

SAMPLE POWER BI APP

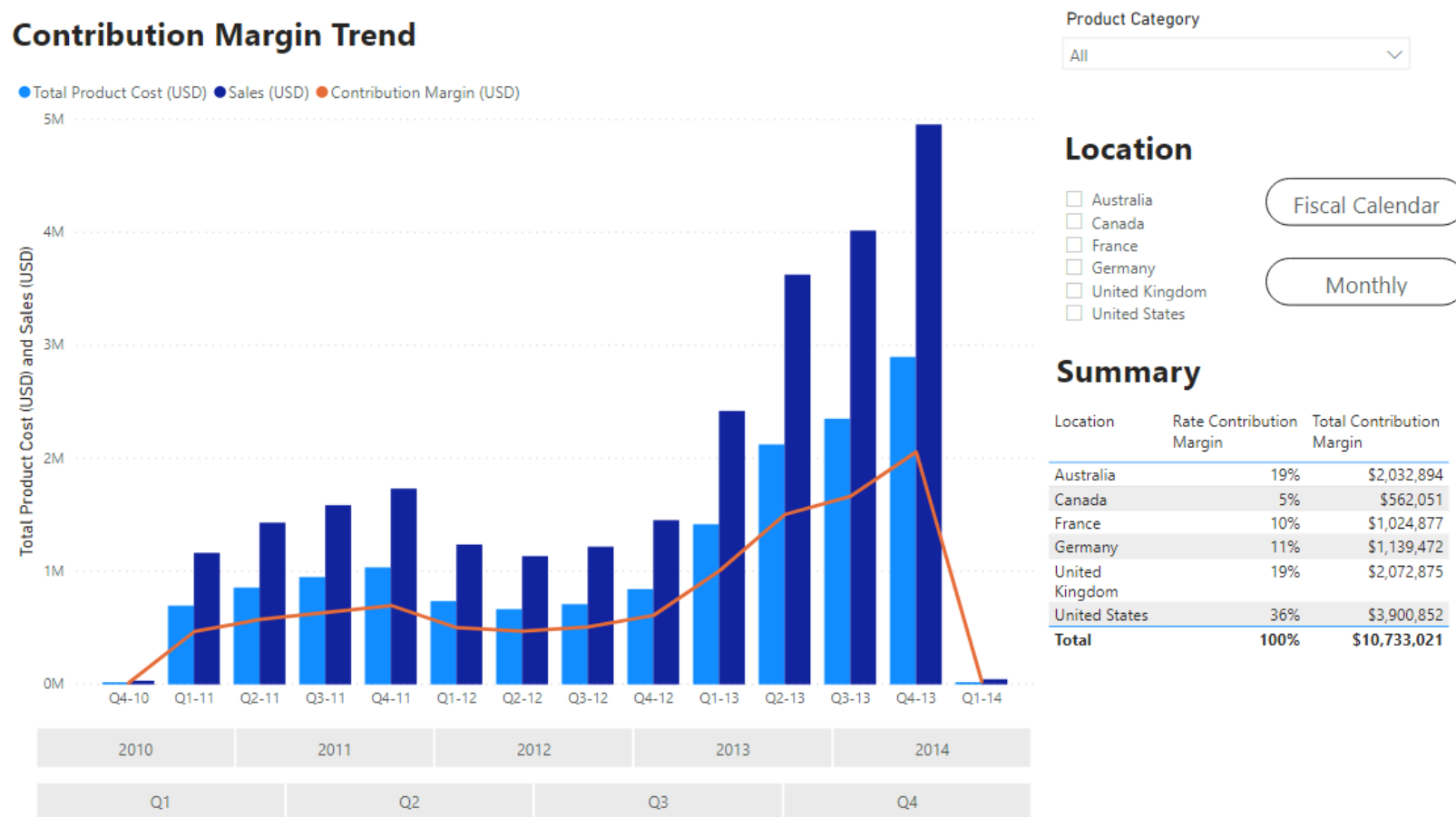
For any potential employer please find attached my sample visualization made **with Power BI** and **SQL Server**.

The dummy database was obtained from **Adventure Works** composed of 15 Dimensional Tables and 16 Fact Tables.

<https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15&tabs=tsql>

The Visualizations represent the Contribution Margin of the organization defined as: (selling Price – Variable Cost) and intends provide an overview of profitability vs sales.

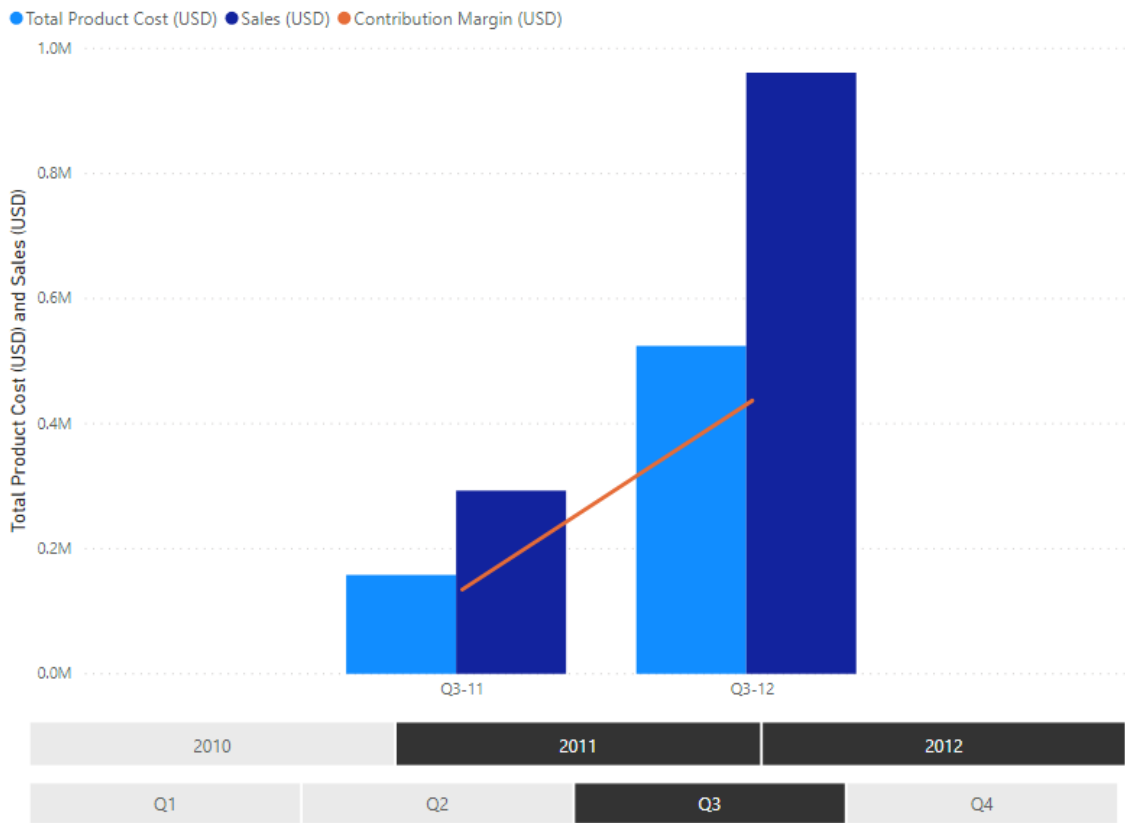
Contribution Margin Trend



The visualization can be slice in location and products, and the **Summary** section provides the **Rate Contribution Margin** for each **Location**.

The trend is available in **Fiscal** and **Gregorian calendar**, and the chart may be slice in **years**, **months** and **quarters** which makes comparison of quarters and month between years insightful and easy for the user.

Contribution Margin Trend



Product Category

Mountain Bikes

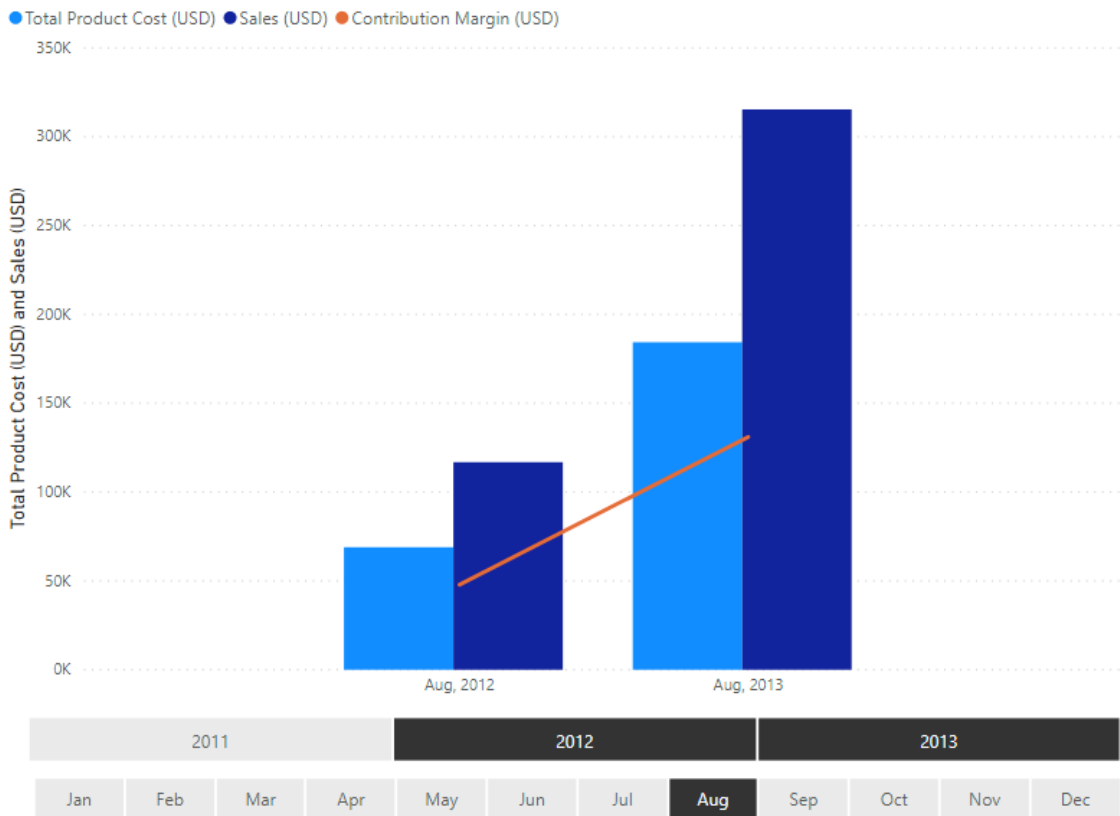
Search

- ☐ Helmets
- ☐ Hydration Packs
- ☐ Jerseys
- ☒ Mountain Bikes
- ☐ Road Bikes
- ☐ Shorts
- ☐ Socks
- ☐ Tires and Tubes
- ☐ Touring Bikes
- ☐ Vests

Summary

Location	Rate Contribution Margin	Total Contribution Margin
Australia	16%	\$90,220
Canada	4%	\$23,868
France	11%	\$63,970
Germany	10%	\$54,818
United Kingdom	20%	\$113,709
United States	39%	\$225,160
Total	100%	\$571,745

Contribution Margin Trend



Product Category

All

Location

- ☐ Australia
- ☐ Canada
- ☒ France
- ☐ Germany
- ☐ United Kingdom
- ☐ United States

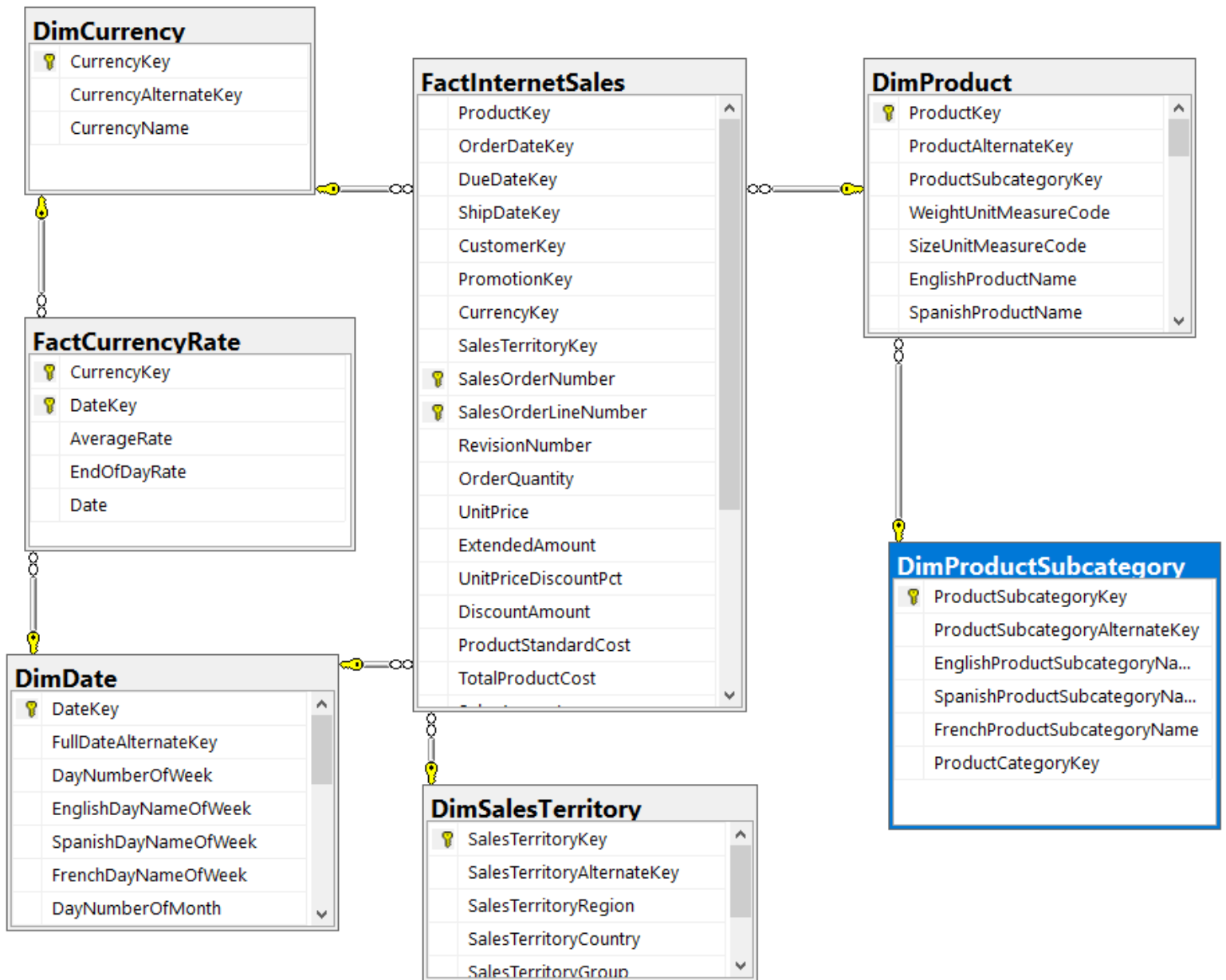
Fiscal Calendar

Quarter

Summary

Location	Rate Contribution Margin	Total Contribution Margin
France	48%	\$86,753
Germany	52%	\$92,175
Total	100%	\$178,928

TABLES INVOLVE DIAGRAM



Adventure Works has a Snowflake schema therefore some of the dimension tables like product are normalized.

FactCurrency Rate used to convert all currencies to (USD)

DimDate is the Adventure Works Calendar, I used this table to generate a Custom Calendar View which allow me to provide the detailed level with easy access.

DimSalesTerritory used to obtain the locations.

DimProductSubcategory used generate the list of products for the slicer.

SQL VIEWS USED FOR THE VISUALIZATIONS

Custom Calendar VIEW used to build the slicers and trend.

```
SELECT *,
-- These string columns can be used as axis in the Trend Charts
CONCAT ( Fiscal_Quarter_Name, '-', RIGHT(FiscalYear, 2) ) AS 'Fiscal Year-Quarter',
-- Sample outupt: Q3-05 (Fiscal starts in Q3)
CONCAT ( Gregorian_Quarter_Name, '-', RIGHT(CalendarYear, 2) ) AS 'Gregorian Year Quarter', --
Sample outupt: Q1-05 (Gregorian starts in Q1)

-- These IDS can be used to OrderBy trend charts.

--- Sample output = 20053 , 20054 , 20061 , 20062 ....
CAST( CONCAT(FiscalYear, FiscalQuarter) AS int) AS 'ID_Fiscal_Year_QUARTER',

--- Sample output = 20052, 20061
CAST( CONCAT(FiscalYear, FiscalSemester) AS int) AS 'ID_Fiscal_Year_Semester',

--- Sample output = 200501, 200502, 200503, 200504
CONCAT (
    CalendarYear,
    REPLICATE('0',2-LEN(RTRIM(CalendarQuarter))) + RTRIM(CalendarQuarter)
) AS ID_Gregorian_Year_QUARTER,

--- Sample output = 200501
CONCAT (
    CalendarYear,
    REPLICATE('0',2-LEN(RTRIM(MonthNumberOfYear))) + RTRIM(MonthNumberOfYear)
) AS ID_Gregorian_Year_Month,

--- Sample output = 20050101
CONCAT (
    CalendarYear,
    REPLICATE('0',2-LEN(RTRIM(DayNumberOfMonth))) + RTRIM(DayNumberOfMonth) ,
    REPLICATE('0',2-LEN(RTRIM(WeekNumberOfYear))) + RTRIM(WeekNumberOfYear)
) AS ID_Gregorian_Year_Month_Week,

--- Sample output = 200501001
CONCAT (
    CalendarYear,
    REPLICATE('0',2-LEN(RTRIM(DayNumberOfMonth))) + RTRIM(DayNumberOfMonth) ,
    REPLICATE('0',3-LEN(RTRIM(MonthNumberOfYear))) + RTRIM(MonthNumberOfYear)
) AS ID_Gregorian_Year_Month_Day
```

```

FROM
(
    SELECT *,
    FORMAT ( FullDateAlternateKey, 'MMM, yyyy') AS 'Gregorian Year-Month',      -- Sample
    outupt: Dec, 2010
    --For Fiscal Quarters
    CASE
        WHEN FiscalQuarter = 1
        THEN 'Q1'
        WHEN FiscalQuarter = 2
        THEN 'Q2'
        WHEN FiscalQuarter = 3
        THEN 'Q3'
        WHEN FiscalQuarter = 4
        THEN 'Q4'
    END AS Fiscal_Quarter_Name,
    --For Gregorian Quarters
    CASE
        WHEN CalendarQuarter = 1
        THEN 'Q1'
        WHEN CalendarQuarter = 2
        THEN 'Q2'
        WHEN CalendarQuarter = 3
        THEN 'Q3'
        WHEN CalendarQuarter = 4
        THEN 'Q4'
    END AS Gregorian_Quarter_Name
    FROM DimDate
) AS temp_Q

```

Sales/Contribution Margin VIEW build to host the data.

```
SELECT
    -- Trend Values
    FIS.SalesAmount * tempC.AverageRate AS 'Sales (USD)',
    FIS.TotalProductCost * tempC.AverageRate AS 'Total Product Cost (USD)',
    (FIS.SalesAmount * tempC.AverageRate) - (FIS.TotalProductCost * tempC.AverageRate) AS
    'Contribution Margin (USD)',

    -- Slice and group by on the Visualization
    dimST.SalesTerritoryCountry,

    -- Trend Axis will depend of these columns
    viewD.[Gregorian Year Quarter],
    viewD.[Fiscal Year-Quarter],
    viewD.[Gregorian Year-Month],

    -- The following columns are used as slice in the visualization.
    CAST(viewD.FiscalYear AS VARCHAR(4)) AS 'Fiscal Year STR',
    CAST(viewD.CalendarYear AS VARCHAR(4)) AS 'Gregorian Year STR',
    LEFT(viewD.EnglishMonthName, 3) AS Month_STR,
    viewD.Fiscal_Quarter_Name AS 'Fiscal Quarter STR',
    viewD.Gregorian_Quarter_Name AS 'Gregorian Quarter STR',

    -- Trend chart will be order by these ID columns.
    -- Sample output INT: 1,2,...,12
    viewD.MonthNumberOfYear AS 'ID_Month_Number',
    -- Sample output INT: 201102
    viewD.ID_Fiscal_Year_QUARTER,
    -- Sample output INT: 201012
    viewD.ID_Gregorian_Year_Month,
    -- Sample output INT: 201004
    viewD.ID_Gregorian_Year_QUARTER,

    -- Product Slicer
    dimPC.EnglishProductSubcategoryName

FROM FactInternetSales AS FIS
    --Added currency conversion rates ( USD )
INNER JOIN tempCurrencyLookup AS tempC
    ON CONCAT(tempC.CurrencyKey,tempC.DateKey) = CONCAT(FIS.CurrencyKey,FIS.OrderDateKey)
    --Added dimension Table Sales Territory
LEFT JOIN DimSalesTerritory AS dimST
    ON FIS.SalesTerritoryKey = dimST.SalesTerritoryKey
    --Added custom calendar View
INNER JOIN vCustomCalendar AS viewD
    ON FIS.OrderDateKey = viewD.DateKey
LEFT JOIN DimProduct as dimP
    ON FIS.ProductKey = dimP.ProductKey
LEFT JOIN DimProductSubcategory AS dimPC
    ON dimP.ProductSubcategoryKey = dimPC.ProductSubcategoryKey
```