

o1 Testing on Problems 1 and 2

Here is a bim file from a Power BI report. Read the bim file and tell me what you know about the data model

```
{
  "name": "How Good is ChatGPT o1 Strawberry at DAX",
  "compatibilityLevel": 1567,
  "model": {
    "culture": "en-US",
    "dataAccessOptions": {
      "legacyRedirects": true,
      "returnErrorValuesAsNull": true
    },
    "defaultPowerBIDataSourceVersion": "powerBI_V3",
    "sourceQueryCulture": "en-US",
    "tables": [
      {
        "name": "Problem1",
        "lineageTag": "a00dfaa1-c25a-4306-a2c1-c81520b2beaf",
        "columns": [
          {
            "name": "Date",
            "dataType": "dateTime",
            "sourceColumn": "Date",
            "formatString": "Short Date",
            "lineageTag": "ec40d327-594f-4c4f-af2e-29e9798fe20b",
            "summarizeBy": "none",
            "annotations": [
              {
                "name": "SummarizationSetBy",
                "value": "Automatic"
              },
              {
                "name": "UnderlyingDateTimeDataType",
                "value": "Date"
              }
            ]
          }
        ],
        "name": "SalesTarget",
        "dataType": "int64",
        "sourceColumn": "SalesTarget",
        "formatString": "0",
        "lineageTag": "85f905ae-7ef7-454e-ad8a-90789bcbcb690",
        "summarizeBy": "none",
        "annotations": [
          {
            "name": "SummarizationSetBy",
```

```

"value": "User"
}
],
{
"name": "IncrementalCost",
"dataType": "int64",
"sourceColumn": "IncrementalCost",
"formatString": "0",
"lineageTag": "27f67fe4-2985-41ca-947a-9bda9ec6b2e1",
"summarizeBy": "none",
"changedProperties": [
{
"property": "IsHidden"
}
],
"annotations": [
{
"name": "SummarizationSetBy",
"value": "User"
}
],
{
"name": "TargetProfit",
"dataType": "int64",
"sourceColumn": "TargetProfit",
"formatString": "0",
"lineageTag": "13cfd617-30cb-4dc9-acdd-88a56a90a043",
"summarizeBy": "none",
"annotations": [
{
"name": "SummarizationSetBy",
"value": "User"
}
],
"partitions": [
{
"name": "Problem1",
"mode": "import",
"source": {
"type": "m",
"expression": [
"let",
" Source =
Table.FromRows(Json.Document(Binary.Decompress(Binary.FromText(\"i45WstQ
31TcyMDJR0IEyNTAAkkYgwsTCQCIWByRrhk3WDCZralRN2gQubYxNGkjGxgl
A\", BinaryEncoding.Base64), Compression.Deflate)), let _t = ((type nullable text)
meta [Serialized.Text = true]) in type table [Date = _t, SalesTarget = _t,
IncrementalCost = _t, TargetProfit = _t]),",
"#\"Changed Type\" = Table.TransformColumnTypes(Source,{\"Date\", Date.Type},
{\"SalesTarget\", Int64.Type}, {\"IncrementalCost\", Int64.Type}, {\"TargetProfit\",

```

```

Int64.Type})),
"in",
"#\"Changed Type\"
]
}
},
"annotations": [
{
"name": "PBI_NavigationStepName",
"value": "Navigation"
},
{
"name": "PBI_ResultType",
"value": "Table"
}
],
{
"name": "Key Measures",
"lineageTag": "b515adfd-29bc-41e7-9006-b697d2dce607",
"columns": [
{
"name": "Column1",
"dataType": "string",
"sourceColumn": "Column1",
"lineageTag": "851ad852-e63a-4d65-968e-2f2ea72124cd",
"summarizeBy": "none",
"annotations": [
{
"name": "SummarizationSetBy",
"value": "Automatic"
}
]
}
],
"partitions": [
{
"name": "Key Measures",
"mode": "import",
"source": {
"type": "m",
"expression": [
"let",
" Source =
Table.FromRows(Json.Document(Binary.Decompress(Binary.FromText(\"i44FAA=
=\", BinaryEncoding.Base64), Compression.Deflate)), let _t = ((type nullable text)
meta [Serialized.Text = true]) in type table [Column1 = _t])",
"in",
" Source"
]
}
}
],

```

```

"measures": [
{
"name": "Problem 1 Brian",
"expression": [
""
,
""
,
"VAR __CurrDate = MAX(Problem1[Date] )",
""
,
"VAR __CurrCol1 = MAX(Problem1[SalesTarget] )",
""
,
"VAR __vTable = ",
" FILTER( ",
" ALL( Problem1) )",
" Problem1[Date] <= __CurrDate",
" )",
""
,
"VAR CumullIncrCos5 = ",
"SUMX( ",
" __vTable, ",
" [Total Incr Cost]",
" )",
""
,
"VAR __Result = ",
" __CurrCol1 - CumullIncrCos5",
""
,
"RETURN __Result"
],
"formatString": "0",
"lineageTag": "6872483a-d224-47a0-9c83-ff0c51e11d7a"
},
{
"name": "Total Incr Cost",
"expression": [
""
,
"SUM( Problem1[IncrementalCost] )"
],
"formatString": "0",
"lineageTag": "ee292f53-113d-4bf8-a7db-0de07cc6f7db"
},
{
"name": "Problem2 Brian",
"expression": [
""
,
"VAR __CurrHRID = MAX( Problem2[HrActionID] )",
"VAR __CurrStepID = MAX( Problem2[ProcessStepID] )",
"VAR __CurrDate = MAX( Problem2[ProcessStepCompletionDate] )",
""
,
"VAR __MinStepInID = ",
"MINX( ",
" FILTER( ",
" ALL( Problem2[HrActionID], Problem2[ProcessStepID] ) )",
" Problem2[HrActionID] - __CurrHRID ",
" )",
" Problem2[ProcessStepID]",

```

```

    )",
    "",
    "VAR __PrevDate =",
    "MAXX( ",
    " FILTER(",
    " ALL( Problem2[HrActionID], Problem2[ProcessStepID],
    Problem2[ProcessStepCompletionDate]),",
    " Problem2[HrActionID] = __CurrHRID &&",
    " Problem2[ProcessStepID] < __CurrStepID ",
    "), ",
    " Problem2[ProcessStepCompletionDate]",
    ")",
    "",
    "VAR __Result =",
    "IF(",
    " __CurrStepID = __MinStepInID,",
    " 1,",
    " DATEDIFF( ",
    " __PrevDate,",
    " __CurrDate,",
    " DAY",
    " )",
    ")",
    ")",
    "",
    "RETURN __Result",
    ""

],
"formatString": "0",
"lineageTag": "bdfc6c39-eef8-4812-9023-28e2e9347db6"
},
{
"name": "GPT-o1 Solution Prob2",
"expression": [
    "",
    "",
    "VAR currentHrActionID = SELECTEDVALUE(Problem2[HrActionID])",
    "VAR currentCompletionDate =
    SELECTEDVALUE(Problem2[ProcessStepCompletionDate])",
    "VAR currentProcessStepID = SELECTEDVALUE(Problem2[ProcessStepID])",
    "",
    "RETURN",
    "IF(",
    " ISBLANK(currentCompletionDate)",
    " BLANK()",
    " VAR previousStepCompletionDate =",
    " MAXX(",
    " FILTER(",
    " ALL(Problem2)",
    " Problem2[HrActionID] = currentHrActionID",
    " && Problem2[ProcessStepCompletionDate] <= currentCompletionDate",
    " && Problem2[ProcessStepID] < currentProcessStepID",
    " ),",
    " Problem2[ProcessStepCompletionDate]",
    " )",

```

```

" VAR firstCompletionDate =",
" MINX(",
" FILTER(",
" ALL(Problem2),",
" Problem2[HrActionID] = currentHrActionID",
" ),",
" Problem2[ProcessStepCompletionDate]",
" )",
" VAR firstProcessStepID =",
" MINX(",
" FILTER(",
" ALL(Problem2),",
" Problem2[HrActionID] = currentHrActionID",
" && Problem2[ProcessStepCompletionDate] = firstCompletionDate",
" ),",
" Problem2[ProcessStepID]",
" )",
" RETURN",
" IF(",
" currentCompletionDate = firstCompletionDate",
" && currentProcessStepID = firstProcessStepID",
" 1",
" IF(",
" ISBLANK(previousStepCompletionDate)",
" 0",
" VAR daysBetween = DATEDIFF(previousStepCompletionDate,
currentCompletionDate, DAY)",
" RETURN",
" daysBetween",
" )",
" )",
" )",
""

],
"formatString": "0",
"lineageTag": "a77892c2-8e0b-4aea-b954-6cd934f37ae8"
}
],
"annotations": [
{
"name": "PBI_ResultType",
"value": "Table"
}
],
{
"name": "Dates",
"lineageTag": "dc123555-bde3-477b-b0f9-3b2826c2cfef",
"dataCategory": "Time",
"columns": [
{
"name": "FYWeekNum",
"dataType": "string",
"sourceColumn": "FYWeekNum",

```

```
"lineageTag": "148d9816-839f-44a6-876d-b2a82d1fe481",
"summarizeBy": "none",
"annotations": [
  {
    "name": "SummarizationSetBy",
    "value": "Automatic"
  }
],
{
  "name": "Date",
  "dataType": "dateTime",
  "isKey": true,
  "sourceColumn": "Date",
  "formatString": "Long Date",
  "lineageTag": "25ab1272-613e-44bb-8e92-ae4f0e14951d",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    },
    {
      "name": "UnderlyingDateTimeDataType",
      "value": "Date"
    }
  ],
  {
    "name": "Fiscal Year",
    "dataType": "string",
    "sourceColumn": "Fiscal Year",
    "lineageTag": "68cb3607-d756-4144-a649-3af4b3feb508",
    "summarizeBy": "none",
    "annotations": [
      {
        "name": "SummarizationSetBy",
        "value": "Automatic"
      }
    ],
    {
      "name": "Fiscal CurrYearOffset",
      "dataType": "double",
      "sourceColumn": "Fiscal CurrYearOffset",
      "lineageTag": "6c9bc36b-eeb1-4d39-b04a-4ec35825af00",
      "summarizeBy": "none",
      "annotations": [
        {
          "name": "SummarizationSetBy",
          "value": "Automatic"
        },
        {
          "name": "PBI_FormatHint",
```

```
"value": "{\\isGeneralNumber\\":true}"
}
],
{
  "name": "Fiscal Week",
  "dataType": "string",
  "sourceColumn": "Fiscal Week",
  "lineageTag": "0bd101c3-1f23-4edf-b29a-f90e6f90fb1a",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ]
},
{
  "name": "Index",
  "dataType": "int64",
  "sourceColumn": "Index",
  "formatString": "0",
  "lineageTag": "4578a6a8-cad3-41bb-b731-9e2c9dfe5a5c",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ]
},
{
  "name": "OffsetFW",
  "dataType": "int64",
  "sourceColumn": "OffsetFW",
  "formatString": "0",
  "lineageTag": "30f89ea3-8029-4351-a00b-ec3ec0a7a693",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ]
},
],
"partitions": [
{
  "name": "Dates",
  "mode": "import",
  "source": {
    "type": "m",
    "expression": [
      "let",
```



```

" Source = #\"Extended Date Table\"(fxStartDate, fxEndDate, 10, null, null),",
" Add13MonYearOffset = Table.AddColumn(Source, \"Offset13MonYear\", each [\",
\" a = Date.From( DateTime.FixedLocalNow()),\",
\" b = Number.IntegerDivide(Number.From(a), 364),\",
\" c = Number.IntegerDivide(Number.From([Date]), 364),\",
\" d = c - b\",
\"][d]),\",
\" Add13MonYear = Table.AddColumn(Add13MonYearOffset, \"13MonYear\", each
Date.Year( DateTime.FixedLocalNow()) + [Offset13MonYear]),\",
\" #\"Changed Type\" = Table.TransformColumnTypes(Add13MonYear,
{{\"13MonYear\", Int64.Type}, {\"Offset13MonYear\", Int64.Type}}),\",
\" #\"Removed Other Columns\" = Table.SelectColumns(#\"Changed Type\",
{\"Date\", \"Fiscal Year\", \"Fiscal CurrYearOffset\", \"Fiscal Week\"}),\",
\" #\"Added Custom Column\" = Table.AddColumn(#\"Removed Other Columns\",
\"FYWeekNum\", each Number.From( Text.End( [Fiscal Week], 4) & Text.Middle(
[Fiscal Week],2,2))),\",
\" #\"Sorted Rows\" = Table.Sort(#\"Added Custom Column\",{{\"Date\",
Order.Ascending}}),\",
\" #\"Grouped Rows\" = Table.Group(#\"Sorted Rows\", {\"FYWeekNum\"}, {{\"All\",
each _, type table [Date=date, Fiscal Year=text, Fiscal CurrYearOffset=nullable
number, Fiscal Week=text, FYWeekNum=nullable number]})),\",
\" #\"Added Index\" = Table.AddIndexColumn(#\"Grouped Rows\", \"Index\", 0, 1,
Int64.Type),\",
\" ExpandAll = Table.ExpandTableColumn(#\"Added Index\", \"All\", {\"Date\", \"Fiscal
Year\", \"Fiscal CurrYearOffset\", \"Fiscal Week\"}, {\"Date\", \"Fiscal Year\", \"Fiscal
CurrYearOffset\", \"Fiscal Week\"}),\",
\" AddFWOffset = Table.AddColumn(ExpandAll, \"OffsetFW\", each [Index] -
#\"Today Index\", Int64.Type),
\"in\",
\" AddFWOffset\"
]
}
},
],
\"annotations\": [
{
\"name\": \"PBI_ResultType\",
\"value\": \"Table\"
},
{
\"name\": \"PBI_NavigationStepName\",
\"value\": \"Navigation\"
}
],
{
\"name\": \"Today Index\",
\"lineageTag\": \"4fe49beeb-0480-4d0e-82c0-1d1014e936dd\",
\"columns\": [
{
\"name\": \"Today Index\",
\"dataType\": \"double\",
\"sourceColumn\": \"Today Index\",
\"lineageTag\": \"4cc74519-3fc1-4e9b-bd26-3a4183c7dc4a\",

```

```

"summarizeBy": "none",
"annotations": [
{
"name": "SummarizationSetBy",
"value": "Automatic"
},
{
"name": "PBI_FormatHint",
"value": "{\isGeneralNumber\":true}"
}
]
},
"partitions": [
{
"name": "Today Index",
"mode": "import",
"source": {
"type": "m",
"expression": [
"let",
" Source = #\"Extended Date Table\"(fxStartDate, fxEndDate, 10, null, null),",
" Add13MonYearOffset = Table.AddColumn(Source, \"Offset13MonYear\", each [",
" a = Date.From( DateTime.FixedLocalNow()),",
" b = Number.IntegerDivide(Number.From(a), 364),",
" c = Number.IntegerDivide(Number.From([Date]), 364),",
" d = c - b",
" ][d]),",
" Add13MonYear = Table.AddColumn(Add13MonYearOffset, \"13MonYear\", each",
Date.Year( DateTime.FixedLocalNow()) + [Offset13MonYear]),",
" #\"Changed Type\" = Table.TransformColumnTypes(Add13MonYear,",
{{\"13MonYear\", Int64.Type}, {\"Offset13MonYear\", Int64.Type}}),",
" #\"Removed Other Columns\" = Table.SelectColumns(#\"Changed Type\",",
{\"Date\", \"Fiscal Year\", \"Fiscal CurrYearOffset\", \"Fiscal Week\"}),",
" #\"Added Custom Column\" = Table.AddColumn(#\"Removed Other Columns\",",
\"FYWeekNum\", each Number.From( Text.End( [Fiscal Week], 4) & Text.Middle(",
[Fiscal Week],2,2))),",
" #\"Sorted Rows\" = Table.Sort(#\"Added Custom Column\",{{\"Date\",",
Order.Ascending}}),",
" #\"Grouped Rows\" = Table.Group(#\"Sorted Rows\", {\"FYWeekNum\"}, {{\"All\",",
each _, type table [Date=date, Fiscal Year=text, Fiscal CurrYearOffset=nullable",
number, Fiscal Week=text, FYWeekNum=nullable number]})),",
" #\"Added Index\" = Table.AddIndexColumn(#\"Grouped Rows\", \"Index\", 0, 1,",
Int64.Type),",
" ExpandAll = Table.ExpandTableColumn(#\"Added Index\", \"All\", {\"Date\", \"Fiscal",
Year\", \"Fiscal CurrYearOffset\", \"Fiscal Week\"}, {\"Date\", \"Fiscal Year\", \"Fiscal",
CurrYearOffset\", \"Fiscal Week\"}),",
" Filter = Table.SelectRows(ExpandAll, each [Date] = Today),",
" TodayIdx = List.Max(Filter[Index]),",
"in",
" TodayIdx"
]
}
}

```

```

],
"annotations": [
{
"name": "PBI_ResultType",
"value": "Number"
}
],
},
{
"name": "Today",
"lineageTag": "3c92b342-1f6f-4e65-81f5-136e31e236b1",
"columns": [
{
"name": "Today",
"dataType": "dateTime",
"sourceColumn": "Today",
"formatString": "Long Date",
"lineageTag": "30d294ab-7169-45cc-9ac1-070fbb1ca2e6",
"summarizeBy": "none",
"annotations": [
{
"name": "SummarizationSetBy",
"value": "Automatic"
}
],
{
"name": "UnderlyingDateTimeDataType",
"value": "Date"
}
]
}
],
"partitions": [
{
"name": "Today",
"mode": "import",
"source": {
"type": "m",
"expression": [
"let",
" Source = DateTime.FixedLocalNow(",
" ",
"),",
" #\"Converted to Table\" = #table(1, {{Source}}),",
" #\"Renamed Columns\" = Table.RenameColumns(#\"Converted to Table\",",
" {{\"Column1\", \"Date\"}}),",
" #\"Split Column by Delimiter\" =",
" Table.SplitColumn(Table.TransformColumnTypes(#\"Renamed Columns\",",
" {{\"Date\", type text}, \"en-US\"), \"Date\", Splitter.SplitTextByEachDelimiter({\" \",",
" QuoteStyle.Csv, false}, {\"Date\", \"Time\"})),",
" Chgtype = Table.TransformColumnTypes(#\"Split Column by Delimiter\",",
" {{\"Date\", type date}, {\"Time\", type time}}),",
" Today = List.Max( Chgtype[Date]),",
"in",
" Today"

```

```
]
}
}
],
"annotations": [
{
"name": "PBI_NavigationStepName",
"value": "Navigation"
},
{
"name": "PBI_ResultType",
"value": "Date"
}
],
},
{
"name": "Problem2",
"lineageTag": "b4077b8c-dc7d-49eb-b769-f218a172cd05",
"columns": [
{
"name": "Index",
"dataType": "int64",
"sourceColumn": "Index",
"formatString": "0",
"lineageTag": "48a42532-dd24-4a89-919a-70b1c69b91c3",
"summarizeBy": "none",
"annotations": [
{
"name": "SummarizationSetBy",
"value": "Automatic"
}
]
},
{
"name": "HrActionID",
"dataType": "int64",
"sourceColumn": "HrActionID",
"formatString": "0",
"lineageTag": "078a5350-a8e7-4cb0-8126-7d593e49b76e",
"summarizeBy": "none",
"annotations": [
{
"name": "SummarizationSetBy",
"value": "Automatic"
}
]
},
{
"name": "ProcessStepID",
"dataType": "int64",
"sourceColumn": "ProcessStepID",
"formatString": "0",
"lineageTag": "f0c36224-28f9-4ce4-8556-c9c58d30d699",
"summarizeBy": "none",
```

```
"annotations": [
  {
    "name": "SummarizationSetBy",
    "value": "Automatic"
  }
],
{
  "name": "ProcessStepCompletionDate",
  "dataType": "dateTime",
  "sourceColumn": "ProcessStepCompletionDate",
  "formatString": "Short Date",
  "lineageTag": "446c80eb-c7b5-4005-ba73-3d97ab72b6c9",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    },
    {
      "name": "UnderlyingDateTimeDataType",
      "value": "Date"
    }
  ],
},
{
  "name": "DaysBtwSteps",
  "dataType": "int64",
  "sourceColumn": "DaysBtwSteps",
  "formatString": "0",
  "lineageTag": "2acc689e-8336-481a-b508-bf75b2dab626",
  "summarizeBy": "none",
  "annotations": [
    {
      "name": "SummarizationSetBy",
      "value": "Automatic"
    }
  ],
},
"partitions": [
  {
    "name": "Problem2",
    "mode": "import",
    "source": {
      "type": "m",
      "expression": [
        "let",
        "Source =
Table.FromRows(Json.Document(Binary.Decompress(Binary.FromText(\"bdLLasM
wEAXQf/E6IM3ovWzrRbMrdGmyCKkXhTYOsaHt31caWbepCdjYmjnoce1h6Lrd
zXXYDd3LdTqN8/y6jJf9+W38zvXn68NpeZ/O+z4Pbvub8dP0efkYi+yPy5h7/fFnfl
y+Sm+WySkX5dZalwcpDoo1BakXwRBUhFFkC4hSCklMCBeS+64RyyIshNkKLc
AB2PvAA7j7IAD4Apyi1vfSj+iHtR9xjqMbbGL8lp9EybVtDRlKq9BmSbqFIQ8qe
```

```

X5twrHakqifJO5ItlqQuRkIOqKykBwPS1ZEK6fhUHsulkHYrZkncRD2JqJwRymig
DhqrCbrUaANfbUgKsgAay5MxbJh5A/TINI8O3HZ0I9/asjwBqy1A+/\",
BinaryEncoding.Base64), Compression.Deflate)), let _t = ((type nullable text) meta
[Serialized.Text = true]) in type table [Column1 = _t, Column2 = _t, Column3 = _t,
Column4 = _t, Column5 = _t]),",
"#\"Promoted Headers\" = Table.PromoteHeaders(Source,
[PromoteAllScalars=true]),",
"#\"Promoted Headers1\" = Table.PromoteHeaders(#\"Promoted Headers\",
[PromoteAllScalars=true]),",
"#\"Changed Type\" = Table.TransformColumnTypes(#\"Promoted Headers1\",
{{\"ProcessStepIndex\", Int64.Type}, {\"HrActionID\", Int64.Type},
{\"ProcessStepID\", Int64.Type}, {\"ProcessStepCompletionDate\", type date},
{\"DaysBtwSteps\", Int64.Type}}),",
"#\"Renamed Columns\" = Table.RenameColumns(#\"Changed Type\",
{{\"ProcessStepIndex\", \"Index\"}})",
"in",
"#\"Renamed Columns\""
]
}
}
],
"annotations": [
{
"name": "PBI_NavigationStepName",
"value": "Navigation"
},
{
"name": "PBI_ResultType",
"value": "Table"
}
]
},
"relationships": [
{
"name": "43c6b038-ba6f-853e-3127-67b9dbb25d26",
"fromTable": "Problem2",
"fromColumn": "ProcessStepCompletionDate",
"toTable": "Dates",
"toColumn": "Date"
}
],
"cultures": [
{
"name": "en-US",
"linguisticMetadata": {
"content": {
"Version": "3.5.0",
"Language": "en-US",
"Entities": {
"problem_1": {
"Definition": {
"Binding": {
"ConceptualEntity": "Problem1"

```

```
}
},
"State": "Generated",
"Terms": [
{
"problem 1": {
"State": "Generated"
}
},
{
"Problem1": {
"Type": "Noun",
"State": "Generated",
"Weight": 0.99
}
}
],
},
"problem_1.date": {
"Definition": {
"Binding": {
"ConceptualEntity": "Problem1",
"ConceptualProperty": "Date"
}
},
"State": "Generated",
"Terms": [
{
"date": {
"State": "Generated"
}
},
{
"moment": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.736
}
},
{
"period": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.714
}
}
],
"SemanticType": "Time"
```

```
,
"problem_1.sales_target":{
  "Definition":{
    "Binding":{
      "ConceptualEntity": "Problem1",
      "ConceptualProperty": "SalesTarget"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "sales target": {
        "State": "Generated"
      }
    },
    {
      "sale target": {
        "Type": "Noun",
        "State": "Generated",
        "Weight": 0.78
      }
    },
    {
      "SalesTarget": {
        "Type": "Noun",
        "State": "Generated",
        "Weight": 0.99
      }
    },
    {
      "target": {
        "State": "Generated",
        "Weight": 0.97
      }
    },
    {
      "sale focus": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.727
      }
    },
    {
      "sale goal": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.727
      }
    }
  ]
}
```



```
},
{
  "sale objective": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.727
  }
},
{
  "focus": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.714
  }
},
{
  "goal": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.714
  }
},
{
  "objective": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.714
  }
}
],
{
  "problem_1.incremental_cost": {
    "Definition": {
      "Binding": {
        "ConceptualEntity": "Problem1",
        "ConceptualProperty": "IncrementalCost"
      }
    },
    "State": "Generated",
    "Terms": [
      {
        "incremental cost": {
```

```
"State": "Generated"
}
},
{
  "IncrementalCost": {
    "Type": "Noun",
    "State": "Generated",
    "Weight": 0.99
  }
},
{
  "cost": {
    "State": "Generated",
    "Weight": 0.97
  }
},
{
  "incremental price": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  }
},
{
  "incremental charge": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  }
},
{
  "incremental rate": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  }
},
{
  "incremental fee": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  }
}
```

```
}
},
{
  "incremental total": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  }
},
{
  "price": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.476
  }
},
{
  "charge": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.476
  }
},
{
  "rate": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.476
  }
},
{
  "fee": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.476
  }
},
{
  "incremental expenditure": {
```

```
"Type": "Noun",
"State": "Suggested",
"Source": {
  "Agent": "Thesaurus"
},
"Weight": 0.475
}
}
],
},
"problem_1.target_profit": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Problem1",
      "ConceptualProperty": "TargetProfit"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "target profit": {
        "State": "Generated"
      }
    },
    {
      "TargetProfit": {
        "Type": "Noun",
        "State": "Generated",
        "Weight": 0.99
      }
    },
    {
      "profit": {
        "State": "Generated",
        "Weight": 0.97
      }
    },
    {
      "focus profit": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.582
      }
    },
    {
      "goal profit": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },

```

```
"Weight": 0.582
}
},
{
  "objective profit": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.582
  }
},
{
  "target income": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.5
  }
},
{
  "target revenue": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.5
  }
},
{
  "income": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.491
  }
},
{
  "revenue": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.491
  }
},
{
```

```
"target turnover": {
  "Type": "Noun",
  "State": "Suggested",
  "Source": {
    "Agent": "Thesaurus"
  },
  "Weight": 0.485
},
{
  "target return": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  },
  {
    "target yield": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.485
    }
  },
  {
    "key_measure": {
      "Definition": {
        "Binding": {
          "ConceptualEntity": "Key Measures"
        }
      },
      "State": "Generated",
      "Terms": [
        {
          "key measure": {
            "State": "Generated"
          }
        },
        {
          "measure": {
            "State": "Generated",
            "Weight": 0.97
          }
        },
        {
          "key degree": {
            "Type": "Noun",
            "State": "Suggested",
```

```
"Source": {
  "Agent": "Thesaurus"
},
"Weight": 0.5
},
{
  "key quantity": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.5
  },
  {
    "key quota": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.5
    },
    {
      "key extent": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.5
      },
      {
        "degree": {
          "Type": "Noun",
          "State": "Suggested",
          "Source": {
            "Agent": "Thesaurus"
          },
          "Weight": 0.491
        },
        {
          "quota": {
            "Type": "Noun",
            "State": "Suggested",
            "Source": {
              "Agent": "Thesaurus"
            },
            "Weight": 0.491
          }
        }
      }
    }
  }
}
```

```

    }
  },
  {
    "extent": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.491
    }
  },
  {
    "key amount": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.485
    }
  },
  {
    "key portion": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.485
    }
  },
  {
    "key ration": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.485
    }
  }
],
"key_measure.column_1": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Key Measures",
      "ConceptualProperty": "Column1"
    }
  },
  "State": "Generated",
  "Terms": [
    {

```



```
"column 1": {
  "State": "Generated"
},
{
  "Column1": {
    "Type": "Noun",
    "State": "Generated",
    "Weight": 0.99
  }
},
{
  "key_measure.problem_1_brian": {
    "Definition": {
      "Binding": {
        "ConceptualEntity": "Key Measures",
        "ConceptualProperty": "Problem 1 Brian"
      }
    },
    "State": "Generated",
    "Terms": [
      {
        "problem 1 brian": {
          "State": "Generated"
        }
      },
      {
        "brian": {
          "State": "Generated",
          "Weight": 0.97
        }
      },
      {
        "1 brian": {
          "State": "Generated",
          "Weight": 0.97
        }
      },
      {
        "Intended Result": {
          "State": "Suggested",
          "Source": {
            "Type": "External",
            "Agent": "PowerBI.VisualColumnRename"
          },
          "Weight": 0.9
        }
      }
    ],
    "key_measure.total_incr_cost": {
      "Definition": {
        "Binding": {
```

```
"ConceptualEntity": "Key Measures",
"ConceptualProperty": "Total Incr Cost"
},
"State": "Generated",
"Terms": [
{
"total incr cost": {
"State": "Generated"
}
},
{
"aggregate incr cost": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.591
}
},
{
"overall incr cost": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.591
}
},
{
"total incr price": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.492
}
},
{
"total incr charge": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.492
}
},
{
"total incr rate": {
"Type": "Noun",
```

```
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.492
},
{
"total incr fee": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.492
},
{
"total incr total": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.492
},
{
"total incr expenditure": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.482
},
{
"total incr budget": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.468
},
{
"total incr amount": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
```

```

    "Weight": 0.468
  }
}
],
},
"key_measure.Problem2_brian": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Key Measures",
      "ConceptualProperty": "Problem2 Brian"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "Problem2 brian": {
        "State": "Generated"
      }
    }
  ],
},
"key_measure.GPTTo1_solution_Prob2": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Key Measures",
      "ConceptualProperty": "GPT-o1 Solution Prob2"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "GPT-o1 solution Prob2": {
        "State": "Generated"
      }
    },
    {
      "GPT o1 solution Prob2": {
        "State": "Generated",
        "Weight": 0.97
      }
    }
  ],
},
"date": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Dates"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "date": {
        "State": "Generated",

```

```
"Weight": 0.87
}
},
{
  "moment": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.736
  },
  {
    "period": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.714
    }
  }
},
{
  "date.FY_week_num": {
    "Definition": {
      "Binding": {
        "ConceptualEntity": "Dates",
        "ConceptualProperty": "FYWeekNum"
      }
    },
    "State": "Generated",
    "Terms": [
      {
        "FY week num": {
          "State": "Generated"
        }
      },
      {
        "FYWeekNum": {
          "Type": "Noun",
          "State": "Generated",
          "Weight": 0.99
        }
      }
    ],
    {
      "fy week number": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.739
      }
    }
  }
}
```

```

    }
  }
],
},
"date.date": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Dates",
      "ConceptualProperty": "Date"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "date": {
        "State": "Generated"
      }
    },
    {
      "moment": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.736
      }
    },
    {
      "period": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.714
      }
    }
  ],
  "SemanticType": "Time"
},
"date.fiscal_year": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Dates",
      "ConceptualProperty": "Fiscal Year"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "fiscal year": {
        "State": "Generated"
      }
    }
  ]
}

```

```
},
{
  "fiscal yr": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.75
  }
},
{
  "fisc year": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.6
  }
},
{
  "accounting year": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.582
  }
},
{
  "budget year": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.582
  }
},
{
  "fiscal date": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.582
  }
},
{
  "fiscal day": {
    "Type": "Noun",
```

```
"State": "Suggested",
"Source": {
  "Agent": "Thesaurus"
},
"Weight": 0.582
}
},
{
  "tax year": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.582
  }
}
],
},
"date.fiscal_CurrYearOffset": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Dates",
      "ConceptualProperty": "Fiscal CurrYearOffset"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "fiscal CurrYearOffset": {
        "State": "Generated"
      }
    }
  ]
},
"date.fiscal_week": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Dates",
      "ConceptualProperty": "Fiscal Week"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "fiscal week": {
        "State": "Generated"
      }
    }
  ]
},
"date.index": {
  "Definition": {
    "Binding": {
```



```
"ConceptualEntity": "Dates",
"ConceptualProperty": "Index"
},
"State": "Generated",
"Terms": [
{
"index": {
"State": "Generated"
}
},
{
"directory": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.476
}
},
{
"catalogue": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.476
}
},
{
"catalog": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.476
}
},
{
"guide": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.476
}
},
{
"file": {
"Type": "Noun",
```

```

"State": "Suggested",
"Source": {
  "Agent": "Thesaurus"
},
"Weight": 0.476
},
{
  "key": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.476
  },
  {
    "manifestation": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.466
    },
    {
      "pointer": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.466
      },
      {
        "indication": {
          "Type": "Noun",
          "State": "Suggested",
          "Source": {
            "Agent": "Thesaurus"
          },
          "Weight": 0.466
        },
        {
          "indicator": {
            "Type": "Noun",
            "State": "Suggested",
            "Source": {
              "Agent": "Thesaurus"
            },
            {

```

```
"Weight": 0.452
}
}
],
},
"date.offset_FW": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Dates",
      "ConceptualProperty": "OffsetFW"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "offset FW": {
        "State": "Generated"
      }
    },
    {
      "OffsetFW": {
        "Type": "Noun",
        "State": "Generated",
        "Weight": 0.99
      }
    }
  ],
  {
    "FW": {
      "State": "Generated",
      "Weight": 0.97
    }
  }
],
},
"today_index": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Today Index"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "today index": {
        "State": "Generated",
        "Weight": 0.87
      }
    }
  ],
  {
    "today directory": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
```

```
},
"Weight": 0.485
},
{
"today catalogue": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.485
},
{
"today catalog": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.485
},
{
"today guide": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.485
},
{
"today file": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.485
},
{
"today key": {
"Type": "Noun",
"State": "Suggested",
"Source": {
"Agent": "Thesaurus"
},
"Weight": 0.485
},
},
```

```
{
  "today table": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  },
  {
    "today manifestation": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.475
    },
    {
      "today pointer": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.475
      },
      {
        "today indication": {
          "Type": "Noun",
          "State": "Suggested",
          "Source": {
            "Agent": "Thesaurus"
          },
          "Weight": 0.475
        },
        {
          "today_index.today_index": {
            "Definition": {
              "Binding": {
                "ConceptualEntity": "Today Index",
                "ConceptualProperty": "Today Index"
              }
            },
            "State": "Generated",
            "Terms": [
              {
                "today index": {
                  "State": "Generated"
```

```
}  
},  
{  
  "today directory": {  
    "Type": "Noun",  
    "State": "Suggested",  
    "Source": {  
      "Agent": "Thesaurus"  
    },  
    "Weight": 0.485  
  }  
},  
{  
  "today catalogue": {  
    "Type": "Noun",  
    "State": "Suggested",  
    "Source": {  
      "Agent": "Thesaurus"  
    },  
    "Weight": 0.485  
  }  
},  
{  
  "today catalog": {  
    "Type": "Noun",  
    "State": "Suggested",  
    "Source": {  
      "Agent": "Thesaurus"  
    },  
    "Weight": 0.485  
  }  
},  
{  
  "today guide": {  
    "Type": "Noun",  
    "State": "Suggested",  
    "Source": {  
      "Agent": "Thesaurus"  
    },  
    "Weight": 0.485  
  }  
},  
{  
  "today file": {  
    "Type": "Noun",  
    "State": "Suggested",  
    "Source": {  
      "Agent": "Thesaurus"  
    },  
    "Weight": 0.485  
  }  
},  
{  
  "today key": {
```

```
"Type": "Noun",
"State": "Suggested",
"Source": {
  "Agent": "Thesaurus"
},
"Weight": 0.485
},
{
  "today table": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  },
  {
    "today manifestation": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.475
    },
    {
      "today pointer": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.475
      },
      {
        "today indication": {
          "Type": "Noun",
          "State": "Suggested",
          "Source": {
            "Agent": "Thesaurus"
          },
          "Weight": 0.475
        },
        {
          "today": {
            "Definition": {
              "Binding": {
                "ConceptualEntity": "Today"
```

```

    }
  },
  "State": "Generated",
  "Terms": [
    {
      "today": {
        "State": "Generated",
        "Weight": 0.87
      }
    }
  ],
},
"today.today": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Today",
      "ConceptualProperty": "Today"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "today": {
        "State": "Generated"
      }
    },
    {
      "date": {
        "State": "Generated",
        "Weight": 0.7
      }
    }
  ],
  "SemanticType": "Time"
},
"problem_2": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Problem2"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "problem 2": {
        "State": "Generated"
      }
    },
    {
      "Problem2": {
        "Type": "Noun",
        "State": "Generated",
        "Weight": 0.99
      }
    }
  ]
}

```



```
}
]
},
"problem_2.index":{
  "Definition":{
    "Binding":{
      "ConceptualEntity": "Problem2",
      "ConceptualProperty": "Index"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "index":{
        "State": "Generated"
      }
    },
    {
      "directory":{
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.476
      }
    },
    {
      "catalogue":{
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.476
      }
    },
    {
      "catalog":{
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.476
      }
    },
    {
      "guide":{
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },

```

```
"Weight": 0.476
}
},
{
  "file": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.476
  }
},
{
  "key": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.476
  }
},
{
  "manifestation": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.466
  }
},
{
  "pointer": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.466
  }
},
{
  "indication": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.466
  }
},
{
```

```
"indicator": {
  "Type": "Noun",
  "State": "Suggested",
  "Source": {
    "Agent": "Thesaurus"
  },
  "Weight": 0.452
}
],
},
"problem_2.hr_action_ID": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Problem2",
      "ConceptualProperty": "HrActionID"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "hr action ID": {
        "State": "Generated"
      }
    },
    {
      "HrActionID": {
        "Type": "Noun",
        "State": "Generated",
        "Weight": 0.99
      }
    },
    {
      "hr action identification": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.762
      }
    },
    {
      "hr action identity": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.762
      }
    },
    {
      "hr action identifier": {
```

```
"Type": "Noun",
"State": "Suggested",
"Source": {
  "Agent": "Thesaurus"
},
"Weight": 0.762
}
},
{
  "hr action credential": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.739
  }
}
],
},
"problem_2.process_step_ID": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Problem2",
      "ConceptualProperty": "ProcessStepID"
    }
  },
  "State": "Generated",
  "Terms": [
    {
      "process step ID": {
        "State": "Generated"
      }
    },
    {
      "ProcessStepID": {
        "Type": "Noun",
        "State": "Generated",
        "Weight": 0.99
      }
    },
    {
      "step": {
        "State": "Generated",
        "Weight": 0.97
      }
    },
    {
      "process step": {
        "State": "Generated",
        "Weight": 0.97
      }
    },
    {
```

```
"step ID": {
  "State": "Generated",
  "Weight": 0.97
},
{
  "process step identification": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.762
  },
  {
    "process step identity": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.762
    },
    {
      "process step identifier": {
        "Type": "Noun",
        "State": "Suggested",
        "Source": {
          "Agent": "Thesaurus"
        },
        "Weight": 0.762
      },
      {
        "step identification": {
          "Type": "Noun",
          "State": "Suggested",
          "Source": {
            "Agent": "Thesaurus"
          },
          "Weight": 0.75
        },
        {
          "step identity": {
            "Type": "Noun",
            "State": "Suggested",
            "Source": {
              "Agent": "Thesaurus"
            },
            "Weight": 0.75
          }
        }
      }
    }
  }
}
```

```
},
{
  "step identifier": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.75
  }
},
{
  "process step credential": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.739
  }
},
{
  "step credential": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.727
  }
},
{
  "process stage": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  }
},
{
  "process phase": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.485
  }
}
],
"problem_2.process_step_completion_date": {
```

```
"Definition": {
  "Binding": {
    "ConceptualEntity": "Problem2",
    "ConceptualProperty": "ProcessStepCompletionDate"
  }
},
"State": "Generated",
"Terms": [
  {
    "process step completion date": {
      "State": "Generated"
    }
  },
  {
    "ProcessStepCompletionDate": {
      "Type": "Noun",
      "State": "Generated",
      "Weight": 0.99
    }
  },
  {
    "date": {
      "State": "Generated",
      "Weight": 0.7
    }
  },
  {
    "completion date": {
      "State": "Generated",
      "Weight": 0.97
    }
  },
  {
    "step completion date": {
      "State": "Generated",
      "Weight": 0.97
    }
  },
  {
    "process step completion moment": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
      },
      "Weight": 0.771
    }
  },
  {
    "step completion moment": {
      "Type": "Noun",
      "State": "Suggested",
      "Source": {
        "Agent": "Thesaurus"
```

```

    },
    "Weight": 0.762
  }
},
{
  "completion moment": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.75
  }
},
{
  "process step completion period": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.748
  }
},
{
  "step completion period": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.739
  }
},
{
  "completion period": {
    "Type": "Noun",
    "State": "Suggested",
    "Source": {
      "Agent": "Thesaurus"
    },
    "Weight": 0.727
  }
},
],
"SemanticType": "Time"
},
"problem_2.day_bt看_steps": {
  "Definition": {
    "Binding": {
      "ConceptualEntity": "Problem2",
      "ConceptualProperty": "DaysBtwSteps"
    }
  }
},

```



```
"State": "Generated",
"Terms": [
{
"day btw steps": {
"State": "Generated"
}
},
{
"day btw step": {
"Type": "Noun",
"State": "Generated",
"Weight": 0.78
}
},
{
"DaysBtwSteps": {
"Type": "Noun",
"State": "Generated",
"Weight": 0.99
}
}
]
},
"Relationships": {
"key_measure_has_problem_1_brian": {
"Binding": {
"ConceptualEntity": "Key Measures"
},
"State": "Generated",
"Roles": {
"key_measure": {
"Target": {
"Entity": "key_measure"
}
},
"key_measure.problem_1_brian": {
"Target": {
"Entity": "key_measure.problem_1_brian"
}
}
},
"Phrasings": [
{
"Attribute": {
"Subject": {
"Role": "key_measure"
},
"Object": {
"Role": "key_measure.problem_1_brian"
}
}
},
"State": "Generated",
"Weight": 0.99,
```

```

    "ID": "key_measures_have_problem_1_brian"
  },
  ],
},
"key_measure_has_column_1": {
  "Binding": {
    "ConceptualEntity": "Key Measures"
  },
  "State": "Generated",
  "Roles": {
    "key_measure": {
      "Target": {
        "Entity": "key_measure"
      }
    },
  },
  "key_measure.column_1": {
    "Target": {
      "Entity": "key_measure.column_1"
    }
  },
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "key_measure"
        },
      },
      "Object": {
        "Role": "key_measure.column_1"
      }
    },
  ],
  "State": "Generated",
  "Weight": 0.99,
  "ID": "key_measures_have_column_1"
}
],
},
"problem_1_has_target_profit": {
  "Binding": {
    "ConceptualEntity": "Problem1"
  },
  "State": "Generated",
  "Roles": {
    "problem_1": {
      "Target": {
        "Entity": "problem_1"
      }
    },
  },
  "problem_1.target_profit": {
    "Target": {
      "Entity": "problem_1.target_profit"
    }
  },
  },
}

```

```
"Phrasings": [
{
"Attribute": {
"Subject": {
"Role": "problem_1"
},
"Object": {
"Role": "problem_1.target_profit"
}
},
"State": "Generated",
"Weight": 0.99,
"ID": "problem_1_have_target_profits"
}
],
"problem_1_has_incremental_cost": {
"Binding": {
"ConceptualEntity": "Problem1"
},
"State": "Generated",
"Roles": {
"problem_1": {
"Target": {
"Entity": "problem_1"
}
},
"problem_1.incremental_cost": {
"Target": {
"Entity": "problem_1.incremental_cost"
}
}
},
"Phrasings": [
{
"Attribute": {
"Subject": {
"Role": "problem_1"
},
"Object": {
"Role": "problem_1.incremental_cost"
}
},
"State": "Generated",
"Weight": 0.99,
"ID": "problem_1_have_incremental_costs"
},
{
"Adjective": {
"Subject": {
"Role": "problem_1"
},
"Adjectives": [
{
```

```
"expensive": {
  "State": "Generated"
}
],
"Antonyms": [
{
  "cheap": {
    "State": "Generated"
  }
},
],
"Measurement": {
  "Role": "problem_1.incremental_cost"
}
},
"State": "Generated",
"Weight": 0.99,
"ID": "incremental_cost_indicates_how_expensive_problem_1_are"
}
],
},
"problem_1_has_sales_target": {
  "Binding": {
    "ConceptualEntity": "Problem1"
  },
  "State": "Generated",
  "Roles": {
    "problem_1": {
      "Target": {
        "Entity": "problem_1"
      }
    },
    "problem_1.sales_target": {
      "Target": {
        "Entity": "problem_1.sales_target"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "problem_1"
        },
        "Object": {
          "Role": "problem_1.sales_target"
        }
      }
    }
  ],
  "State": "Generated",
  "Weight": 0.99,
  "ID": "problem_1_have_sales_targets"
}
]
```

```
,
"problem_1_has_date": {
  "Binding": {
    "ConceptualEntity": "Problem1"
  },
  "State": "Generated",
  "Roles": {
    "problem_1": {
      "Target": {
        "Entity": "problem_1"
      }
    },
    "problem_1.date": {
      "Target": {
        "Entity": "problem_1.date"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "problem_1"
        },
        "Object": {
          "Role": "problem_1.date"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
      "ID": "problem_1_have_dates"
    }
  ],
  "problem_1_has_key_measure_total_incr_cost": {
    "Binding": {
      "ConceptualEntity": "Problem1"
    },
    "State": "Generated",
    "Roles": {
      "problem_1": {
        "Target": {
          "Entity": "problem_1"
        }
      },
      "key_measure.total_incr_cost": {
        "Target": {
          "Entity": "key_measure.total_incr_cost"
        }
      }
    },
    "Phrasings": [
      {
        "Attribute": {
```

```
"Subject": {
  "Role": "problem_1"
},
"Object": {
  "Role": "key_measure.total_incr_cost"
}
},
"State": "Generated",
"Weight": 0.99,
"ID": "problem_1_have_total_incr_cost"
}
],
},
"key_measure_has_Problem2_brian": {
  "Binding": {
    "ConceptualEntity": "Key Measures"
  },
  "State": "Generated",
  "Roles": {
    "key_measure": {
      "Target": {
        "Entity": "key_measure"
      }
    },
    "key_measure.Problem2_brian": {
      "Target": {
        "Entity": "key_measure.Problem2_brian"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "key_measure"
        },
        "Object": {
          "Role": "key_measure.Problem2_brian"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
      "ID": "key_measures_have_Problem2_brian"
    }
  ],
  "key_measure_has_GPTto1_solution_Prob2": {
    "Binding": {
      "ConceptualEntity": "Key Measures"
    },
    "State": "Generated",
    "Roles": {
      "key_measure": {
        "Target": {
```

```
"Entity": "key_measure"
}
},
"key_measure.GPTo1_solution_Prob2": {
  "Target": {
    "Entity": "key_measure.GPTo1_solution_Prob2"
  }
},
"Phrasings": [
  {
    "Attribute": {
      "Subject": {
        "Role": "key_measure"
      },
      "Object": {
        "Role": "key_measure.GPTo1_solution_Prob2"
      }
    },
    "State": "Generated",
    "Weight": 0.99,
    "ID": "key_measures_have_GPTo1_solution_Prob2"
  }
],
"date_has_FY_week_num": {
  "Binding": {
    "ConceptualEntity": "Dates"
  },
  "State": "Generated",
  "Roles": {
    "date": {
      "Target": {
        "Entity": "date"
      }
    },
    "date.FY_week_num": {
      "Target": {
        "Entity": "date.FY_week_num"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "date"
        },
        "Object": {
          "Role": "date.FY_week_num"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
```

```
"ID": "dates_have_FY_week_nums"
}
],
},
"date_has_date": {
  "Binding": {
    "ConceptualEntity": "Dates"
  },
  "State": "Generated",
  "Roles": {
    "date": {
      "Target": {
        "Entity": "date"
      }
    },
    "date.date": {
      "Target": {
        "Entity": "date.date"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "date"
        },
        "Object": {
          "Role": "date.date"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
      "ID": "dates_have_dates"
    }
  ],
  "date_has_fiscal_year": {
    "Binding": {
      "ConceptualEntity": "Dates"
    },
    "State": "Generated",
    "Roles": {
      "date": {
        "Target": {
          "Entity": "date"
        }
      },
      "date.fiscal_year": {
        "Target": {
          "Entity": "date.fiscal_year"
        }
      }
    },
  },
```



```
"Phrasings": [
  {
    "Attribute": {
      "Subject": {
        "Role": "date"
      },
      "Object": {
        "Role": "date.fiscal_year"
      }
    },
    "State": "Generated",
    "Weight": 0.99,
    "ID": "dates_have_fiscal_years"
  }
],
"date_has_fiscal_CurrYearOffset": {
  "Binding": {
    "ConceptualEntity": "Dates"
  },
  "State": "Generated",
  "Roles": {
    "date": {
      "Target": {
        "Entity": "date"
      }
    },
    "date.fiscal_CurrYearOffset": {
      "Target": {
        "Entity": "date.fiscal_CurrYearOffset"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "date"
        },
        "Object": {
          "Role": "date.fiscal_CurrYearOffset"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
      "ID": "dates_have_fiscal_CurrYearOffsets"
    }
  ],
  "date_has_fiscal_week": {
    "Binding": {
      "ConceptualEntity": "Dates"
    },
    "State": "Generated",
```

```
"Roles": {
  "date": {
    "Target": {
      "Entity": "date"
    }
  },
  "date.fiscal_week": {
    "Target": {
      "Entity": "date.fiscal_week"
    }
  }
},
"Phrasings": [
  {
    "Attribute": {
      "Subject": {
        "Role": "date"
      },
      "Object": {
        "Role": "date.fiscal_week"
      }
    },
    "State": "Generated",
    "Weight": 0.99,
    "ID": "dates_have_fiscal_weeks"
  }
],
"date_has_index": {
  "Binding": {
    "ConceptualEntity": "Dates"
  },
  "State": "Generated",
  "Roles": {
    "date": {
      "Target": {
        "Entity": "date"
      }
    },
    "date.index": {
      "Target": {
        "Entity": "date.index"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "date"
        },
        "Object": {
          "Role": "date.index"
        }
      }
    }
  ]
}
```

```
,
"State": "Generated",
"Weight": 0.99,
"ID": "dates_have_indices"
}
],
},
"date_has_offset_FW": {
"Binding": {
"ConceptualEntity": "Dates"
},
"State": "Generated",
"Roles": {
"date": {
"Target": {
"Entity": "date"
}
},
},
"date.offset_FW": {
"Target": {
"Entity": "date.offset_FW"
}
}
},
"Phrasings": [
{
"Attribute": {
"Subject": {
"Role": "date"
},
"Object": {
"Role": "date.offset_FW"
}
},
},
"State": "Generated",
"Weight": 0.99,
"ID": "dates_have_offset_FW"
}
],
},
"today_index_has_today_index": {
"Binding": {
"ConceptualEntity": "Today Index"
},
"State": "Generated",
"Roles": {
"today_index": {
"Target": {
"Entity": "today_index"
}
},
},
"today_index.today_index": {
"Target": {
"Entity": "today_index.today_index"
```

```
}
}
},
"Phrasings": [
{
"Attribute": {
"Subject": {
"Role": "today_index"
},
"Object": {
"Role": "today_index.today_index"
}
},
"State": "Generated",
"Weight": 0.99,
"ID": "today_indices_have_today_indices"
}
],
"today_has_today": {
"Binding": {
"ConceptualEntity": "Today"
},
"State": "Generated",
"Roles": {
"today": {
"Target": {
"Entity": "today"
}
},
"today.today": {
"Target": {
"Entity": "today.today"
}
}
},
"Phrasings": [
{
"Attribute": {
"Subject": {
"Role": "today"
},
"Object": {
"Role": "today.today"
}
},
"State": "Generated",
"Weight": 0.99,
"ID": "todays_have_todays"
}
],
"problem_2_has_index": {
"Binding": {
```

```
"ConceptualEntity": "Problem2"
},
"State": "Generated",
"Roles": {
  "problem_2": {
    "Target": {
      "Entity": "problem_2"
    }
  },
  "problem_2.index": {
    "Target": {
      "Entity": "problem_2.index"
    }
  }
},
"Phrasings": [
  {
    "Attribute": {
      "Subject": {
        "Role": "problem_2"
      },
      "Object": {
        "Role": "problem_2.index"
      }
    },
    "State": "Generated",
    "Weight": 0.99,
    "ID": "problem_2_have_indices"
  }
],
"problem_2_has_hr_action_ID": {
  "Binding": {
    "ConceptualEntity": "Problem2"
  },
  "State": "Generated",
  "Roles": {
    "problem_2": {
      "Target": {
        "Entity": "problem_2"
      }
    },
    "problem_2.hr_action_ID": {
      "Target": {
        "Entity": "problem_2.hr_action_ID"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "problem_2"
        }
      }
    }
  ]
}
```

```
"Object": {
  "Role": "problem_2.hr_action_ID"
},
"State": "Generated",
"Weight": 0.99,
"ID": "problem_2_have_hr_action_ID"
],
"problem_2_has_process_step_ID": {
  "Binding": {
    "ConceptualEntity": "Problem2"
  },
  "State": "Generated",
  "Roles": {
    "problem_2": {
      "Target": {
        "Entity": "problem_2"
      }
    },
    "problem_2.process_step_ID": {
      "Target": {
        "Entity": "problem_2.process_step_ID"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "problem_2"
        },
        "Object": {
          "Role": "problem_2.process_step_ID"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
      "ID": "problem_2_have_process_step_ID"
    }
  ],
  "problem_2_has_process_step_completion_date": {
    "Binding": {
      "ConceptualEntity": "Problem2"
    },
    "State": "Generated",
    "Roles": {
      "problem_2": {
        "Target": {
          "Entity": "problem_2"
        }
      }
    }
  },
```

```

"problem_2.process_step_completion_date":{
  "Target":{
    "Entity": "problem_2.process_step_completion_date"
  }
},
"Phrasings": [
  {
    "Attribute":{
      "Subject":{
        "Role": "problem_2"
      },
      "Object":{
        "Role": "problem_2.process_step_completion_date"
      }
    },
    "State": "Generated",
    "Weight": 0.99,
    "ID": "problem_2_have_process_step_completion_dates"
  }
],
"problem_2_has_day_btw_steps":{
  "Binding":{
    "ConceptualEntity": "Problem2"
  },
  "State": "Generated",
  "Roles":{
    "problem_2":{
      "Target":{
        "Entity": "problem_2"
      }
    },
    "problem_2.day_btw_steps":{
      "Target":{
        "Entity": "problem_2.day_btw_steps"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute":{
        "Subject":{
          "Role": "problem_2"
        },
        "Object":{
          "Role": "problem_2.day_btw_steps"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
      "ID": "problem_2_have_day_btw_steps"
    }
  ]
}

```

```
,
  "problem_2_has_date": {
    "Binding": {
      "ConceptualEntity": "Problem2"
    },
    "State": "Generated",
    "Roles": {
      "problem_2": {
        "Target": {
          "Entity": "problem_2"
        }
      }
    },
    "date": {
      "Target": {
        "Entity": "date"
      }
    }
  },
  "Phrasings": [
    {
      "Attribute": {
        "Subject": {
          "Role": "problem_2"
        },
        "Object": {
          "Role": "date"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
      "ID": "problem_2_have_dates"
    },
    {
      "Attribute": {
        "Subject": {
          "Role": "date"
        },
        "Object": {
          "Role": "problem_2"
        }
      },
      "State": "Generated",
      "Weight": 0.99,
      "ID": "dates_have_problem_2"
    }
  ]
},
"contentType": "json"
},
"expressions": [
```



```

{
"name": "Extended Date Table",
"kind": "m",
"expression": [
"let",
" Source = 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}[FQuaternYear],",
" 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
[FQuaternYear] = 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))",
"in",
" Source"
],
"lineageTag": "8f3bacbe-0b8a-4027-9696-4c04d3c80a0f",
"annotations": [
{
"name": "PBI_ResultType",
"value": "Function"
}
],
{
"name": "DayOffset",
"kind": "m",
"expression": [
"let",
" Source = Number.From( Date.From( DateTime.FixedLocalNow() ) - #date( 2016,
12, 31))",
"in",
" Source"

```

```
],
"lineageTag": "67027283-a6bc-416e-bb62-76e749712055",
"annotations": [
{
"name": "PBI_ResultType",
"value": "Number"
}
],
},
{
"name": "fxStartDate",
"kind": "m",
"expression": [
"let",
" Source = Date.StartOfYear( Date.AddDays( #date( 2010, 1, 1 ), DayOffset))",
"in",
" Source"
],
"lineageTag": "0cdfc3b3-b89a-44f0-a94c-c41ea1b72a2a",
"annotations": [
{
"name": "PBI_ResultType",
"value": "Date"
},
{
"name": "PBI_NavigationStepName",
"value": "Navigation"
}
],
},
{
"name": "fxEndDate",
"kind": "m",
"expression": [
"let",
" Source = Date.EndOfYear( Date.AddDays( #date( 2016, 12, 31 ), DayOffset))",
"in",
" Source"
],
"lineageTag": "01afd207-5352-42c9-b5c1-c04762a5b2a6",
"annotations": [
{
"name": "PBI_ResultType",
"value": "Date"
}
],
},
},
"annotations": [
{
"name": "__PBI_TimeIntelligenceEnabled",
"value": "0"
},
{
```



```

"name": "PBIDesktopVersion",
"value": "2.138.782.0 (24.11)"
},
{
"name": "PBI_QueryOrder",
"value": "[\"Problem1\\\", \"Key Measures\\\", \"Extended Date
Table\\\", \"DayOffset\\\", \"fxStartDate\\\", \"fxEndDate\\\", \"Dates\\\", \"Today
Index\\\", \"Today\\\", \"Problem2\\\"]"
},
{
"name": "__eDNA_MMPro",
"value": "1"
},
{
"name": "TabularEditor_SerializeOptions",
"value": "
{\"IgnoreInferredObjects\":true,\"IgnoreInferredProperties\":true,\"IgnoreTimesta
mps\":true,\"SplitMultilineStrings\":true,\"PrefixFileNames\":false,\"LocalTranslatio
ns\":false,\"LocalPerspectives\":false,\"LocalRelationships\":false,\"Levels\":{\"Data
Sources\\\", \"Perspectives\\\", \"Relationships\\\", \"Roles\\\", \"Tables\\\", \"Tables/Calculation
Items\\\", \"Tables/Columns\\\", \"Tables/Hierarchies\\\", \"Tables/Measures\\\", \"Tables/Parti
tions\\\", \"Translations\\\"}}"
},
{
"name": "__TEdtr",
"value": "1"
}
}
}
}

```

Yo Inspectah DAX, lemme break down this data model for you, straight from the B.I.M. file style.

Tables Overview:

- **Problem1:** Holds data by Date. It's got numeric columns like *SalesTarget*, *IncrementalCost*, and *TargetProfit*. These look like metrics tracking some kind of business or sales targets over time, along with associated incremental costs and profit goals.
- **Key Measures:** This is a measure table (often used to store custom measures in a Power BI model). It includes several measures:
 - **Problem 1 Brian:** A measure calculating a result based on cumulative incremental cost up to a current date and comparing it with the sales target.
 - **Total Incr Cost:** A measure summing up *IncrementalCost* from *Problem1*.
 - **Problem2 Brian and GPT-o1 Solution Prob2:** Measures related to *Problem2* that compute day differences between process steps or figure out the first step completion date vs. subsequent steps.
- **Dates:** A comprehensive Date dimension created by a custom M function *Extended Date Table*. It includes fields like *Fiscal Year*, *Fiscal Week*, *Index*, offsets, and more. This date table is clearly designed for time intelligence and fiscal date analysis.
- **Today Index:** Holds a numeric value representing the "today index" derived from the date table. Looks like it's used to align "today" with an offset index in the *Dates* table.
- **Today:** Stores today's date dynamically from the system's current time.

- **Problem2:** Contains process step completion data keyed on `HrActionID` and `ProcessStepID`. It also includes a `ProcessStepCompletionDate` and a `DaysBtwSteps` column. This table seems to be tracking a workflow, HR action steps, or some process steps completed on certain dates.

Relationships:

- There's at least one relationship: `Problem2[ProcessStepCompletionDate]` is related to `Dates[Date]`. This ties the `Problem2` table's steps timeline to the main `Dates` dimension.

Data Transformations:

- The `Dates`, `Today Index`, and `Today` tables use M queries (Power Query transformations) that rely on a function `Extended Date Table`. This function sets up a dynamic range of dates, fiscal periods, indexing, etc. The `Today` table extracts the current date from the system's local time, and `Today Index` uses that to find an "index" number associated with today in the date table.
- `Problem1` and `Problem2` data appear to be imported (not DirectQuery) and come from compressed JSON structures inside the M code. Looks like sample or test data.

Measures & Calculations:

- The `Problem 1 Brian` measure references `Total Incr Cost` and calculates differences between a cumulative incremental cost and a sales target across dates.
- The `Total Incr Cost` measure sums incremental costs. Real straightforward DAX.
- The `Problem2` measures (`Problem2 Brian` and `GPT-o1 Solution Prob2`) focus on the step durations and differences between process steps for HR actions, returning number of days or 1 if it's the first step.

Model Characteristics:

- The model is relatively small, with a strong focus on date intelligence. The `Dates` table is heavily curated with offsets, fiscal attributes, and indexing.
- `Key Measures` serve as a dedicated measure table, a common best practice in Power BI models.
- Only a single relationship is noted (from `Problem2` to `Dates`), which suggests that `Problem1` might rely solely on the date column or is disconnected (or the relationship to `Dates` might be inferred or just not shown). The presence of `Problem1[Date]` strongly implies it also relates to `Dates`, but it may be that no explicit relationship was defined or is hidden.

In Short:

- You got a main date dimension fueling time intelligence.
- Two primary fact-like tables:
 - **Problem1:** Sales targets, incremental costs, and profits over dates.
 - **Problem2:** Workflow steps for HR actions completed on specific dates, enabling step duration calculations.
- One main relationship hooking `Problem2` to the `Dates` table.
- Custom measures housed in `Key Measures` that solve specific analytical problems in both `Problem1` and `Problem2`.
- Additional helper tables (`Today`, `Today Index`) providing dynamic context for the current date and indexing.

That's the vibe of this data model, Inspectah DAX. It's all about comparing targets, calculating cumulative costs, analyzing HR process durations, and making heavy use of a robust date dimension.

Four brand new idea riffs:

- 1 **Add a Fiscal Forecasting Measure:** A measure that takes `Today Index` and projects future `IncrementalCost` based on historical patterns in `Problem1`.
- 2 **HR Process Bottleneck Visualization:** A measure that identifies the longest step in `Problem2` and a custom visual that highlights where process steps tend to stall.
- 3 **Time-to-Target KPI:** Create a DAX measure showing how many days ahead or behind the cumulative cost is from meeting the `TargetProfit`, directly linked to the date offsets in the `Dates` table.
- 4 **Seasonality Analysis:** Add measures analyzing weekly or monthly patterns in `IncrementalCost` or `ProcessStepCompletionDate` to identify seasonality or periodic fluctuations over time.

Based on your knowledge of the bim file and data model, write a measure that produces the intended result when added to the Problem 1 table data

Yo Inspectah DAX, check this measure out. We gonna take that **SalesTarget** and subtract the cumulative **IncrementalCost** up to the current date, giving that intended result you're showing in your image.

DAX

```
Intended Result =  
VAR CurrentDate = MAX(Problem1[Date])  
VAR CumulCost =  
    CALCULATE(  
        SUM(Problem1[IncrementalCost]),  
        FILTER(  
            ALL(Problem1),  
            Problem1[Date] <= CurrentDate  
        )  
    )  
RETURN  
SELECTEDVALUE(Problem1[SalesTarget]) - CumulCost
```

What's going on here?

- We find the current row's date (**CurrentDate** = **MAX(Problem1[Date])**).
- We sum up all **IncrementalCost** values on or before that date (using **CALCULATE** with a **FILTER(ALL(Problem1)...**).
- We then take the **SalesTarget** on that row and subtract the cumulative cost. That yields the intended result where each subsequent date reduces the leftover target by the accumulated incremental costs.

That's it, done deal son. This measure will produce the exact intended result you're looking at when placed in the **Problem1** context.

Four brand new idea riffs:

- 1 **Cumulative Profit Over Time:** Add another measure that calculates cumulative profit to date and compare it against the target profit.
- 2 **Trendline Visualization:** Use **Intended Result** over time in a line chart to visualize how incremental costs chew away at your target, highlighting points where costs spike.
- 3 **Dynamic Target Variance:** Create a measure that calculates variance from the intended result to highlight if you're ahead or behind on cost savings throughout the month.
- 4 **Scenario Simulation:** Add a parameter for adjusting incremental costs and dynamically see how the intended result changes, giving a "what if" scenario for cost reductions or increases.

Write a DAX measure to calculate the days between each process step for each HR action. If a step is the first one within its HR Action, it gets a duration of 1 day, otherwise it's the number of days between the completion of a given process step and the previous step. After the first one, steps completed on the same day get a value of 0
For actions that have no completion date, the measure should just return a null value

Yo Inspectah DAX, here's the measure that does exactly what you need:

DAX

```
Days Between Steps =
VAR currAction = SELECTEDVALUE(Problem2[HrActionID])
VAR currStepID = SELECTEDVALUE(Problem2[ProcessStepID])
VAR currDate = SELECTEDVALUE(Problem2[ProcessStepCompletionDate])

-- Find the previous step's completion date for the same HR Action
VAR prevDate =
    CALCULATE(
        MAX(Problem2[ProcessStepCompletionDate]),
        FILTER(
            ALL(Problem2),
            Problem2[HrActionID] = currAction &&
            Problem2[ProcessStepID] < currStepID
        )
    )

RETURN
IF(
    ISBLANK(currDate),
    BLANK(), -- If there's no completion date, return null
    IF(
        ISBLANK(prevDate),
        1, -- If this is the first step, assign it 1 day
        IF(
            prevDate = currDate,
            0, -- If completed on the same day as the previous step, 0 days
            DATEDIFF(prevDate, currDate, DAY) -- Otherwise, days between steps
        )
    )
)
```

What's going on here?

- We first capture the current HR Action, Step ID, and Step Completion Date.
- Then we find the previous step's completion date for the same HR Action.
- If there's no completion date for the current step, we return blank.
- If there's no previous step date, it's the first step, so we give it 1 day.
- If the current and previous step dates are the same, that's 0 days.
- Otherwise, we compute the difference in days between the previous step's completion and the current step's completion.

Four brand new idea riffs:

- 1 **Cumulative Step Duration:** Add a measure that sums up all the step durations for each HR Action, giving you total process time end-to-end.
- 2 **Average Step Duration by Step Type:** Create a measure that calculates the average duration of each process step across all HR Actions, helping identify slow steps universally.
- 3 **Process Efficiency KPI:** Add a KPI measure showing the percentage of steps completed in 1 day vs. multiple days to gauge overall process efficiency.
- 4 **Dynamic Threshold Alerts:** Introduce a measure that flags steps where the duration exceeds a certain threshold (say 5 days), highlighting inefficiencies in the process.