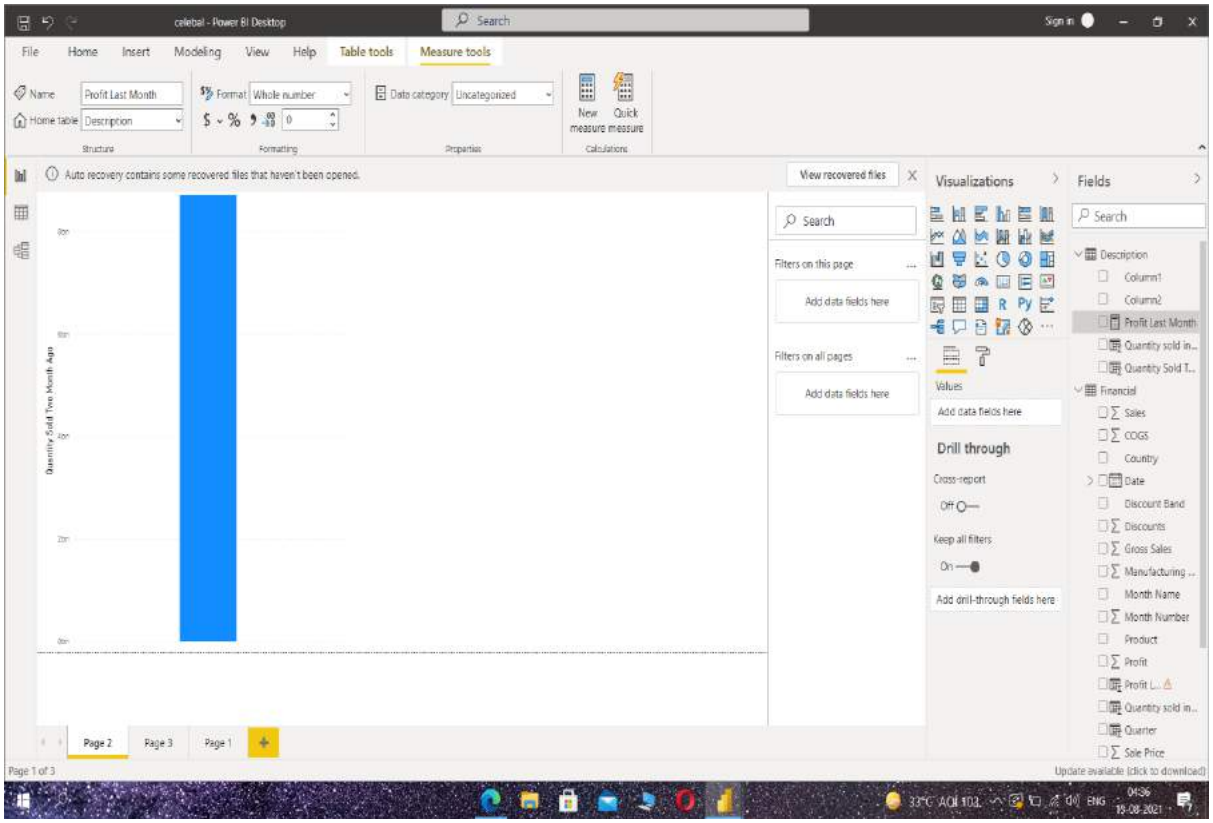
Quantity sold in Germany = CALCULATE (SUM ('financials'[Units Sold]),'financials'[Country] IN {"Germany"})

Result-

Quantity Sold in "Germany" and "Dell" & "HP" = **53,005.00**

Report No. 5

Task- **Quantity Sold Two Months Ago**



Description-

-This data shows Total Quantity sold two months ago in a cement factory. DAX formula used here-

DAX commands used-

Quantity Sold Two Month Ago = CALCULATE (SUM (Financial [Total_quan_sold]), PARALLELPERIOD('Financial'[Date],-2,MONTH))

-Where, total quantity sold (Total_quan_sold) is calculated by given DAX formula-

Total_quan_sold = SUM (Financial [Units Sold])

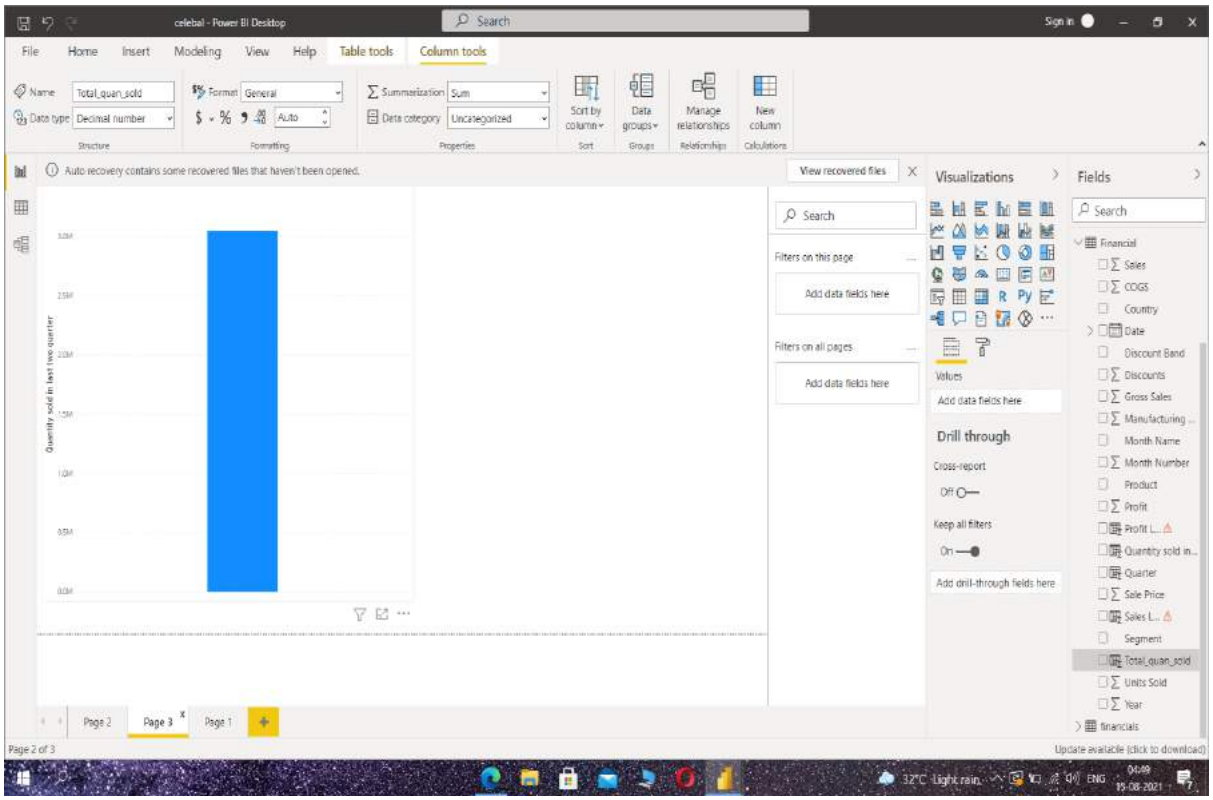
-Here, we've used the **PARALLELPERIOD** function which returns a table that contains a column of dates that represents a period parallel to the dates in the specified dates column, in the current context, with the dates shifted a number of intervals either forward in time or back in time.

Result-

Total Quantity sold two months ago= **8,70,81,09,410.00**

Report No. 6

Task- **Quantity Sold In Last Two Quarter**



Description-

This sheet shows Total Quantity Sold in the last two quarters in a cement factory.
DAX formula used here-

DAX commands used-

Quantity sold in last two quarter = CALCULATE (SUM(Financial[Sales]),FILTER(ALLSELECTED(Financial), Financial[Quarter] =MAX(Financial[Quarter])-2))

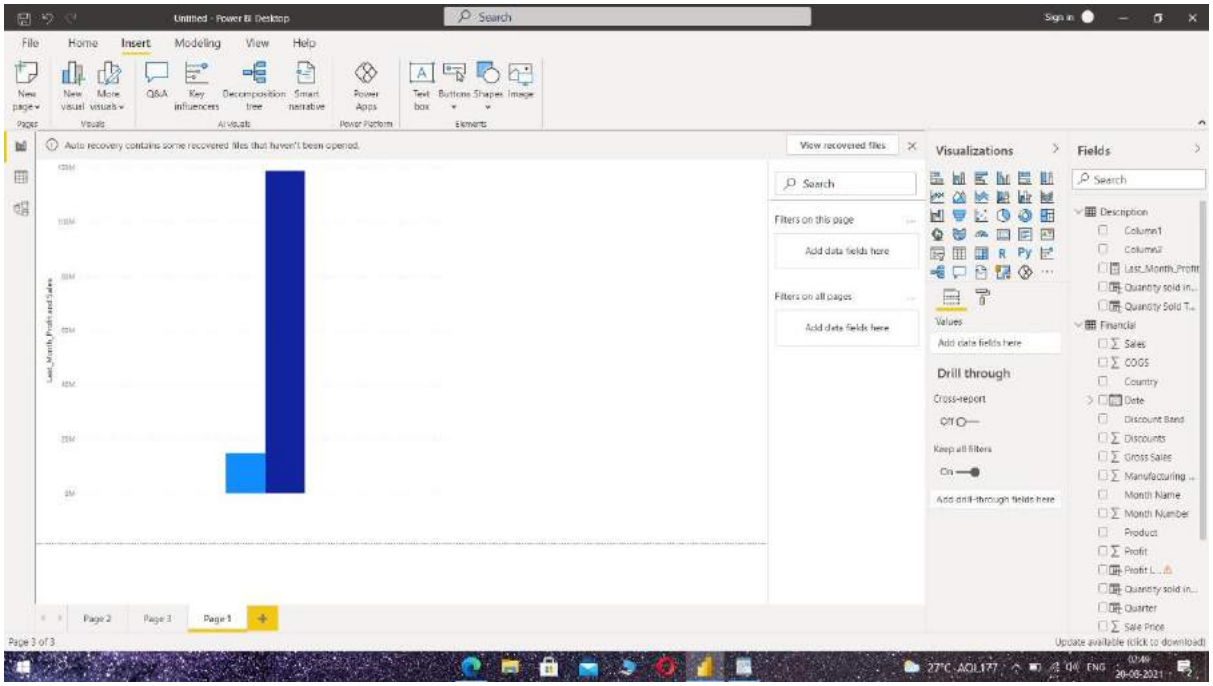
- The **FILTER** functions let you manipulate data context to create dynamic calculations.
- The **ALLSELECTED** function gets the context that represents all rows and columns in the query, while keeping explicit filters and contexts other than row and column filters.

Result-

Total Quantity Sold in Last Two Quarter= **30,41,473.50**

Report No. 7

Task- **Last month profit**



Description:
This data shows profit and sales last month in cement factory. We can calculate profit last month using given DAX formula:

DAX commands used-

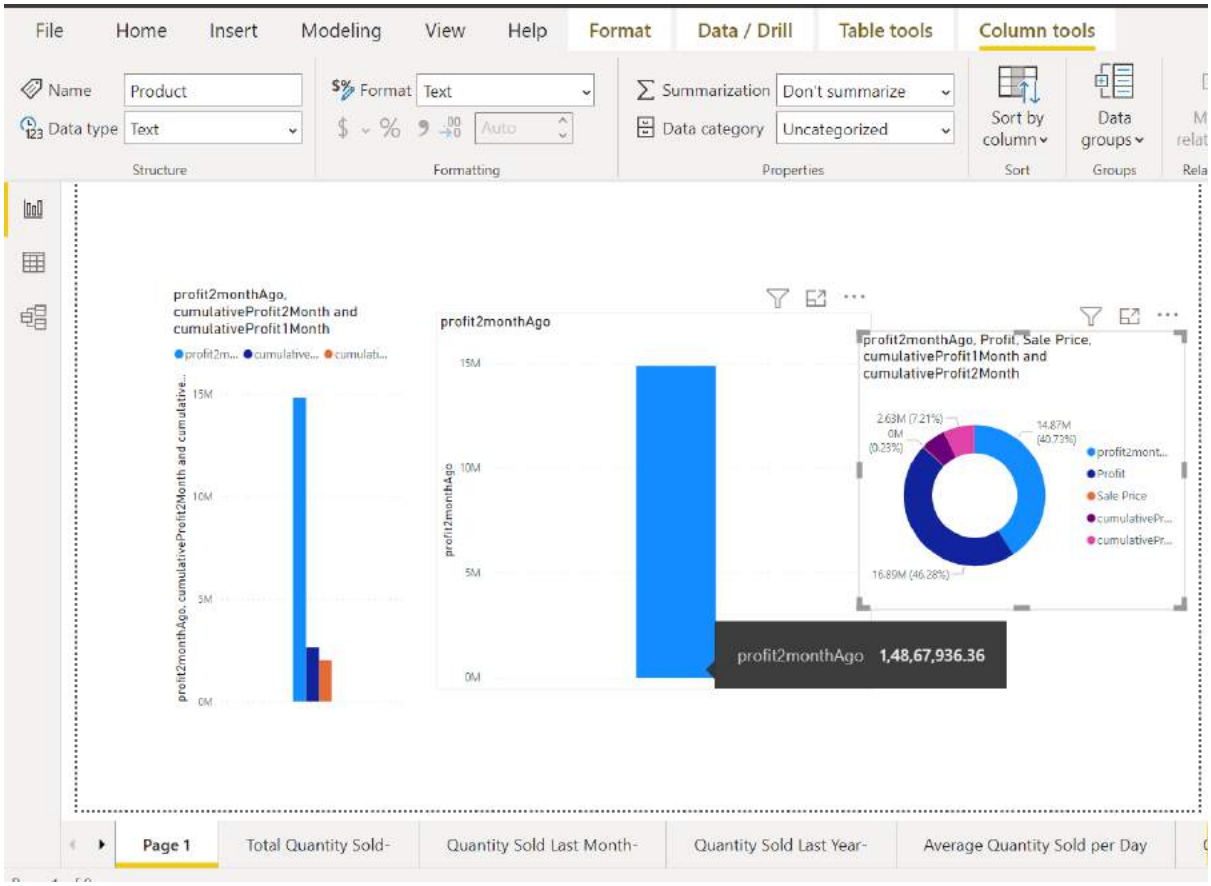
Last_Month_Profit = CALCULATE (SUM ('Financial'[Profit]), DATESBETWEEN('Financial'[Date], MIN('Financial'[Date]), MAX('Financial'[Date]) - 1))

Here we have used **DATESBETWEEN FUNCTION** which returns a table that contains a column of dates that begins with a specified start date and continues until a specified end date. This function is suited to pass as a filter to the CALCULATE function. Use it to filter an expression by a custom date range.

Result-

Last Month Profit: **1,48,67,936. 36**

Profit 2 Month Ago



Description:

This data shows the profit two months ago in Cement factory data.
We can see the profit using this DAX command-

DAX command used-

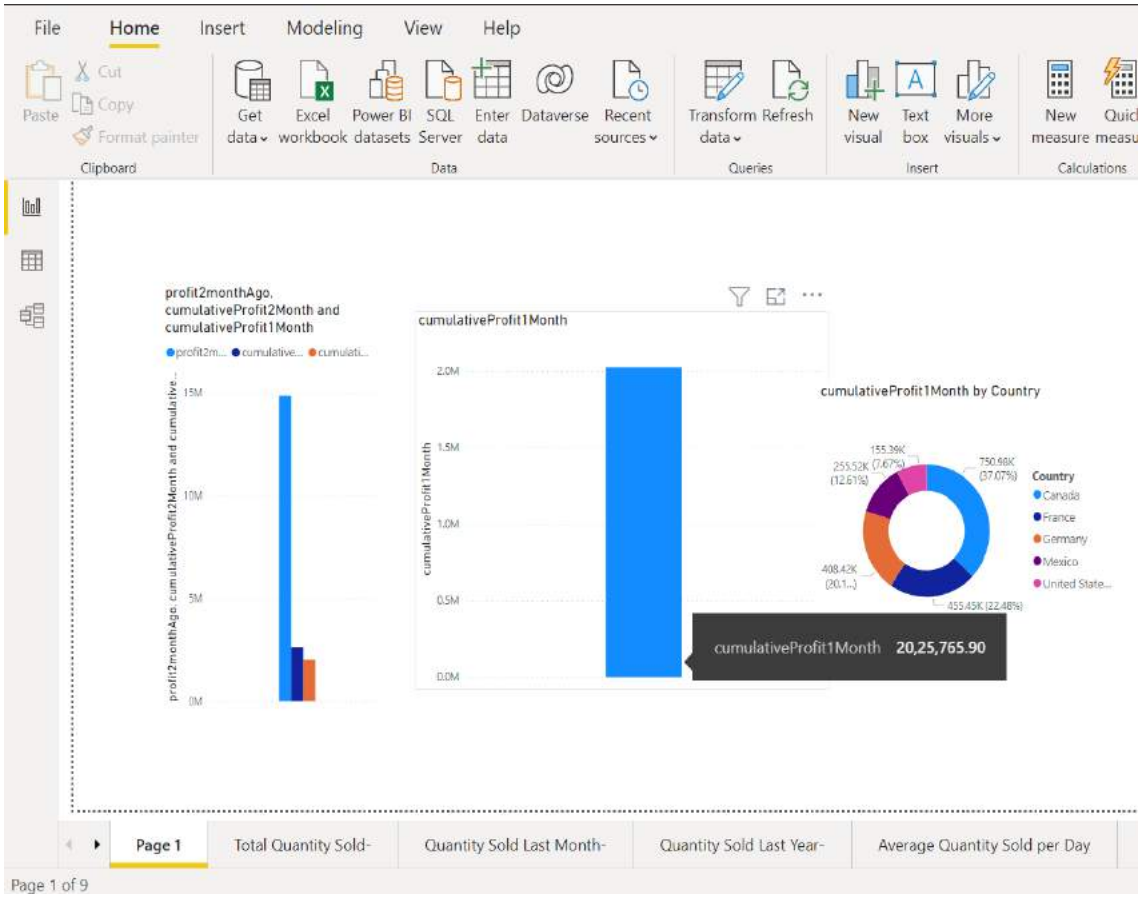
Profit2monthAgo =CALCULATE (SUM ('Financial'[Profit]), DATESBETWEEN('Financial'[Date],MIN('Financial'[Date]),MAX('Financial'[Date]) - 2))

Here we have used **DATESBETWEEN FUNCTION-** Returns a table that contains a column of dates that begins with a specified start date and continues until a specified end date. This function is suited to pass as a filter to the CALCULATE function. Use it to filter an expression by a custom date range.

Result-

Total Profit2monthAgo **= 1,48,67,936.36**

Last Month Cumulative Profit-



Description:

This data shows Last Month Cumulative Profit in Cement factory data.
We can see the using this DAX command-

DAX command used-

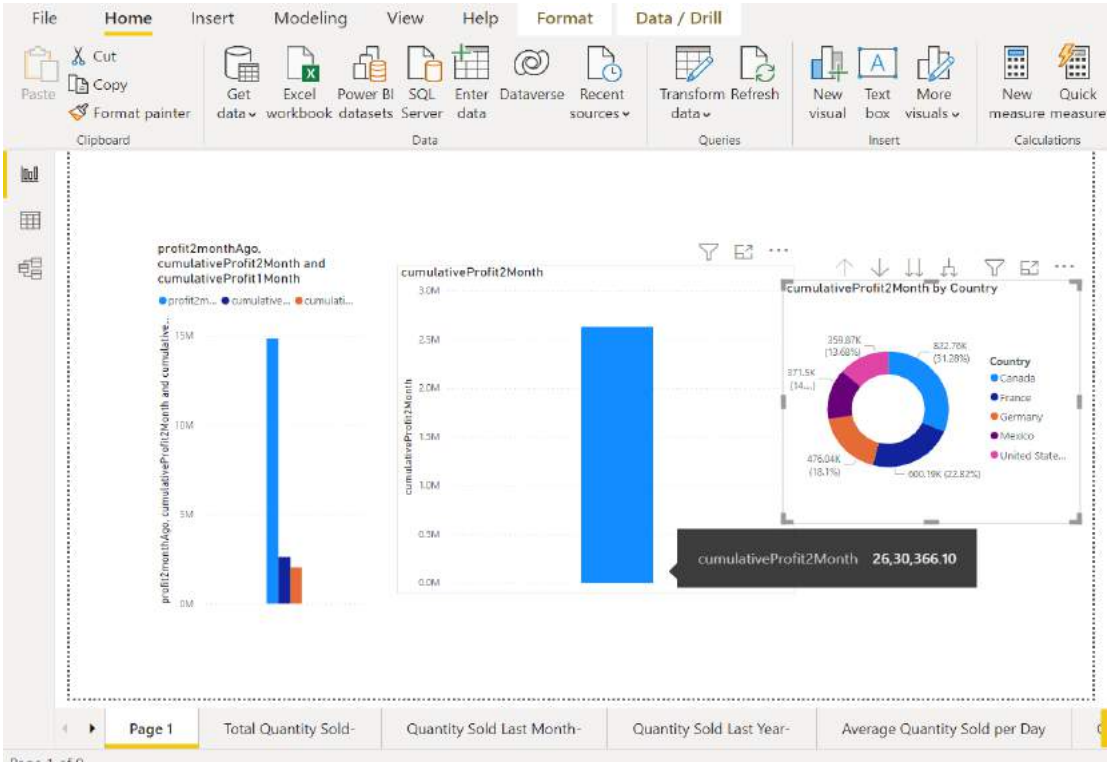
cumulativeProfit1Month = CALCULATE(SUM('Financial'[Profit]), DATESINPERIOD ('Financial'[Date], MAX ('Financial'[Date]), -1, MONTH))

Here we have used **DATESINPERIOD**-Returns the dates from the given period. The dates argument must be a reference to a date/time column and the result table includes only dates that exist in the dates column.

Result-

Total Last Month Cumulative Profit**-20,25,765**

Cumulative Profit 2 Month Ago



Description:
This Visualization shows Cumulative Profit 2 Month Ago using DAX command.

DAX command used-

cumulativeProfit2Month = CALCULATE(SUM('Financial'[Profit]), DATESINPERIOD ('Financial'[Date], MAX ('Financial'[Date]), -2, MONTH))

Here we have used **DATESINPERIOD**-Returns the dates from the given period.
The dates argument must be a reference to a date/time column and the result table includes only dates that exist in the dates column.
All dates need to be present for the years required. The *Date* table must be marked as a date table in the model in case the relationship between the *Date* table and any other table is not based on the *Date*.

Result-

Total Cumulative Profit 2 Month Ago -**26,30,336.10**

Report no.11

Task-**Total Profit running total in Date.**

Description-

This visualization shows the running total in date.

We have used a list view to show the total date-wise which cumulatively adds up every day when new orders are registered.

We have also added a waterfall chart to show the increase per year and it also gives a rough idea of what is its contribution in the total sales. The scorecard shows the grand total.

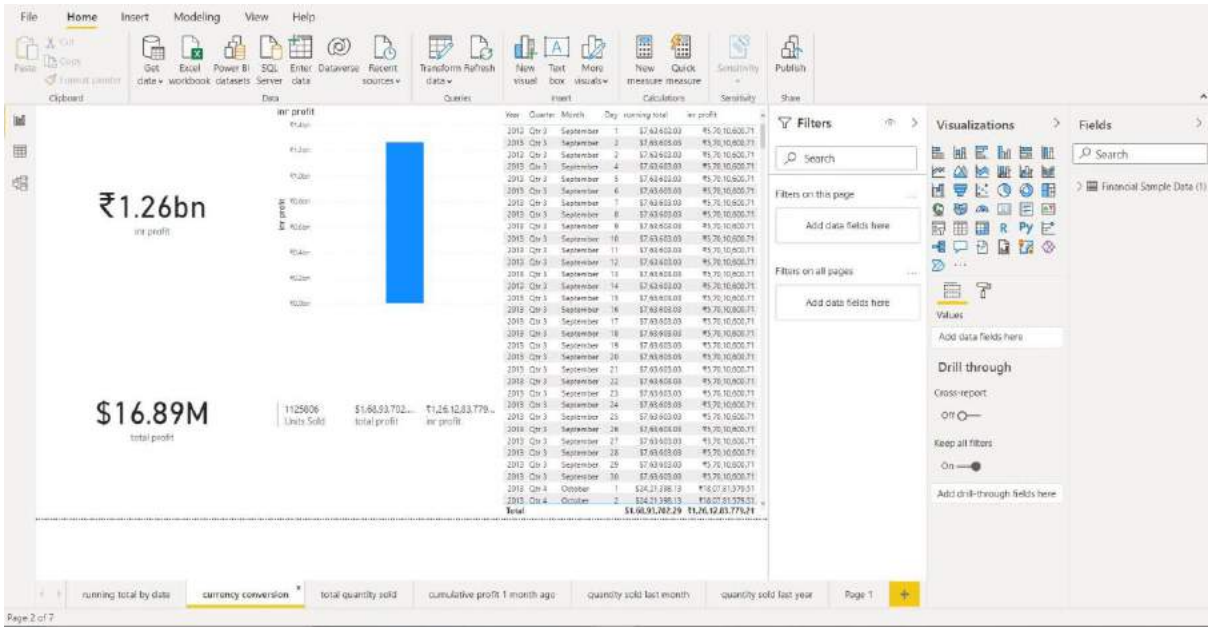
DAX command used-

Running total = CALCULATE([total profit],FILTER(ALL('Financial Sample Data (1)'),'Financial Sample Data (1)'[date].[Date] <= MAX('Financial Sample Data (1)'[date].[Date])))

Result-
Running total in date: $ 16.89 million.

Report no. 12

Task-**Apply Currency Conversion in it.**



Description-

The data is recorded in USD and has to be converted into INR (or any other currency) thus conversion needs to be applied to it.
1 USD = 74.96 INR

This conversion can be applied to each and every cell of the data table.

The data is added to show the grand total in USD and then the currency is converted into it by multiplying the effective conversion.

For multiple conversions, a table can be made which stores the conversion rates and then it can be multiplied to the data as per the currency that we want to see.

DAX commands used-
inr profit = SUMX(ALL('Financial Sample Data (1)'),0.10665714*CALCULATE([total profit],FILTER(ALL('Financial Sample Data (1)'),'Financial Sample Data (1)'[date].[Date] <= MAX('Financial Sample Data (1)'[date].[Date]))))

Result-

Total sales in USD: **16.89 million dollars**
Totals sales in INR: **1.26 billion rupees**

Report No. 13

How to apply the Fiscal Year.

A fiscal year is a one-year period that companies and governments use for financial reporting and budgeting. A fiscal year is most used for accounting purposes to prepare financial statements. Although a fiscal year can start on January 1st and end on December 31st, not all fiscal years correspond with the calendar year.

A fiscal year is a one-year period chosen by a company to report its financial information.

Financial reports, external audits, and federal tax filings are based on a company's fiscal year.

Companies may choose to report their financial information on a non-calendar fiscal year based on the specific nature and revenue cycle of that business.

Steps to create fiscal year on the data:

1. Date table is created in DAX in power BI Desktop
2. Fiscal year number sort column is created in DAX
3. A relationship is established between the fact table and the date table.
4. A column chart and a slicer with month name field are created.
5. The month name is sorted with the fiscal year number field**.**

CONCLUSION

Visualisation can provide a quick, high-level summary of the main information contained in the data.
Quite often the initial data investigations can lead to more questions and further exploration. Sometimes the data shows some unexpected patterns and outliers – data points which are well outside the normal data range.
Exploring these data points can lead to new discoveries. Graphical representations of data are more effective as a means of communication than long textual files. A story can be told more efficiently, and the time to understand a picture is a fraction of the time that it takes to understand the textual data.
Visualisation can provide quick answers and can improve situational awareness. This in turn can lead to faster and timely decisions. Using innovative data visualisations can make your audience more enthusiastic about what you are trying to communicate to them. This in turn can inspire your audience to act based on your more persuasive visualisation.
Thus, using the Financial Cement Company Dataset, we have created visual reports about the company's sales data of different locations over different sectors from 2015 to 2020. We can effectively investigate past actions and their consequences and can take measured decisions in future.