

## Prompting for SVG Graphic Measure Code

Here is the bim file for a power bi report. I will be asking you to create two SVG measures for this report:

```
{
  "name": "Untitled",
  "compatibilityLevel": 1550,
  "model": {
    "culture": "en-US",
    "dataAccessOptions": {
      "legacyRedirects": true,
      "returnErrorValuesAsNull": true
    },
    "defaultPowerBIDataSourceVersion": "powerBI_V3",
    "sourceQueryCulture": "en-US",
    "tables": [
      {
        "name": "Dates",
        "lineageTag": "9fd86c9b-fff3-4113-a3df-c2c2f2dbe3fd",
        "dataCategory": "Time",
        "columns": [
          {
            "name": "Date",
            "dataType": "dateTime",
            "isKey": true,
            "sourceColumn": "Date",
            "lineageTag": "6ae0a8ec-5f5c-4540-a297-90f76ffa0d3c",
            "summarizeBy": "none",
            "annotations": [
              {
                "name": "SummarizationSetBy",
                "value": "Automatic"
              }
            ]
          },
          {
            "name": "UnderlyingDateTimeDataType",
            "value": "Date"
          }
        ],
        {
          "name": "PBI_FormatHint",
          "value": "{\\isDateTimeCustom\\":true}"
        }
      ],
      {
        "name": "Year",
        "dataType": "int64",
        "sourceColumn": "Year",
        "formatString": "0",
```

```
"lineageTag": "26d13aee-ac03-42cd-97dc-821cf064fc88",
    "summarizeBy": "none",
    "annotations": [
      {
        "name": "SummarizationSetBy",
        "value": "Automatic"
      }
    ],
    {
      "name": "DateInt",
      "dataType": "int64",
      "sourceColumn": "DateInt",
      "formatString": "0",
      "lineageTag": "6236ffe4-a6d8-4f70-abd1-d368fdc2a173",
      "summarizeBy": "none",
      "annotations": [
        {
          "name": "SummarizationSetBy",
          "value": "Automatic"
        }
      ],
      {
        "name": "Month Name",
        "dataType": "string",
        "sourceColumn": "Month Name",
        "lineageTag": "dd997f30-71ba-45d6-9b8b-7e8042c732a1",
        "summarizeBy": "none",
        "annotations": [
          {
            "name": "SummarizationSetBy",
            "value": "Automatic"
          }
        ],
        {
          "name": "Month & Year",
          "dataType": "string",
          "sourceColumn": "Month & Year",
          "lineageTag": "155f9a56-f90f-4846-90e0-c0aebc593eac",
          "summarizeBy": "none",
          "annotations": [
            {
              "name": "SummarizationSetBy",
              "value": "Automatic"
            }
          ],
          {
            "name": "Quarter & Year",
            "dataType": "string",
            "sourceColumn": "Quarter & Year",
            "lineageTag": "79196840-6e52-4306-ad47-68dc16810239",
```

```
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "ISO Year",
            "dataType": "double",
            "sourceColumn": "ISO Year",
            "formatString": "0",
            "lineageTag": "c3be1652-40cf-4961-bf54-147c3467a3bc",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        {
            "name": "Week & Year",
            "dataType": "string",
            "sourceColumn": "Week & Year",
            "lineageTag": "dfb2ddb4-e6d5-4f0b-9ac8-2abda4aa3ce3",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        {
            "name": "WeeknYear",
            "dataType": "int64",
            "sourceColumn": "WeeknYear",
            "formatString": "0",
            "lineageTag": "48f5e4db-88fc-4a06-96ee-403ab322bac9",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        {
            "name": "MonthnYear",
            "dataType": "int64",
            "sourceColumn": "MonthnYear",
            "formatString": "0",
```

```
"lineageTag": "2af533d7-d2e1-4285-999c-175bf2245f93",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ],
  {
    "name": "QuarternYear",
    "dataType": "int64",
    "sourceColumn": "QuarternYear",
    "formatString": "0",
"lineageTag": "780fed5f-89fd-4b4c-b9da-724dbe811ab1",
    "summarizeBy": "none",
    "annotations": [
      {
        "name": "SummarizationSetBy",
        "value": "Automatic"
      }
    ],
    },
    {
      "name": "Fiscal Year",
      "dataType": "string",
      "sourceColumn": "Fiscal Year",
"lineageTag": "f79bf6c6-8832-48b4-94f2-782e941e24b2",
      "summarizeBy": "none",
      "annotations": [
        {
          "name": "SummarizationSetBy",
          "value": "Automatic"
        }
      ],
      {
        "name": "Fiscal Quarter",
        "dataType": "string",
        "sourceColumn": "Fiscal Quarter",
"lineageTag": "986ce0bb-bc2b-4438-99e0-c66b655af7e5",
        "summarizeBy": "none",
        "annotations": [
          {
            "name": "SummarizationSetBy",
            "value": "Automatic"
          }
        ],
        },
        {
          "name": "Fiscal Period",
          "dataType": "string",
          "sourceColumn": "Fiscal Period",
"lineageTag": "cdc47be0-6c3c-4165-be48-49be8ce54fe6",
```

```
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "IsAfterToday",
            "dataType": "boolean",
            "sourceColumn": "IsAfterToday",
            "formatString": "\"TRUE\";\"TRUE\";\"FALSE\"",
            "lineageTag": "1bbbe355-0959-40fa-b0c5-a0b117a35a5c",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
            {
                "name": "IsHoliday",
                "dataType": "string",
                "sourceColumn": "IsHoliday",
                "lineageTag": "92ac3735-0108-44d0-b9c7-de62f9a32b8e",
                "summarizeBy": "none",
                "annotations": [
                    {
                        "name": "SummarizationSetBy",
                        "value": "Automatic"
                    }
                ],
                {
                    "name": "IsBusinessDay",
                    "dataType": "boolean",
                    "sourceColumn": "IsBusinessDay",
                    "formatString": "\"TRUE\";\"TRUE\";\"FALSE\"",
                    "lineageTag": "8e93a651-22a2-4165-a575-41f2d8746667",
                    "summarizeBy": "none",
                    "annotations": [
                        {
                            "name": "SummarizationSetBy",
                            "value": "Automatic"
                        }
                    ],
                    {
                        "name": "Day Type",
                        "dataType": "string",
                        "sourceColumn": "Day Type",
                        "lineageTag": "eca1c15b-63ce-4b48-bf2b-0173390f8d08",
```

```

        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ]
    },
    {
        "name": "WeekCompleted",
        "dataType": "boolean",
        "sourceColumn": "WeekCompleted",
        "formatString": "\\TRUE\\;\\TRUE\\;\\FALSE\\",
        "lineageTag": "01397d07-bcfb-4145-b6ec-895c3a47c824",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ]
    },
    {
        "name": "MonthCompleted",
        "dataType": "boolean",
        "sourceColumn": "MonthCompleted",
        "formatString": "\\TRUE\\;\\TRUE\\;\\FALSE\\",
        "lineageTag": "43769826-3e25-47d5-acf6-81f2e2d75904",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ]
    },
    {
        "name": "QuarterCompleted",
        "dataType": "boolean",
        "sourceColumn": "QuarterCompleted",
        "formatString": "\\TRUE\\;\\TRUE\\;\\FALSE\\",
        "lineageTag": "cb394596-9eb7-495f-a83f-68590fc8688b",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ]
    },
    {
        "name": "CurrYearOffset",
        "dataType": "int64",
        "sourceColumn": "CurrYearOffset"
    }
]

```

```
        "formatString": "0",
        "lineageTag": "2b5059b6-7b6c-41fa-8ac1-0c1009c3c3df",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "YearCompleted",
            "dataType": "boolean",
            "sourceColumn": "YearCompleted",
            "formatString": "\\\"TRUE\\\";\\\"TRUE\\\";\\\"FALSE\\\"",
            "lineageTag": "bfb8e492-71ea-4bcd-b0cf-1ef3c8efc70c",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        {
            "name": "Quarter Number",
            "dataType": "int64",
            "sourceColumn": "Quarter Number",
            "formatString": "0",
            "lineageTag": "017024f0-894a-4044-a4ac-ebcdb2c434a0",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        {
            "name": "Quarter",
            "dataType": "string",
            "sourceColumn": "Quarter",
            "lineageTag": "1e18f16c-513f-461b-9763-d0301e0d6c43",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        {
            "name": "Start of Quarter",
            "dataType": "dateTime",
```

```
        "sourceColumn": "Start of Quarter",
        "formatString": "Long Date",
        "lineageTag": "1661052d-a7eb-4803-b22d-7965a934c4e0",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            },
            {
                "name": "UnderlyingDateTimeDataType",
                "value": "Date"
            }
        ],
        {
            "name": "End of Quarter",
            "dataType": "dateTime",
            "sourceColumn": "End of Quarter",
            "formatString": "Long Date",
            "lineageTag": "2930d9c1-f38b-4675-97d2-584e0ced4936",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                },
                {
                    "name": "UnderlyingDateTimeDataType",
                    "value": "Date"
                }
            ],
            {
                "name": "CurrQuarterOffset",
                "dataType": "int64",
                "sourceColumn": "CurrQuarterOffset",
                "formatString": "0",
                "lineageTag": "dbd652e6-4bc2-4a86-9d73-71b198a4af19",
                "summarizeBy": "none",
                "annotations": [
                    {
                        "name": "SummarizationSetBy",
                        "value": "Automatic"
                    }
                ],
                {
                    "name": "Month",
                    "dataType": "int64",
                    "sourceColumn": "Month",
                    "formatString": "0",
                    "lineageTag": "26f4110c-2417-4480-b21c-16882e7a4133",
                    "summarizeBy": "none",
```



```
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "Start of Month",
            "dataType": "dateTime",
            "sourceColumn": "Start of Month",
            "formatString": "Long Date",
            "lineageTag": "151ab920-8421-418d-adeb-8750faadb801",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
            "name": "UnderlyingDateTimeDataType",
            "value": "Date"
        }
    ],
    {
        "name": "End of Month",
        "dataType": "dateTime",
        "sourceColumn": "End of Month",
        "formatString": "Long Date",
        "lineageTag": "42b7dabd-5807-4b34-8f8f-2c448d297673",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        "name": "UnderlyingDateTimeDataType",
        "value": "Date"
    }
],
{
    "name": "CurrMonthOffset",
    "dataType": "int64",
    "sourceColumn": "CurrMonthOffset",
    "formatString": "0",
    "lineageTag": "e79d180b-70fc-4ebc-952c-fe3112c98123",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
}
```

```
}
]
},
{
    "name": "Month Short",
    "dataType": "string",
    "sourceColumn": "Month Short",
    "lineageTag": "8bd77115-b915-4c49-b640-5b60c63fbbf2",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "Month Initial",
    "dataType": "string",
    "sourceColumn": "Month Initial",
    "lineageTag": "4b0ee63e-9ede-4cb3-91b8-5f8099d3c14a",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "Day of Month",
    "dataType": "int64",
    "sourceColumn": "Day of Month",
    "formatString": "0",
    "lineageTag": "c99bdabb-d136-479c-bc01-0bb2de17d029",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "Week Number",
    "dataType": "int64",
    "sourceColumn": "Week Number",
    "formatString": "0",
    "lineageTag": "03fa7715-c04c-45ee-8e04-3e2fdfb86fa7",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
}
```

```
    }
  ]
},
{
  "name": "Start of Week",
  "dataType": "dateTime",
  "sourceColumn": "Start of Week",
  "formatString": "Long Date",
  "lineageTag": "0cfb4563-8967-4609-aebe-0fbc6c67065",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    },
    {
      "name": "UnderlyingDateTimeDataType",
      "value": "Date"
    }
  ]
},
{
  "name": "End of Week",
  "dataType": "dateTime",
  "sourceColumn": "End of Week",
  "formatString": "Long Date",
  "lineageTag": "dc636ce2-e878-4c91-9da6-e81213c9e2a9",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    },
    {
      "name": "UnderlyingDateTimeDataType",
      "value": "Date"
    }
  ]
},
{
  "name": "CurrWeekOffset",
  "dataType": "int64",
  "sourceColumn": "CurrWeekOffset",
  "formatString": "0",
  "lineageTag": "051a5d06-bc3f-4c55-adc6-e0f63f4ceb48",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ]
},
{
```

```

        "name": "Day of Week Number",
        "dataType": "int64",
        "sourceColumn": "Day of Week Number",
        "formatString": "0",
        "lineageTag": "e695c4c5-6ca2-4086-814a-f105777c79bf",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
    },
    {
        "name": "Day of Week Name",
        "dataType": "string",
        "sourceColumn": "Day of Week Name",
        "lineageTag": "c774a8c1-1f97-4110-81e7-7ee671d8bde0",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
    },
    {
        "name": "Day of Week Initial",
        "dataType": "string",
        "sourceColumn": "Day of Week Initial",
        "lineageTag": "bc204368-d7ab-454f-a6cf-a1c33aa2b205",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
    },
    {
        "name": "CurrDayOffset",
        "dataType": "double",
        "sourceColumn": "CurrDayOffset",
        "lineageTag": "20d1f9e4-1f3a-4aa3-a0bb-5489698a64e1",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
    },
    {
        "name": "PBI_FormatHint",
        "value": "{\"isGeneralNumber\":true}"
    }
}

```

```
]
},
{
    "name": "IsWeekDay",
    "dataType": "boolean",
    "sourceColumn": "IsWeekDay",
    "formatString": "\\TRUE\\;\\TRUE\\;\\FALSE\\",
    "lineageTag": "ed6083c6-6543-4d12-b420-bd19689ef137",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "ISO CurrYearOffset",
    "dataType": "int64",
    "sourceColumn": "ISO CurrYearOffset",
    "formatString": "0",
    "lineageTag": "3bb65e97-4598-4a03-939c-6ddeaf59176a",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "ISO Quarter",
    "dataType": "string",
    "sourceColumn": "ISO Quarter",
    "lineageTag": "c4285d1e-262d-4dfa-b084-a7266034158b",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "ISO Quarter & Year",
    "dataType": "string",
    "sourceColumn": "ISO Quarter & Year",
    "lineageTag": "2fb4589c-9c41-491e-ae5b-accbc05385c8",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
}
```

```
]
},
{
  "name": "ISO QuarternYear",
  "dataType": "int64",
  "sourceColumn": "ISO QuarternYear",
  "formatString": "0",
  "lineageTag": "60e5da4b-0d14-4904-8ab5-e7d236b2e616",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ]
},
{
  "name": "ISO CurrQuarterOffset",
  "dataType": "int64",
  "sourceColumn": "ISO CurrQuarterOffset",
  "formatString": "0",
  "lineageTag": "b41d7521-0aa0-4fea-bb7c-66712ba5834b",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ]
},
{
  "name": "Fiscal CurrYearOffset",
  "dataType": "int64",
  "sourceColumn": "Fiscal CurrYearOffset",
  "formatString": "0",
  "lineageTag": "dd895a2b-fe0a-4c5b-ac26-33a692e30e6b",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ]
},
{
  "name": "FQuarternYear",
  "dataType": "int64",
  "sourceColumn": "FQuarternYear",
  "formatString": "0",
  "lineageTag": "3f326010-1176-4a75-bb44-3f7aaa2bc83c",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
```

```
        "value": "Automatic"
      }
    ]
  },
  {
    "name": "FPeriodnYear",
    "dataType": "int64",
    "sourceColumn": "FPeriodnYear",
    "formatString": "0",
    "lineageTag": "Od26f193-8621-41ec-b4e3-9b5666f4a26b",
    "summarizeBy": "none",
    "annotations": [
      {
        "name": "SummarizationSetBy",
        "value": "Automatic"
      }
    ]
  },
  {
    "name": "Fiscal Week",
    "dataType": "string",
    "sourceColumn": "Fiscal Week",
    "lineageTag": "370862e1-3935-4c9e-a053-e811643cfbd0",
    "summarizeBy": "none",
    "annotations": [
      {
        "name": "SummarizationSetBy",
        "value": "Automatic"
      }
    ]
  },
  {
    "name": "FWeeknYear",
    "dataType": "int64",
    "sourceColumn": "FWeeknYear",
    "formatString": "0",
    "lineageTag": "92c91f4e-314e-4540-bfab-6123a6bae676",
    "summarizeBy": "none",
    "annotations": [
      {
        "name": "SummarizationSetBy",
        "value": "Automatic"
      }
    ]
  },
  {
    "name": "IsCurrentFY",
    "dataType": "boolean",
    "sourceColumn": "IsCurrentFY",
    "formatString": "\"TRUE\";\"TRUE\";\"FALSE\"",
    "lineageTag": "ef18a2ad-f425-4591-998e-0187305a6278",
    "summarizeBy": "none",
    "annotations": [
      {
```

```
        "name": "SummarizationSetBy",
        "value": "Automatic"
    }
]
},
{
    "name": "IsCurrentFQ",
    "dataType": "boolean",
    "sourceColumn": "IsCurrentFQ",
    "formatString": "\\TRUE\\";\\TRUE\\";\\FALSE\\",
    "lineageTag": "d1d27b20-441a-442f-98d1-b13b8a001eb4",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "IsCurrentFP",
    "dataType": "boolean",
    "sourceColumn": "IsCurrentFP",
    "formatString": "\\TRUE\\";\\TRUE\\";\\FALSE\\",
    "lineageTag": "a016e896-6444-4fec-b701-2c8b79f0b5d1",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "IsCurrentFW",
    "dataType": "boolean",
    "sourceColumn": "IsCurrentFW",
    "formatString": "\\TRUE\\";\\TRUE\\";\\FALSE\\",
    "lineageTag": "d428c8d9-76bd-4096-b302-47179d2ca7b6",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "IsPYTD",
    "dataType": "boolean",
    "sourceColumn": "IsPYTD",
    "formatString": "\\TRUE\\";\\TRUE\\";\\FALSE\\",
    "lineageTag": "6b6ce0ed-c7b2-4207-8226-50ab13cef680",
    "summarizeBy": "none",
```



```

        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "IsPFYTD",
            "dataType": "boolean",
            "sourceColumn": "IsPFYTD",
            "formatString": "\\TRUE\\;\\TRUE\\;\\FALSE\\",
            "lineageTag": "5e216d1f-c1a0-4c34-b3e0-8d34b60d6cba",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ]
        },
        "partitions": [
            {
                "name": "Dates-0da6e4d2-656b-4b7e-aca5-34ab2fe0c288",
                "mode": "import",
                "queryGroup": "Data Model",
                "source": {
                    "type": "m",
                    "expression": [
                        "let",
                        " Source = #\\\"Extended Date Table\\\"(fxStartDate, fxEndDate, 7, null, null)",
                        "in",
                        " Source"
                    ]
                }
            },
            {
                "name": "PBI_ResultType",
                "value": "Table"
            }
        ],
        {
            "name": "Sales",
            "lineageTag": "a67a5af8-8fb4-424e-bd7f-d3085a7873a6",
            "columns": [
                {
                    "name": "OrderNumber",
                    "dataType": "string",
                    "sourceColumn": "OrderNumber",
                    "lineageTag": "c2221886-407b-48cc-a3ea-b6ce1db5349d",

```

```
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "OrderDate",
            "dataType": "dateTime",
            "sourceColumn": "OrderDate",
            "formatString": "Long Date",
            "lineageTag": "c9e846fb-8960-48d6-b6f9-afae4235537c",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
            "name": "UnderlyingDateTimeDataType",
            "value": "Date"
        }
    ],
    {
        "name": "Customer Name Index",
        "dataType": "int64",
        "sourceColumn": "Customer Name Index",
        "formatString": "0",
        "lineageTag": "c95f4389-2a76-414c-966b-ea42705683c4",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "Channel",
            "dataType": "string",
            "sourceColumn": "Channel",
            "lineageTag": "46bc3675-7762-433f-ad8d-12aba3db7ab5",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ]
        },
    ],
    {
```

```
        "name": "Currency Code",
        "dataType": "string",
        "sourceColumn": "Currency Code",
        "lineageTag": "7c363a5f-2fbd-4deb-8efc-fdea1a443b82",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "Warehouse Code",
            "dataType": "string",
            "sourceColumn": "Warehouse Code",
            "lineageTag": "b6af3048-9d35-4393-ab95-f90f98e0f32a",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
            {
                "name": "Delivery Region Index",
                "dataType": "int64",
                "sourceColumn": "Delivery Region Index",
                "formatString": "0",
                "lineageTag": "293ab1f0-5eea-49e8-b8d9-6be2747b5e60",
                "summarizeBy": "none",
                "annotations": [
                    {
                        "name": "SummarizationSetBy",
                        "value": "Automatic"
                    }
                ],
                {
                    "name": "Product Description Index",
                    "dataType": "int64",
                    "sourceColumn": "Product Description Index",
                    "formatString": "0",
                    "lineageTag": "7581aaf0-7be6-42a0-80e9-a3cb27ae3f8e",
                    "summarizeBy": "none",
                    "annotations": [
                        {
                            "name": "SummarizationSetBy",
                            "value": "Automatic"
                        }
                    ],
                    {
```

```
        "name": "Order Quantity",
        "dataType": "int64",
        "sourceColumn": "Order Quantity",
        "formatString": "0",
        "lineageTag": "defc8fdf-479b-4256-948b-5569cb62d70d",
        "summarizeBy": "sum",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "Unit Price",
            "dataType": "double",
            "sourceColumn": "Unit Price",
            "lineageTag": "47b200c7-aaf0-4768-98d4-83e64dc7623d",
            "summarizeBy": "sum",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
            {
                "name": "PBI_FormatHint",
                "value": "{\\isGeneralNumber\\:true}"
            }
        ],
        {
            "name": "Line Total",
            "dataType": "double",
            "sourceColumn": "Line Total",
            "lineageTag": "05b89b4e-557d-47ad-a7bd-cc6fbd831e55",
            "summarizeBy": "sum",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
            {
                "name": "PBI_FormatHint",
                "value": "{\\isGeneralNumber\\:true}"
            }
        ],
        {
            "name": "Total Unit Cost",
            "dataType": "double",
            "sourceColumn": "Total Unit Cost",
            "lineageTag": "dd122d77-b232-48dc-8e7c-ab81a0c0a504",
            "summarizeBy": "sum",
            "annotations": [
```

```

        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        },
        {
            "name": "PBI_FormatHint",
            "value": "{\\"isGeneralNumber\\":true}"
        }
    ]
},
"partitions": [
    {
        "name": "Sales-bacb6b5a-89e4-4297-9714-047dc1722c49",
        "mode": "import",
        "queryGroup": "Data Model",
        "source": {
            "type": "m",
            "expression": [
                "let",
                " Source = Table.Combine( { Sales_2014, Sales_2015, Sales_2016 } ),",
                " #\\"Update OrderDate\\" = Table.TransformColumns(",
                " Source,",
                " {",
                " {",
                " \\"OrderDate\\",",
                " each Date.AddDays(, DayOffset)",",
                " type date",
                " }",",
                " }",",
                " )",",
                " in",
                " #\\"Update OrderDate\\"
            ]
        }
    }
],
"annotations": [
    {
        "name": "PBI_ResultType",
        "value": "Table"
    }
],
{
    "name": "Products",
    "lineageTag": "a9978b5d-cc6a-4ba4-8e70-49bfd16068d2",
    "columns": [
        {
            "name": "Index",
            "dataType": "int64",
            "sourceColumn": "Index",
            "formatString": "0",
            "lineageTag": "9fde6254-df25-4a56-91a8-6418d7d2b062",

```

```

        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "Product Name",
            "dataType": "string",
            "sourceColumn": "Product Name",
            "lineageTag": "f44964cd-0499-4ece-9013-3ecdd26e5ae1",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ]
        },
        "partitions": [
            {
                "name": "Products-11e5c890-bcc2-49b8-9d06-94d5e43c04e9",
                "mode": "import",
                "queryGroup": "Data Model",
                "source": {
                    "type": "m",
                    "expression": [
                        "let",
                        " Source = Excel.Workbook(File.Contents(pSource), null, true),",
                        " Products_Data_Table = Source[[Item=\"Products_Data\",Kind=\"Table\"]][Data],",
                        " #\"Changed Type\" = Table.TransformColumnTypes(Products_Data_Table,",
                        " {{\"Index\", Int64.Type}, {\"Product Name\", type text}})",
                        "in",
                        " #\"Changed Type\""
                    ]
                }
            },
            {
                "name": "PBI_ResultType",
                "value": "Table"
            }
        ],
        {
            "name": "LinkedQueryName",
            "value": "Products"
        }
    ],
    {
        "name": "Customers",

```

```
"lineageTag": "ea152682-ed5f-4e01-80d9-cebbc2c51763",
  "columns": [
    {
      "name": "Customer Index",
      "dataType": "int64",
      "sourceColumn": "Customer Index",
      "formatString": "0",
      "lineageTag": "6e91d423-c0a2-4c23-892f-47321e9457b6",
      "summarizeBy": "none",
      "annotations": [
        {
          "name": "SummarizationSetBy",
          "value": "Automatic"
        }
      ]
    },
    {
      "name": "Customer Names",
      "dataType": "string",
      "sourceColumn": "Customer Names",
      "lineageTag": "7f85488c-c8e1-468c-ade5-893f170a0184",
      "summarizeBy": "none",
      "annotations": [
        {
          "name": "SummarizationSetBy",
          "value": "Automatic"
        }
      ]
    },
    {
      "name": "Emails",
      "dataType": "string",
      "sourceColumn": "Emails",
      "lineageTag": "0104126b-7313-4ad5-aa5d-b242e1b06b9a",
      "summarizeBy": "none",
      "annotations": [
        {
          "name": "SummarizationSetBy",
          "value": "Automatic"
        }
      ]
    },
    {
      "name": "Head Office Index",
      "dataType": "int64",
      "sourceColumn": "Head Office Index",
      "formatString": "0",
      "lineageTag": "d4f71d54-6435-4fa4-a125-5b8a1f09883d",
      "summarizeBy": "none",
      "annotations": [
        {
          "name": "SummarizationSetBy",
          "value": "Automatic"
        }
      ]
    }
  ]
}
```

```

    ]
  }
},
"partitions": [
  {
    "name": "Customers-07bdf62c-8594-49dc-aec4-32363d535168",
    "mode": "import",
    "queryGroup": "Data Model",
    "source": {
      "type": "m",
      "expression": [
        "let",
        " Source = Excel.Workbook(File.Contents(pSource), null, true),",
        " Customer_Data_Table = Source[[Item=\"Customer_Data\";Kind=\"Table\"]",
        "[Data],",
        " #\"Changed Type\" = Table.TransformColumnTypes(Customer_Data_Table,",
        "{{\"Customer Index\", Int64.Type}, {\"Customer Names\", type text}, {\"Emails\", type",
        "text}, {\"Head Office Index\", Int64.Type}})",
        "in",
        " #\"Changed Type\""]
      ]
    }
  ],
"annotations": [
  {
    "name": "PBI_ResultType",
    "value": "Table"
  },
  {
    "name": "LinkedQueryName",
    "value": "Customers"
  }
],
{
  "name": "Channels",
  "lineageTag": "c4d74f09-76c9-4e03-bffd-09fa312a7763",
  "columns": [
    {
      "name": "Channel Name",
      "dataType": "string",
      "sourceColumn": "Channel Name",
      "lineageTag": "b8dddf8d-eff0-41fc-b354-b247b2231d1a",
      "summarizeBy": "none",
      "annotations": [
        {
          "name": "SummarizationSetBy",
          "value": "Automatic"
        }
      ]
    },
    {
      "name": "Channel Code",

```



```

        "dataType": "string",
        "sourceColumn": "Channel Code",
        "lineageTag": "d24838cc-56b8-443c-96e7-19e51b475038",
        "summarizeBy": "none",
        "annotations": [
            {
                "name": "SummarizationSetBy",
                "value": "Automatic"
            }
        ],
        {
            "name": "Alt. Name",
            "dataType": "string",
            "sourceColumn": "Alt. Name",
            "lineageTag": "4a89e75e-3d70-4259-a607-3efbaf7aa09c",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        {
            "name": "Importance",
            "dataType": "string",
            "sourceColumn": "Importance",
            "lineageTag": "052b0524-88fe-47d3-886b-f0ceb05cce72",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        {
            "name": "Prior Code",
            "dataType": "string",
            "sourceColumn": "Prior Code",
            "lineageTag": "8fadfc2f-c306-43a8-a9ad-5e63a3098ce9",
            "summarizeBy": "none",
            "annotations": [
                {
                    "name": "SummarizationSetBy",
                    "value": "Automatic"
                }
            ],
        },
        "partitions": [
            {
                "name": "Channels-3e1b3435-29ec-46b9-bf8e-b70ba709d88b",

```

```

        "mode": "import",
        "queryGroup": "Data Model",
        "source": {
            "type": "m",
            "expression": [
                "let",
                " Source = Excel.Workbook(File.Contents(pSource), null, true);",
                " Channel_Details_Table = Source[{Item=\"Channel_Details\",Kind=\"Table\"}]
                    [Data];",
                " #\"Changed Type\" = Table.TransformColumnTypes(Channel_Details_Table,
                {\"Channel Name\", type text}, {\"Channel Code\", type text}, {\"Alt. Name\", type
                text}, {\"Importance\", type text}, {\"Prior Code\", type text}})",
                "in",
                " #\"Changed Type\"
            ]
        }
    },
    "annotations": [
        {
            "name": "PBI_ResultType",
            "value": "Table"
        },
        {
            "name": "LinkedQueryName",
            "value": "Channels"
        },
        {
            "name": "Regions",
            "lineageTag": "be6249bb-7cea-446e-adb0-b9096fea0f4a",
            "columns": [
                {
                    "name": "Index",
                    "dataType": "int64",
                    "sourceColumn": "Index",
                    "formatString": "0",
                    "lineageTag": "66d99d3b-5d04-48f7-b799-7429f0f54d61",
                    "summarizeBy": "none",
                    "annotations": [
                        {
                            "name": "SummarizationSetBy",
                            "value": "Automatic"
                        }
                    ]
                },
                {
                    "name": "City",
                    "dataType": "string",
                    "sourceColumn": "City",
                    "lineageTag": "475dc351-1ded-4917-9161-670c2ae38ceb",
                    "summarizeBy": "none",
                    "annotations": [

```

```

        "name": "SummarizationSetBy",
        "value": "Automatic"
    }
    ]
},
{
    "name": "Country",
    "dataType": "string",
    "sourceColumn": "Country",
    "lineageTag": "5c5eb761-715d-4616-b464-51d04bffb4",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
{
    "name": "Full Name",
    "dataType": "string",
    "sourceColumn": "Full Name",
    "lineageTag": "2da24670-f2da-48b4-9923-759e61eec039",
    "summarizeBy": "none",
    "annotations": [
        {
            "name": "SummarizationSetBy",
            "value": "Automatic"
        }
    ]
},
    "partitions": [
        {
            "name": "Regions-3d84dc38-66ea-4c1e-8f50-4d39eb6fc732",
            "mode": "import",
            "queryGroup": "Data Model",
            "source": {
                "type": "m",
                "expression": [
                    "let",
                    " Source = Excel.Workbook(File.Contents(pSource), null, true),",
                    " Regions_Table_Table = Source[[Item=\\\"Regions_Table\\\",Kind=\\\"Table\\\"]][Data],",
                    " #\\\"Changed Type\\\" = Table.TransformColumnTypes(Regions_Table_Table,",
                    "{\\\"Index\\\", Int64.Type}, {\\\"City\\\", type text}, {\\\"Country\\\", type text}, {\\\"Full Name\\\",",
                    "type text})",
                    "in",
                    " #\\\"Changed Type\\\"""
                ]
            }
        }
    ],
    "annotations": [

```

```

        {
            "name": "PBI_ResultType",
            "value": "Table"
        },
        {
            "name": "LinkedQueryName",
            "value": "Regions"
        }
    ]
},
{
    "name": "Measures",
    "lineageTag": "4f2be79e-0893-41c6-b73f-95a3e128ac41",
    "partitions": [
        {
            "name": "Measures-b0549638-03af-4c7a-b111-76faf88d0cce",
            "mode": "import",
            "source": {
                "type": "m",
                "expression": [
                    "let",
                    " Source = #table({\"Column1\"},{null}),",
                    "\" #\"Removed Columns\" = Table.RemoveColumns(Source,{\"Column1\"}),",
                    "in",
                    " #\"Removed Columns\""
                ]
            }
        }
    ],
    "measures": [
        {
            "name": "Total Sales",
            "expression": [
                "",
                "SUM ( Sales[Line Total])"
            ],
            "lineageTag": "e61c6aea-8800-4627-9988-9197d563a827",
            "annotations": [
                {
                    "name": "PBI_FormatHint",
                    "value": "{\"isGeneralNumber\":true}"
                }
            ]
        },
        {
            "name": "Total Sales PY",
            "expression": [
                "",
                "VAR __CurrYrOffset = MAX(Dates[CurrYearOffset]),",
                "VAR __PrevYrOffset = __CurrYrOffset - 1",
                "",
                "VAR __vTable =",
                "FILTER(",
                "SUMMARIZE(",

```

```

        " ALL( Sales ),",
        " Dates[Date],",
        " Dates[Year],",
        " Dates[CurrYearOffset],",
        " Dates[Quarter & Year],",
        " Sales[Line Total]",
        " ),",
        " Dates[CurrYearOffset] = __PrevYrOffset",
        ")",
        "",
        "VAR __Result =",
        "SUMX(",
        " __vTable,",
        " [Total Sales]",
        ")",
        "",
        "RETURN __Result"
    ],
    "lineageTag": "755e40ad-377a-4f2e-8969-0f10070c42f4",
    "annotations": [
        {
            "name": "PBI_FormatHint",
            "value": "{\\isGeneralNumber\\":true}"
        }
    ],
    {
        "name": "Sales Target",
        "expression": [
            "",
            "[Total Sales PY] * 1.2"
        ],
        "lineageTag": "87ad6ac2-4520-4efe-acde-e4c5a464da3f",
        "annotations": [
            {
                "name": "PBI_FormatHint",
                "value": "{\\isGeneralNumber\\":true}"
            }
        ],
        {
            "name": "BulletChartSVG",
            "expression": [
                "",
                "VAR __TotSales = [Total Sales]",
                "VAR __SalesTarget = [Sales Target]",
                "VAR __FillColor = \"DarkSlateBlue\"",
                "VAR __ChartWidth = 100",
                "VAR __ChartHeight = 20",
                "VAR __ScaleFactor = DIVIDE(__ChartWidth, __SalesTarget, 0)",
                "VAR __BarLength = MIN(__TotSales * __ScaleFactor, __ChartWidth)",
                "VAR __TargetPosition = __ChartWidth",
                "VAR __SVG =",
                "\"data:image/svg+xml;utf8,\" &"
            ]
        }
    ]
}

```

```

"\"<svg width=\"\" & __ChartWidth & \"\" height=\"\" & __ChartHeight & \"\"
    xmlns=\"\"http://www.w3.org/2000/svg\">\",
    "<rect x=\"\"0\"\" y=\"\"5\"\" width=\"\" & __BarLength & \"\" height=\"\"10\"\"
        fill=\"\" & __FillColor & \"\"></rect>",
    "<line x1=\"\" & __TargetPosition & \"\" y1=\"\"0\"\" x2=\"\" & __TargetPosition
        & \"\" y2=\"\" & __ChartHeight & \"\" stroke=\"\"black\"\" stroke-
            width=\"\"2\"\"></line>",
    "</svg>\"",
    "RETURN __SVG"
    ],
    "lineageTag": "2fa3ba17-5f7e-4804-82ba-e1168aa6055d",
    "dataCategory": "ImageUrl"
    }
    ],
    "annotations": [
        {
            "name": "PBI_ResultType",
            "value": "Table"
        }
    ],
    "relationships": [
        {
            "name": "8deccdd1-6e71-479a-8710-52b4d95e5c2f",
            "fromTable": "Sales",
            "fromColumn": "OrderDate",
            "toTable": "Dates",
            "toColumn": "Date"
        },
        {
            "name": "e33098a6-e27b-463f-9114-0cda10974231",
            "fromTable": "Sales",
            "fromColumn": "Product Description Index",
            "toTable": "Products",
            "toColumn": "Index"
        },
        {
            "name": "df095986-8cfb-44db-b53d-8d7e5a3a366f",
            "fromTable": "Sales",
            "fromColumn": "Delivery Region Index",
            "toTable": "Regions",
            "toColumn": "Index"
        },
        {
            "name": "2e5c434a-a11a-45eb-8130-f58651f7796e",
            "fromTable": "Sales",
            "fromColumn": "Channel",
            "toTable": "Channels",
            "toColumn": "Channel Name"
        },
        {
            "name": "49e87e24-24e7-445d-b001-86f987d32882",
            "fromTable": "Sales",

```

```

        "fromColumn": "Customer Name Index",
        "toTable": "Customers",
        "toColumn": "Customer Index"
    }
},
    "cultures": [
        {
            "name": "en-US",
            "linguisticMetadata": {
                "content": {
                    "Version": "1.0.0",
                    "Language": "en-US"
                }
            },
            "contentType": "json"
        }
    ],
    "expressions": [
        {
            "name": "Sales_2015",
            "kind": "m",
            "expression": [
                "let",
                " Source = Excel.Workbook(File.Contents(pSource), null, true),",
                " Sales_2014_Table = Source[[Item=\"Sales_2015\",Kind=\"Table\"]][Data],",
                " #\"Changed Type\" = Table.TransformColumnTypes(Sales_2014_Table,",
                "{\"OrderNumber\", type text}, {\"OrderDate\", type date}, {\"Customer Name",
                "Index\", Int64.Type}, {\"Channel\", type text}, {\"Currency Code\", type text},",
                {\"Warehouse Code\", type text}, {\"Delivery Region Index\", Int64.Type}, {\"Product",
                "Description Index\", Int64.Type}, {\"Order Quantity\", Int64.Type}, {\"Unit Price\",",
                "type number}, {\"Line Total\", type number}, {\"Total Unit Cost\", type number}})",
                "in",
                " #\"Changed Type\""
            ],
            "queryGroup": "Data Prep",
            "lineageTag": "d07de681-d86e-49ec-adc5-5b82a5956605",
            "annotations": [
                {
                    "name": "PBI_ResultType",
                    "value": "Table"
                }
            ],
            "name": "Sales_2016",
            "kind": "m",
            "expression": [
                "let",
                " Source = Excel.Workbook(File.Contents(pSource), null, true),",
                " Sales_2014_Table = Source[[Item=\"Sales_2016\",Kind=\"Table\"]][Data],",
                " #\"Changed Type\" = Table.TransformColumnTypes(Sales_2014_Table,",
                "{\"OrderNumber\", type text}, {\"OrderDate\", type date}, {\"Customer Name",
                "Index\", Int64.Type}, {\"Channel\", type text}, {\"Currency Code\", type text},",
                {\"Warehouse Code\", type text}, {\"Delivery Region Index\", Int64.Type}, {\"Product

```

```

Description Index\", Int64.Type}, {\\"Order Quantity\\", Int64.Type}, {\\"Unit Price\\",
type number}, {\\"Line Total\\", type number}, {\\"Total Unit Cost\\", type number}}),
    "in",
    "#\\"Changed Type\\"
    ],
    "queryGroup": "Data Prep",
    "lineageTag": "69e79bd6-c9c5-4bec-9bd7-2cae0626f441",
    "annotations": [
        {
            "name": "PBI_ResultType",
            "value": "Table"
        }
    ],
    {
        "name": "Extended Date Table",
        "kind": "m",
        "expression": [
            "let fnDateTable = ( StartDate as date, EndDate as date, optional
FYStartMonthNum as number, optional Holidays as list, optional WDStartNum as
number, optional AddRelativeNetWorkdays as logical ) as table =>",
            " let",
            " FYStartMonth = List.Select( {1..12}, each _ = FYStartMonthNum ){0}? ?? 1,",
            " WDStart = List.Select( {0..1}, each _ = WDStartNum ){0}? ?? 0,",
            " CurrentDate = Date.From( DateTime.FixedLocalNow()),",
            " DayCount = Duration.Days( Duration.From( EndDate - StartDate)) + 1,",
            " Source = List.Dates( StartDate, DayCount, #duration(1,0,0,0)),",
            " AddToday = if EndDate < CurrentDate then List.Combine( {Source,
{CurrentDate}}) else Source,",
            " ToTable = Table.FromList(AddToday, Splitter.SplitByNothing(), type table [Date =
Date.Type] ),",
            " InsertYear = Table.AddColumn(ToTable, \\"Year\\", each Date.Year([Date]), type
number),",
            " InsertYearOffset = Table.AddColumn(InsertYear, \\"CurrYearOffset\\", each
Date.Year([Date]) - Date.Year( Date.From(CurrentDate)), type number),",
            " InsertCompletedYear = Table.AddColumn(InsertYearOffset, \\"YearCompleted\\",
each Date.EndOfYear([Date]) < Date.From( Date.EndOfYear(CurrentDate)), type
logical),",
            "",
            " InsertQuarterNum = Table.AddColumn(InsertCompletedYear, \\"Quarter
Number\\", each Date.QuarterOfYear([Date]), type number),",
            " InsertQuarter = Table.AddColumn(InsertQuarterNum, \\"Quarter\\", each \\"Q\\" &
Number.ToText([Quarter Number]), type text),",
            " InsertStartOfQuarter = Table.AddColumn(InsertQuarter, \\"Start of Quarter\\",
each Date.StartOfQuarter([Date]), type date),",
            " InsertEndOfQuarter = Table.AddColumn(InsertStartOfQuarter, \\"End of
Quarter\\", each Date.EndOfQuarter([Date]), type date),",
            " InsertCalendarQtr = Table.AddColumn(InsertEndOfQuarter, \\"Quarter & Year\\",
each \\"Q\\" & Number.ToText( Date.QuarterOfYear([Date])) & Date.ToText([Date],
[Format = \\" yyyy\\"]), type text),",
            " InsertQuarternYear = Table.AddColumn(InsertCalendarQtr, \\"QuarternYear\\",
each [Year] * 10 + [Quarter Number], type number),",
            " InsertQuarterOffset = Table.AddColumn(InsertQuarternYear,
\\"CurrQuarterOffset\\", each ((4 * Date.Year([Date])) +

```



```

Date.QuarterOfYear([Date])) - ((4 * Date.Year(Date.From(CurrentDate))) +
Date.QuarterOfYear(Date.From(CurrentDate))), type number),"
" InsertCompletedQuarter = Table.AddColumn(InsertQuarterOffset,
\QuarterCompleted", each Date.EndOfQuarter([Date]) <
Date.From(Date.EndOfQuarter(CurrentDate)), type logical),"
",
" InsertMonth = Table.AddColumn(InsertCompletedQuarter, \Month", each
Date.Month([Date]), type number),"
" InsertStartOfMonth = Table.AddColumn(InsertMonth, \Start of Month", each
Date.StartOfMonth([Date]), type date),"
" InsertEndOfMonth = Table.AddColumn(InsertStartOfMonth, \End of Month",
each Date.EndOfMonth([Date]), type date),"
" InsertCalendarMonth = Table.AddColumn(InsertEndOfMonth, \Month & Year",
each Text.Proper( Date.ToText([Date], [Format = \MMM yyyy\])), type text),"
" InsertMonthnYear = Table.AddColumn(InsertCalendarMonth, \MonthnYear",
each [Year] * 100 + [Month], type number),"
" InsertMonthOffset = Table.AddColumn(InsertMonthnYear, \CurrMonthOffset",
each ((12 * Date.Year([Date])) + Date.Month([Date])) - ((12 *
Date.Year(Date.From(CurrentDate))) + Date.Month(Date.From(CurrentDate))),
type number),"
" InsertCompletedMonth = Table.AddColumn(InsertMonthOffset,
\MonthCompleted", each Date.EndOfMonth([Date]) <
Date.From(Date.EndOfMonth(CurrentDate)), type logical),"
" InsertMonthName = Table.AddColumn(InsertCompletedMonth, \Month Name",
each Text.Proper( Date.ToText([Date], \MMMM")), type text),"
" InsertMonthShort = Table.AddColumn( InsertMonthName, \Month Short", each
Text.Proper( Date.ToText([Date], \MMM")), type text),"
" InsertMonthInitial = Table.AddColumn(InsertMonthShort, \Month Initial", each
Text.Start([Month Name], 1) & Text.Repeat( Character.FromNumber(8203),
Date.Month([Date]) ), type text),"
" InsertDayOfMonth = Table.AddColumn(InsertMonthInitial, \Day of Month",
each Date.Day([Date]), type number),"
",
" InsertWeekNumber = Table.AddColumn(InsertDayOfMonth, \Week Number",
each
" if Number.RoundDown((Date.DayOfYear([Date])-(Date.DayOfWeek([Date],
Day.Monday)+1)+10)/7)=0",
" then Number.RoundDown((Date.DayOfYear(#date(Date.Year([Date])-1,12,31))-
(Date.DayOfWeek(#date(Date.Year([Date])-1,12,31), Day.Monday)+1)+10)/7)",
" else if (Number.RoundDown((Date.DayOfYear([Date])-(
Date.DayOfWeek([Date], Day.Monday)+1)+10)/7)=53 and
(Date.DayOfWeek(#date(Date.Year([Date]),12,31), Day.Monday)+1<4))",
" then 1 else Number.RoundDown((Date.DayOfYear([Date])-(
Date.DayOfWeek([Date], Day.Monday)+1)+10)/7), type number),"
" InsertStartOfWeek = Table.AddColumn(InsertWeekNumber, \Start of Week",
each Date.StartOfWeek([Date], Day.Monday), type date),"
" InsertWeekEnding = Table.AddColumn(InsertStartOfWeek, \End of Week",
each Date.EndOfWeek( [Date], Day.Monday), type date),"
" InsertCalendarWk = Table.AddColumn(InsertWeekEnding, \Week & Year", each
\W" & Text.PadStart( Text.From( [Week Number] ), 2, \0\)) & \ " &
Text.From(Date.Year( Date.AddDays( Date.StartOfWeek([Date], Day.Monday), 3
))), type text),"
" InsertWeeknYear = Table.AddColumn(InsertCalendarWk, \WeeknYear", each
Date.Year( Date.AddDays( Date.StartOfWeek([Date], Day.Monday), 3 )) * 100 +

```

```

[Week Number], Int64.Type),"
" InsertWeekOffset = Table.AddColumn(InsertWeeknYear, \"CurrWeekOffset\",
    each (Number.From(Date.StartOfWeek([Date], Day.Monday))-
Number.From(Date.StartOfWeek(CurrentDate, Day.Monday)))/7, type number),",
    " InsertCompletedWeek = Table.AddColumn(InsertWeekOffset,
    \"WeekCompleted\", each Date.EndOfWeek( [Date], Day.Monday) <
    Date.From(Date.EndOfWeek(CurrentDate, Day.Monday)), type logical),",
    " ",
    " InsertDayWeek = Table.AddColumn(InsertCompletedWeek, \"Day of Week
Number\", each Date.DayOfWeek([Date], Day.Monday) + WDStart, Int64.Type),",
    " InsertDayName = Table.AddColumn(InsertDayWeek, \"Day of Week Name\",
    each Text.Proper( Date.ToText([Date], \"dddd\" )), type text),",
    " InsertDayInitial = Table.AddColumn(InsertDayName, \"Day of Week Initial\",
    each Text.Proper(Text.Start([Day of Week Name], 1)) & Text.Repeat(
    Character.FromNumber(8203), Date.DayOfWeek([Date], Day.Monday) +
    WDStart ), type text),",
    " InsertDayOfYear = Table.AddColumn(InsertDayInitial, \"Day of Year\", each
    Date.DayOfYear([Date]), Int64.Type),",
    " InsertDayInt = Table.AddColumn(InsertDayOfYear, \"DateInt\", each [Year] *
    10000 + [Month] * 100 + [Day of Month], type number),",
    " InsertDayOffset = Table.AddColumn(InsertDayInt, \"CurrDayOffset\", each
    Number.From([Date]) - Number.From(CurrentDate), type number),",
    " InsertIsAfterToday = Table.AddColumn(InsertDayOffset, \"IsAfterToday\", each
    not ([Date] <= Date.From(CurrentDate)), type logical),",
    " InsertIsWorkingDay = Table.AddColumn(InsertIsAfterToday, \"IsWeekDay\",
    each if Date.DayOfWeek([Date], Day.Monday) > 4 then false else true, type
    logical),",
    " InsertIsHoliday = Table.AddColumn(InsertIsWorkingDay, \"IsHoliday\", each if
    Holidays = null then \"Unknown\" else List.Contains( Holidays, [Date] ), if Holidays
    = null then type text else type logical),",
    " InsertIsBusinessDay = Table.AddColumn(InsertIsHoliday, \"IsBusinessDay\", each
    if [IsWeekDay] = true and [IsHoliday] <> true then true else false, type logical),",
    " InsertDayType = Table.AddColumn(InsertIsBusinessDay, \"Day Type\", each if
    [IsHoliday] = true then \"Holiday\" else if [IsWeekDay] = false then \"Weekend\"
    else if [IsWeekDay] = true then \"Weekday\" else null, type text),",
    " ",
    " InsertISOYear = Table.AddColumn( InsertDayType, \"ISO Year\", each Date.Year(
    Date.AddDays( Date.StartOfWeek([Date], Day.Monday), 3 )), type number),",
    " InsertISOQNum = Table.AddColumn(InsertISOYear, \"ISO Quarter Number\",
    each if [Week Number] >39 then 4 else if [Week Number] >26 then 3 else if [Week
    Number] >13 then 2 else 1, Int64.Type),",
    " InsertISOQtr = Table.AddColumn(InsertISOQNum, \"ISO Quarter\", each \"Q\" &
    Number.ToText([ISO Quarter Number]), type text),",
    " InsertISOQuarter = Table.AddColumn(InsertISOQtr, \"ISO Quarter & Year\", each
    \"Q\" & Number.ToText([ISO Quarter Number]) & \" \" & Number.ToText([ISO Year]),
    type text),",
    " InsertISOQny = Table.AddColumn(InsertISOQuarter, \"ISO QuarternYear\", each
    [ISO Year] * 10 + [ISO Quarter Number], type number),",
    " ",
    " // BufferTable = Table.Buffer(Table.Distinct( InsertISOQny[[ISO Year],
    [DateInt]])),",
    " // InsertISODay = Table.AddColumn(InsertISOQny, \"ISO Day of Year\", (OT) =>
    Table.RowCount( Table.SelectRows( BufferTable, (IT) => IT[DateInt] <=
    OT[DateInt] and IT[ISO Year] = OT[ISO Year])), Int64.Type),",

```

```

" AddFY = Table.AddColumn(InsertISOqNy, \"Fiscal Year\", each \"FY\" & (if
[Month] >= FYStartMonth and FYStartMonth >1 then Text.From([Year] + 1) else
Text.From([Year])), type text),
" //AddFYs = Table.AddColumn(AddFY, \"Fiscal Year short\", each \"FY\" & (if
[Month] >= FYStartMonth and FYStartMonth >1 then Text.PadEnd( Text.End(
Text.From([Year] + 1), 2), 2, \"0\") else Text.End( Text.From([Year]), 2)), type text),
" AddFQ = Table.AddColumn(AddFY, \"Fiscal Quarter\", each \"FQ\" & Text.From(
Number.RoundUp( Date.Month( Date.AddMonths( [Date], - (FYStartMonth - 1) )) /
3 )) & \" \" & (if [Month] >= FYStartMonth and FYStartMonth >1 then
Text.From([Year] + 1) else Text.From([Year])), type text),
" AddFQnYr = Table.AddColumn(AddFQ, \"FQuarternYear\", each (if [Month] >=
FYStartMonth and FYStartMonth >1 then [Year] + 1 else [Year]) * 10 +
Number.RoundUp( Date.Month( Date.AddMonths( [Date], - (FYStartMonth - 1) )) /
3 ), type number),
" AddFM = Table.AddColumn(AddFQnYr, \"Fiscal Period Number\", each if [Month]
>= FYStartMonth and FYStartMonth >1 then [Month] - (FYStartMonth-1) else if
[Month] >= FYStartMonth and FYStartMonth = 1 then [Month] else [Month] + (12-
FYStartMonth+1), type number),
" AddFP = Table.AddColumn(AddFM, \"Fiscal Period\", each \"FP\" & Text.PadStart(
Text.From([Fiscal Period Number]), 2, \"0\") & \" \" & (if [Month] >= FYStartMonth
and FYStartMonth >1 then Text.From([Year] + 1) else Text.From([Year])), type
text),
" AddFMnYr = Table.AddColumn(AddFP, \"FPeriodnYear\", each (if [Month] >=
FYStartMonth and FYStartMonth >1 then [Year] + 1 else [Year]) * 100 + [Fiscal
Period Number], type number),
" FYCalendarStart = #date( Date.Year(StartDate)-1, FYStartMonth, 1 ),
" InsertFFD = Table.AddColumn( AddFMnYr, \"FiscalFirstDay\", each if [Month] >=
FYStartMonth and FYStartMonth >1 then #date( Date.Year([Date])+1,
FYStartMonth, 1) else #date( Date.Year([Date]), FYStartMonth, 1), type date ),
"",
" InitTable = Table.FromList( List.Transform( {Number.From(FYCalendarStart) ..
Number.From(EndDate)}, Date.From), Splitter.SplitByNothing(), type table
[DateFW = Date.Type]),
" AddFFD = Table.AddColumn( InitTable, \"FiscalFirstDay\", each if
Date.Month([DateFW]) < FYStartMonth then #date(Date.Year([DateFW]),
FYStartMonth, 1) else #date(Date.Year([DateFW]) + 1, FYStartMonth, 1)),
" AddFWSD = Table.AddColumn( AddFFD, \"FWStartDate\", each
Date.AddYears(Date.StartOfWeek([DateFW], Day.Monday), 1)),
" Group1 = Table.Group( AddFWSD, {\"FiscalFirstDay\", \"FWStartDate\"},
{{\"AllRows\", each _, type table [DateFW = nullable date, FiscalFirstDay = date,
FWStartDate = date]}}),
" Group2 = Table.Group( Group1, {\"FiscalFirstDay\"}, {{\"AllRows2\", each _, type
table [FiscalFirstDay = date, FWStartDate = date, AllRows = table]}}),
" AddIndex = Table.AddColumn( Group2, \"Custom\", each
Table.AddIndexColumn([AllRows2], \"Fiscal Week Number\", 1, 1) [[Custom]],
" ExpandG2 = Table.ExpandTableColumn( AddIndex, \"Custom\",
{\"FiscalFirstDay\", \"FWStartDate\", \"AllRows\", \"Fiscal Week Number\"},
{\"FiscalFirstDay\", \"FWStartDate\", \"AllRows\", \"Fiscal Week Number\"}),
" ExpandG1 = Table.ExpandTableColumn( ExpandG2, \"AllRows\", {\"DateFW\",
\"DateFW\"}) [[DateFW], [Fiscal Week Number]],
" MergeFYW = Table.Join( InsertFFD, {\"Date\"}, ExpandG1, {\"DateFW\"},
JoinKind.LeftOuter, JoinAlgorithm.SortMerge ),
" FWlogic = List.Contains( {null}, FYStartMonthNum),
" UpdateFYWeek = if FWlogic then Table.ReplaceValue(MergeFYW, each [Fiscal

```

```

Week Number], each if FYStartMonth = 1 then [Week Number] else [Fiscal Week
Number], Replacer.ReplaceValue, {"Fiscal Week Number\"]) else MergeFYW",
" AddFYW = Table.AddColumn( UpdateFYWeek, "Fiscal Week\)", each if FWlogic
then "F" & [# "Week & Year\)" else if FYStartMonth = 1 then "FW" &
Text.PadStart( Text.From([Fiscal Week Number]), 2, "0") & Date.ToText([Date], "
yyyy\)" else if Date.Month([Date]) < FYStartMonth then "FW" & Text.PadStart(
Text.From([Fiscal Week Number]), 2, "0") & Date.ToText([Date], " yyyy\)" else
"FW" & Text.PadStart(Text.From([Fiscal Week Number]), 2, "0") & " " &
Text.From( Date.Year([Date])+1, type text),
" InsertFWeeknYear = Table.AddColumn(AddFYW, "FWeeknYear", each if
FWlogic then [WeeknYear] else (if FYStartMonth = 1 then Date.Year([Date]) else if
Date.Month([Date]) < FYStartMonth then Date.Year([Date]) else
Date.Year([Date])+1) * 100 + [Fiscal Week Number], Int64.Type),
" ",
" CurrentDateRecord = Table.SelectRows(InsertFWeeknYear, each ([Date] =
CurrentDate)),
" CurrentISOyear = CurrentDateRecord{0}[ISO Year],
" CurrentISOqtr = CurrentDateRecord{0}[ISO Quarter Number],
" CurrentYear = CurrentDateRecord{0}[Year],
" CurrentMonth = CurrentDateRecord{0}[Month],
" CurrentFiscalFirstDay = CurrentDateRecord{0}[FiscalFirstDay],
" PrevFiscalFirstDay = Date.AddYears(CurrentFiscalFirstDay, -1),
" CurrentFY = CurrentDateRecord{0}[Fiscal Year],
" CurrentFQ = CurrentDateRecord{0}[FQuarternYear],
" CurrentFP = CurrentDateRecord{0}[FPeriodnYear],
" CurrentFW = CurrentDateRecord{0}[FWeeknYear],
"",
" InsertISOYrOffset = Table.AddColumn(InsertFWeeknYear, "ISO
CurrYearOffset", each [ISO Year] - CurrentISOyear, type number),
" InsertISOQtrOffset = Table.AddColumn(InsertISOYrOffset, "ISO
CurrQuarterOffset", each ((4 * [ISO Year]) + [ISO Quarter Number]) - ((4 *
CurrentISOyear) + CurrentISOqtr), type number),
" InsertFYoffset = Table.AddColumn(InsertISOQtrOffset, "Fiscal
CurrYearOffset", each try (if [Month] >= FYStartMonth then [Year]+1 else [Year]) -
(if CurrentMonth >= FYStartMonth then CurrentYear+1 else CurrentYear)
otherwise null, type number),
" InsertCurrentFY = Table.AddColumn(InsertFYoffset, "IsCurrentFY", each if
[Fiscal Year] = CurrentFY then true else false, type logical),
" InsertCurrentFQ = Table.AddColumn(InsertCurrentFY, "IsCurrentFQ", each if
[FQuarternYear] = CurrentFQ then true else false, type logical),
" InsertCurrentFP = Table.AddColumn(InsertCurrentFQ, "IsCurrentFP", each if
[FPeriodnYear] = CurrentFP then true else false, type logical),
" InsertCurrentFW = Table.AddColumn(InsertCurrentFP, "IsCurrentFW", each if
[FWeeknYear] = InsertISOYrOffset then true else false, type logical),
" InsertPYTD = Table.AddColumn(InsertCurrentFW, "IsPYTD", each if
CurrentYear-1 = [Year] and [Day of Year] <= CurrentDateRecord{0}[Day of Year]
then true else false, type logical),
" ListPrevFYDates = List.Buffer( Table.SelectRows( Table.ExpandTableColumn(
Table.NestedJoin(
" Table.AddIndexColumn( Table.RenameColumns( Table.TransformColumnTypes(
Table.FromList( List.Dates( PrevFiscalFirstDay,
Number.From(CurrentFiscalFirstDay-PrevFiscalFirstDay),#duration(1,0,0,0)),
Splitter.SplitByNothing()),{"Column1", type date})), {"Column1", "DateFY"},
"Index", 1, 1), {"Index"},

```

```

" Table.AddIndexColumn( Table.RenameColumns( Table.TransformColumnTypes(
Table.FromList( List.Dates( Date.AddYears( PrevFiscalFirstDay, -1), Number.From(
PrevFiscalFirstDay - Date.AddYears( PrevFiscalFirstDay, -1)),#duration(1,0,0,0)),
Splitter.SplitByNothing()),{{\"Column1\", type date}}, {{\"Column1\", \"DateFY\"}},
\"Index\", 1, 1),
, {\"Index\"}, \"Table\", JoinKind.LeftOuter), \"Table\", {\"DateFY\"},
{\"PrevDateFY\"}), each [DateFY] <= CurrentDate[PrevDateFY] ),,
" InsertPFYTD = Table.AddColumn(InsertPYTD, \"IsPFYTD\", each if [Fiscal
CurrYearOffset] = -1 and List.Contains(ListPrevFYDates, [Date] ) then true else
false, type logical),,
" InsertNetWorkdays = if AddRelativeNetWorkdays = true then
Table.AddColumn(InsertPFYTD, \"Relative Networkdays\", each
fxNETWORKDAYS( StartDate, [Date], Holidays ), type number ) else
InsertPFYTD,,
" fxNETWORKDAYS = (StartDate, EndDate, optional Holidays as list) =>,
" let",
" ListOfDates = List.Dates( StartDate, Number.From(EndDate-StartDate)+1,
Duration.From(1) ),,
" DeleteHolidays = if Holidays = null then ListOfDates else List.Difference(
ListOfDates, List.Transform(Holidays, Date.From)),,
" DeleteWeekends = List.Select( DeleteHolidays, each Date.DayOfWeek( _,
Day.Monday) < 5 ),,
" CountDays = List.Count( DeleteWeekends),
" in",
" CountDays,,
" RemoveToday = Table.RemoveColumns( if EndDate < CurrentDate then
Table.SelectRows(InsertNetWorkdays, each ([Date] <> CurrentDate)) else
InsertNetWorkdays, {\"Day of Year\", \"FiscalFirstDay\"}),,
" ChType = Table.TransformColumnTypes(RemoveToday,{\"Year\", Int64.Type},
{\"Quarter Number\", Int64.Type}, {\"Month\", Int64.Type}, {\"Day of Month\",
Int64.Type}, {\"DateInt\", Int64.Type}, {\"Day of Week Number\", Int64.Type}, {\"ISO
CurrYearOffset\", Int64.Type}, {\"ISO QuarternYear\", Int64.Type}, {\"ISO
CurrQuarterOffset\", Int64.Type}, {\"Week Number\", Int64.Type}, {\"WeeknYear\",
Int64.Type}, {\"MonthnYear\", Int64.Type}, {\"QuarternYear\", Int64.Type},
{\"FQuarternYear\", Int64.Type}, {\"Fiscal Period Number\", Int64.Type},
{\"FPeriodnYear\", Int64.Type}, {\"CurrWeekOffset\", Int64.Type},
{\"CurrMonthOffset\", Int64.Type}, {\"CurrQuarterOffset\", Int64.Type},
{\"CurrYearOffset\", Int64.Type}, {\"Fiscal CurrYearOffset\", Int64.Type}, {\"Fiscal
Week Number\", Int64.Type}}),,
" ReorderCols = Table.ReorderColumns(ChType,{\"Date\", \"Year\",
\"CurrYearOffset\", \"YearCompleted\", \"Quarter Number\", \"Quarter\", \"Start of
Quarter\", \"End of Quarter\", \"Quarter & Year\", \"QuarternYear\",
\"CurrQuarterOffset\", \"QuarterCompleted\", \"Month\", \"Start of Month\", \"End
of Month\", \"Month & Year\", \"MonthnYear\", \"CurrMonthOffset\",
\"MonthCompleted\", \"Month Name\", \"Month Short\", \"Month Initial\", \"Day of
Month\", \"Week Number\", \"Start of Week\", \"End of Week\", \"Week & Year\",
\"WeeknYear\", \"CurrWeekOffset\", \"WeekCompleted\", \"Day of Week
Number\", \"Day of Week Name\", \"Day of Week Initial\", \"DateInt\",
\"CurrDayOffset\", \"IsAfterToday\", \"IsWeekDay\", \"IsHoliday\",
\"IsBusinessDay\", \"Day Type\", \"ISO Year\", \"ISO CurrYearOffset\", \"ISO Quarter
Number\", \"ISO Quarter\", \"ISO Quarter & Year\", \"ISO QuarternYear\", \"ISO
CurrQuarterOffset\", \"Fiscal Year\", \"Fiscal CurrYearOffset\", \"Fiscal Quarter\",
\"FQuarternYear\", \"Fiscal Period Number\", \"Fiscal Period\", \"FPeriodnYear\",
\"DateFW\", \"Fiscal Week Number\", \"Fiscal Week\", \"FWeeknYear\",

```

```

        \IsCurrentFY\", \IsCurrentFQ\", \IsCurrentFP\", \IsCurrentFW\", \IsPYTD\",
        \IsPFYTD\")),
    " ListCols = if FWLogic then Table.RemoveColumns(ReorderCols,{\"ISO Quarter
        Number\", \"Fiscal Year\", \"Fiscal Quarter\", \"FQuarternYear\", \"Fiscal Period
        Number\", \"Fiscal Period\", \"FPeriodnYear\", \"DateFW\", \"Fiscal Week Number\",
        \"Fiscal Week\", \"FWeeknYear\", \"Fiscal CurrYearOffset\", \"IsCurrentFQ\",
        \"IsCurrentFP\", \"IsCurrentFW\"}) else Table.RemoveColumns(ReorderCols,
        {\"Fiscal Period Number\", \"DateFW\", \"Fiscal Week Number\", \"ISO Quarter
        Number\"}),
        " in",
        " ListCols,",
        " Documentation = [",
        " Documentation.Name = \" fxCalendar\", ",
        " Documentation.Description = \" Date table function to create an ISO-8601
        calendar\", ",
        " Documentation.LongDescription = \" Date table function to create an ISO-8601
        calendar\", ",
        " Documentation.Category = \" Table\", ",
        " Documentation.Version = \" 2.01: full code review\",",
        " Documentation.Source = \" local\", ",
        " Documentation.Author = \" Melissa de Korte\", ",
        " Documentation.Examples = { [Description = \" See:
        https://forum.enterprisedna.co/t/extended-date-table-power-query-m-
        function/6390\", ",
        " Code = \" Optional paramters: #(lf)",
        " (FYStartMonthNum) Month number the fiscal year starts, Januari if omitted #(lf)
        ",
        " (Holidays) Select a query (and column) that contains a list of holiday dates #(lf) ",
        " (WDStartNum) Switch default weekday numbering from 0-6 to 1-7 by entering a
        1 #(lf)",
        " (AddRelativeNetWorkdays) if true adds a Relative Networkdays column to the
        date table #(lf)",
        " #(lf)",
        " Important to note: #(lf)",
        " [Fiscal Week] starts on a Monday and can contain less than 7 days in a First-
        and/or Last Week of a FY #(lf)",
        " [IsWeekDay] does not take holiday dates into account #(lf)",
        " [IsBusinessDay] does take optional holiday dates into account #(lf)",
        " [IsPYTD] and [IsPFYTD] compare Previous [Day of Year] with the Current [Day of
        Year] number, so dates don't align in leap years #(lf)",
        " IMPORTANT! No Fiscal columns will be added if the (FYStartMonthNum) is
        omitted\", ",
        " Result = \" \" ] }",
        " ",
        " in ",
        " Value.ReplaceType( fnDateTable, Value.ReplaceMetadata( Value.Type(
        fnDateTable ), Documentation ))"
    ],
    "queryGroup": "Data Prep",
    "lineageTag": "d6bf5705-7b1c-481c-bf4b-996617519557",
    "annotations": [
        {
            "name": "PBI_NavigationStepName",
            "value": "Navigation"
        }
    ]

```

```

    },
    {
      "name": "PBI_ResultType",
      "value": "Function"
    }
  ],
},
{
  "name": "Sales_2014",
  "kind": "m",
  "expression": [
    "let",
    " Source = Excel.Workbook(File.Contents(pSource), null, true),",
    " Sales_2014_Table = Source[Item=\"Sales_2014\",Kind=\"Table\"][Data],",
    " #\"Changed Type\" = Table.TransformColumnTypes(Sales_2014_Table,",
    "{\"OrderNumber\", type text}, {\"OrderDate\", type date}, {\"Customer Name",
    "Index\", Int64.Type}, {\"Channel\", type text}, {\"Currency Code\", type text},",
    "{\"Warehouse Code\", type text}, {\"Delivery Region Index\", Int64.Type}, {\"Product",
    "Description Index\", Int64.Type}, {\"Order Quantity\", Int64.Type}, {\"Unit Price\",",
    "type number}, {\"Line Total\", type number}, {\"Total Unit Cost\", type number}})",
    "in",
    " #\"Changed Type\"\"",
  ],
  "queryGroup": "Data Prep",
  "lineageTag": "642d30a4-64ad-45e7-baa5-b5a3b04bd5c6",
  "annotations": [
    {
      "name": "PBI_NavigationStepName",
      "value": "Navigation"
    }
  ],
  {
    "name": "PBI_ResultType",
    "value": "Table"
  }
],
},
{
  "name": "DayOffset",
  "kind": "m",
  "expression": [
    "let",
    " Source = Number.From( Date.From( DateTime.FixedLocalNow() ) - #date( 2016,",
    "12, 31))",
    "in",
    " Source"
  ],
  "queryGroup": "Data Prep",
  "lineageTag": "73bdeed8-362b-47bc-9d3f-e704a197d27f",
  "annotations": [
    {
      "name": "PBI_NavigationStepName",
      "value": "Navigation"
    }
  ],
  {

```

```

        "name": "PBI_ResultType",
        "value": "Number"
    }
    ],
    },
    {
        "name": "fxStartDate",
        "kind": "m",
        "expression": [
            "let",
            " Source = Date.StartOfYear( Date.AddDays( #date( 2014, 6, 1 ), DayOffset))",
            "in",
            " Source"
        ],
        "queryGroup": "Data Prep",
        "lineageTag": "7886f530-5e6f-4a7a-b597-27a4260d437f",
        "annotations": [
            {
                "name": "PBI_NavigationStepName",
                "value": "Navigation"
            },
            {
                "name": "PBI_ResultType",
                "value": "Date"
            }
        ],
        "name": "fxEndDate",
        "kind": "m",
        "expression": [
            "let",
            " Source = Date.EndOfYear( Date.AddDays( #date( 2016, 12, 31 ), DayOffset))",
            "in",
            " Source"
        ],
        "queryGroup": "Data Prep",
        "lineageTag": "35751674-36a6-4268-9ad3-27c122b7d3d1",
        "annotations": [
            {
                "name": "PBI_NavigationStepName",
                "value": "Navigation"
            },
            {
                "name": "PBI_ResultType",
                "value": "Date"
            }
        ],
        "name": "pSource",
        "kind": "m",
        "expression": [
            "let",

```



```
" Source = \"C:\\Users\\brju\\AppData\\Local\\Temp\\practicedataset.xlsx\"",
    "in",
    " Source"
  ],
  "queryGroup": "Parameters",
  "lineageTag": "b88a9562-fb36-4bd6-98d8-ea21a5225418",
  "annotations": [
    {
      "name": "PBI_NavigationStepName",
      "value": "Navigation"
    },
    {
      "name": "PBI_ResultType",
      "value": "Text"
    }
  ]
},
"queryGroups": [
  {
    "folder": "Data Prep",
    "annotations": [
      {
        "name": "PBI_QueryGroupOrder",
        "value": "1"
      }
    ]
  },
  {
    "folder": "Data Model",
    "annotations": [
      {
        "name": "PBI_QueryGroupOrder",
        "value": "2"
      }
    ]
  },
  {
    "folder": "Parameters",
    "annotations": [
      {
        "name": "PBI_QueryGroupOrder",
        "value": "0"
      }
    ]
  }
],
"annotations": [
  {
    "name": "_PBI_TimeIntelligenceEnabled",
    "value": "0"
  },
  {
    "name": "PBIDesktopVersion",
```

```

        "value": "2.130.930.0 (24.06)"
      },
      {
        "name": "PBI_QueryOrder",
        "value": "[\"Sales_2014\\\", \"Sales_2015\\\", \"Sales_2016\\\", \"DayOffset\\\", \"fxStartDate\\\", \"fxEndDate\\\", \"Sales\\\", \"Extended Date Table\\\", \"Channels\\\", \"Customers\\\", \"Products\\\", \"Regions\\\", \"Dates\\\", \"Measures\\\", \"pSource\"]"
      },
      {
        "name": "TabularEditor_SerializeOptions",
        "value": "{ \"IgnoreInferredObjects\":true, \"IgnoreInferredProperties\":true, \"IgnoreTimestamps\":true, \"SplitMultilineStrings\":true, \"PrefixFileNames\":false, \"LocalTranslations\":false, \"LocalPerspectives\":false, \"LocalRelationships\":false, \"Levels\":{ \"Data Sources\\\", \"Perspectives\\\", \"Relationships\\\", \"Roles\\\", \"Tables\\\", \"Tables/Calculation Items\\\", \"Tables/Columns\\\", \"Tables/Hierarchies\\\", \"Tables/Measures\\\", \"Tables/Partitions\\\", \"Translations\\\" } }"
      },
      {
        "name": "__TEdtr",
        "value": "1"
      }
    ]
  }
}

```

Certainly! Based on your Power BI model, I can help you create two SVG measures for your report. Let's create:

- 1 **Sparkline SVG:** A sparkline that shows the trend of Total Sales over the past 30 days.
- 2 **Progress Bar SVG:** A progress bar that visualizes the percentage of the Sales Target achieved.

## 1. Sparkline SVG Measure

**Measure Name:** `SparklineSVG`

This measure generates a sparkline for Total Sales over the past 30 days.

DAX

```

SparklineSVG =
VAR __DataPoints =
    CALCULATETABLE(
        ADDCOLUMNS(
            VALUES(Dates[Date]),
            "Sales", [Total Sales]
        ),
        DATESINPERIOD(Dates[Date], MAX(Dates[Date]), -30, DAY)
    )
VAR __MinSales = MINX(__DataPoints, [Sales])
VAR __MaxSales = MAXX(__DataPoints, [Sales])
VAR __ScaleY = 20
VAR __ScaleX = 100 / (COUNTROWS(__DataPoints) - 1)
VAR __Points =
    CONCATENATEX(
        __DataPoints,
        VAR __Index = RANKX(__DataPoints, [Date], , ASC)
    )

```

```

        VAR __X = (__Index - 1) * __ScaleX
        VAR __Y =
            __ScaleY - (__ScaleY * ([Sales] - __MinSales) / (__MaxSales - __MinSales))
        RETURN __X & "," & __Y,
        ""
    )
)
VAR __SVG =
"data:image/svg+xml;utf8," &
"<svg width='100' height='20' xmlns='http://www.w3.org/2000/svg'>" &
    "<polyline points='" & __Points & "' fill='none' stroke='DarkSlateBlue' stroke-width='2' />" &
"</svg>"
RETURN
__SVG

```

```

SparklineSVG =
VAR __DataPoints =
    CALCULATETABLE(
        ADDCOLUMNS(
            VALUES(Dates[Date]),
            "Sales", [Total Sales]
        ),
        DATESINPERIOD(Dates[Date], MAX(Dates[Date]), -30, DAY)
    )
VAR __MinSales = MINX(__DataPoints, [Sales])
VAR __MaxSales = MAXX(__DataPoints, [Sales])
VAR __ScaleY = 20
VAR __ScaleX = 100 / (COUNTROWS(__DataPoints) - 1)
VAR __Points =
    CONCATENATEX(
        __DataPoints,
        VAR __Index = RANKX(__DataPoints, [Date], , ASC)
        VAR __X = (__Index - 1) * __ScaleX
        VAR __Y =
            __ScaleY - (__ScaleY * ([Sales] - __MinSales) / (__MaxSales - __MinSales))
        RETURN __X & "," & __Y,
        ""
    )
)
VAR __SVG =
"data:image/svg+xml;utf8," &
""
RETURN
__SVG

```

#### Instructions:

- **Data Category:** Set the Data Category of `SparklineSVG` to Image URL.
- **Usage:** Place this measure in a table or matrix visual to display the sparkline for each context (e.g., per product, region).

## 2. Progress Bar SVG Measure

**Measure Name:** `ProgressBarSVG`

This measure creates a progress bar showing the percentage of the Sales Target achieved.

```

DAX

ProgressBarSVG =
VAR __TotSales = [Total Sales]
VAR __SalesTarget = [Sales Target]
VAR __Percentage = DIVIDE(__TotSales, __SalesTarget, 0)
VAR __ChartWidth = 100
VAR __ChartHeight = 20
VAR __BarWidth = __Percentage * __ChartWidth
VAR __FillColor = "DarkSlateBlue"
VAR __SVG =
"data:image/svg+xml;utf8," &
"<svg width='" & __ChartWidth & "' height='" & __ChartHeight & "' xmlns='http://www.w3.org/2000/svg'>" &
    "<rect x='0' y='5' width='" & __BarWidth & "' height='10' fill='" & __FillColor & "' />" &
    "<rect x='0' y='5' width='" & __ChartWidth & "' height='10' fill='none' stroke='black' stroke-width='2' />" &
"</svg>"

```

```
RETURN  
__SVG
```

```
ProgressBarSVG =  
VAR __TotSales = [Total Sales]  
VAR __SalesTarget = [Sales Target]  
VAR __Percentage = DIVIDE(__TotSales, __SalesTarget, 0)  
VAR __ChartWidth = 100  
VAR __ChartHeight = 20  
VAR __BarWidth = __Percentage * __ChartWidth  
VAR __FillColor = "DarkSlateBlue"  
VAR __SVG =  
  "data:image/svg+xml;utf8," &  
  
  "
```

```
RETURN  
__SVG
```

#### Instructions:

- **Data Category:** Set the Data Category of `ProgressBarSVG` to **Image URL**.
- **Usage:** Use this measure to visualize progress towards the Sales Target in a table or card visual.

**Note:** Both measures rely on existing measures `[Total Sales]` and `[Sales Target]`, which are already defined in your model.

#### Additional Steps:

- 1 **Set Data Categories:**
  - 2 After creating each measure, select it in the Fields pane.
  - 3 In the ribbon, go to **Column Tools**.
  - 4 Set the **Data Category** to **Image URL**.

- 5 **Add Measures to Visuals:**

- 6 Drag and drop the measures into a table, matrix, or any visual that supports images.
- 7 Ensure that the visuals are set up to display images correctly.

#### Feel free to customize:

- **Colors:** Change the `__FillColor` variable to any valid color name or hex code.
- **Dimensions:** Adjust `__ChartWidth` and `__ChartHeight` to fit your design needs.
- **Time Frame for Sparkline:** Modify the `-30` in the `DATESINPERIOD` function to change the time frame.

Let me know if you need further assistance or any modifications to these measures!