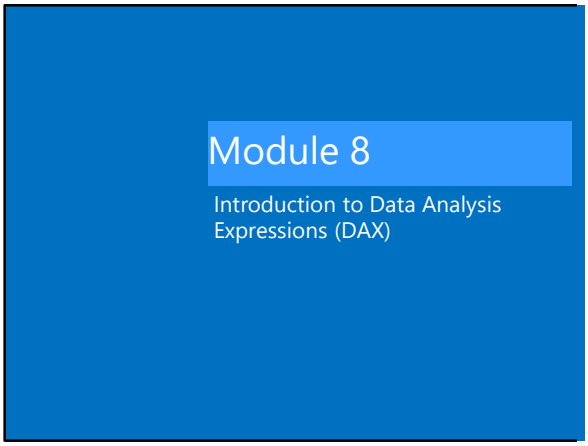
8: Introduction to Data Analysis Expressions (DAX)



The **20768C-MIA-SQL** virtual machine used in the lab for this module includes software services that can take a while to start. For the best experience, have students start the **20768C-MIA-DC** and **20768C-MIA-SQL** virtual machines at the beginning of the module so that the services are ready before they start the lab.

20768C

8: Introduction to Data Analysis Expressions (DAX)

Module Overview

- DAX Fundamentals
- Using DAX to Create Calculated Columns and Measures in a Tabular Data Model

20768C

8: Introduction to Data Analysis Expressions (DAX)

Lesson 1: DAX Fundamentals

- Overview of DAX
- DAX Functions
- DAX Syntax and Data Types
- Aggregations
- Context
- DAX Queries

8: Introduction to Data Analysis Expressions (DAX)

Overview of DAX

- Formula-based language for building business logic and queries in tabular data models:
  - Create columns and measures based on complex custom calculations
  - Create columns that reference data in related tables
  - Define parent-child hierarchies
  - Create Key Performance Indicators (KPIs)
- Syntax is similar to Microsoft Excel formulas

DAX Functions

- Text functions
- Information functions
- Filter and value functions
- Logical functions
- Mathematical and trigonometric functions
- Statistical and aggregation functions
- Date and time functions
- Time intelligence functions
- Navigation and Lookup functions

Emphasize that only a few of the available DAX functions are mentioned here. DAX is a rich expression language, with more than 200 functions.

DAX Syntax and Data Types

- DAX Syntax
  - DAX formulas start with the equal sign (=) followed by an expression
  - Expressions can contain functions, operators, constants, and references to columns
  - Column references:
    - **'table name'[column name] –** a fully qualified name
    - **[column name]** – an unqualified name
  - Measure names must be enclosed in brackets
  - Table names containing spaces must be enclosed in single quotation marks

DAX Data Types and Operators

- DAX Data Types:
  - Integer (8-byte)
  - Real (8-byte)
  - Boolean
  - String
  - Date/time
  - Currency
  - BLANK
  - Table
- DAX Operators:
  - Arithmetic
  - Comparison
  - Text concatenation
  - Logical

Aggregations

- Summarize underlying detailed data for analysis
- Use automatic aggregation for simple calculations:
  - SUM
  - COUNT
  - MIN
  - MAX
  - AVERAGE
- Create a measure for more complex aggregations

Emphasize that the list of DAX statistical functions is extensive; this is a list of some more commonly used aggregations and is not exhaustive.

Context

- A DAX measure or calculated column defines a field that you can use in a PivotTable table or a PivotChart chart
- The exact values that appear in PivotTable tables and PivotChart charts vary with context:
  - Row context
  - Filter context
  - Query context

DAX Queries

- Client applications, such as the Power View reporting tool, issue DAX queries
- You can write queries manually by using the DAX query language
- You can filter, order, and summarize results

```
EVALUATE(FILTER('Reseller Sales', 'Reseller Sales'
[OrderDateKey]>20040101))
```

Question

When a DAX expression is evaluated, which contexts apply?

( )Option 1: The row context.

( )Option 2: The filter context.

( )Option 3: Both the row context and the filter context.

Answer

(√) Option 3: Both the row context and the filter context.

8: Introduction to Data Analysis Expressions (DAX)

Lesson 2: Using DAX to Create Calculated Columns and Measures in a Tabular Data Model

- Calculated Columns
- Measures
- KPIs
- Demonstration: Creating and Using Measures and Calculated Columns
- Multiple Relationships
- Demonstration: Using a Specific Relationship in a Measure
- Introduction to Time Intelligence
- Demonstration: Using Time Intelligence
- Implementing Parent-Child Hierarchies

Calculated Columns

- Named columns that are populated by using a DAX formula
- A value is calculated for each row in the table when the calculated column is created

Measures

- Named formulas that can contain sophisticated business logic
- Usually aggregating numerical values
- Defined in the Measure Grid of a table
- Client tools associate measures with the tables in which they are defined

Make the point that measures are associated with the table in which they are defined. There may be cases where you want to create a blank table for measures that aren't specific to any one table.

For example, a data model could include existing tables named **Internet Sales** and **Reseller Sales**, which both contain a **SalesAmount** column. In each table, you could define a measure named **Revenue**, based on the expression **SUM([SalesAmount])**. In client tools, users could easily identify the **Revenue** measures for **Internet Sales** and **Reseller Sales**.

You might also choose to create a single Total Revenue measure, based on the expression **SUM('Internet Sales'[SalesAmount]) + SUM('Reseller Sales'[SalesAmount])**. To avoid confusing users, you could define this measure in the measure grid of a blank table named **All Sales** that you have created by pasting a worksheet into the data model.

KPIs

- Used to compare measures against target values
- Definition includes:
  - Base Value: the actual measure result
  - Target Value: the goal for the measure
  - Status Thresholds: indicators to show how the base value compares to the target value

Demonstration: Creating and Using Measures and Calculated Columns

In this demonstration, you will see how to:

- Create calculated columns
- Create measures
- Create a KPI
- Analyze calculated columns and measures

Preparation Steps

Start the **20768C-MIA-DC** and **20768C-MIA-SQL** virtual machines.

Demonstration Steps

Create Calculated Columns

1. Ensure that the **20768C-MIA-DC** and **20768C-MIA-SQL** virtual machines are both running, and log on to the **20768C-MIA-SQL** virtual machine as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.

2. In the **D:\Demofiles\Mod08** folder, right-click **Setup.cmd** and click **Run as administrator**. When prompted, click **Yes**.

3. Start Visual Studio 2015 and open the **TabularDemo.sln** solution in the **D:\Demofiles\Mod08** folder.

4. If the **Tabular model designer** dialog box appears, in the **Workspace server** list, specify **localhost\SQL2** and then click **OK**.

5. If the **Tabular Model Designer** dialog box appears, click **Yes**.

6. In Solution Explorer, double-click **Model.bim** to open the data model designer.

7. On the **Model** menu, point to **Process**, and click **Process All**. If you are prompted for impersonation credentials, specify the user name **ADVENTUREWORKS\ServiceAcct** with the password **Pa55w.rd** and click **OK**.

8. In the **Data Processing** dialog box, when all of the tables have been processed, click **Close**.

9. On the **Customer** tab, in the first empty column after the existing columns, double-click **Add Column** and name the new column **Full Name**.

10. With the **Full Name** column selected, in the formula bar, enter the following DAX expression:

    =CONCATENATE([FirstName], " " & [LastName])

    Wait for the table to finish updating, and note the calculated values in the new column.

(More notes on the next slide)

11. Click the **FirstName** column heading, hold **Shift** and click the **LastName** column heading, and then right-click either of the selected column headings and click **Hide from Client Tools**.

12. On the **Internet Sales** tab, in the first empty column after the existing columns, double-click **Add Column** and name the new column **SalesProfit**.

13. With the **SalesProfit** column selected, in the formula bar, enter the following DAX expression:

    =[SalesAmount]-[TotalProductCost]

    Wait for the table to finish updating, and note the calculated values in the new column.

Create Measures

1. On the **Internet Sales** tab, click the first empty cell in the measure grid under the **SalesProfit** column.

2. In the formula bar, enter the following DAX expression:

    Profit:=SUM([SalesProfit])

3. Select the cell containing the Profit measure, and press F4 to display the properties pane. Then set the **Format** property to **Currency**.

4. Click the empty cell in the measure grid under the **Profit** measure.

5. In the formula bar, enter the following DAX expression:

    Margin:=[Profit]/[Revenue]

6. Select the cell containing the **Margin** measure, and press F4 to display the properties pane. Then set the **Format** property to **Percentage**.

7. Right-click the **SalesProfit** column header and click **Hide from Client Tools**.

8. On the **File** menu, click **Save All**.

(More notes on the next slide)

Create a KPI

1. On the **Internet Sales** tab, click the empty cell in the measure grid under the **Margin** measure.
2. In the formula bar, enter the following DAX expression:

   Target Margin:=(SUM('Product'[ListPrice]) - SUM('Product'[StandardCost])) / SUM('Product'[ListPrice])

3. Select the cell containing the **Margin** measure, and press F4 to display the properties pane. Then set the **Format** property to **Percentage**.
4. On the **Internet Sales** tab, right-click the cell in the measure grid containing the **Margin** measure and click **Create KPI**.
5. In the **Key Performance Indicator (KPI)** dialog box, note that the **KPI base measure (value)** is defined by the **Margin** measure.
6. Under **Target**, ensure that **Measure** is selected and select **Target Margin**.
7. Set the first status threshold to **65%** and the second to **90%**.
8. Note the default icon style, and click **OK**.
9. On the **File** menu, click **Save All**.

Analyze Columns and Measures

1. On the **Model** menu, click **Analyze in Excel**.
2. In the **Analyze in Excel** dialog box, ensure that **Current Windows User** is selected and that the **(Default)** perspective is specified, and click **OK**.
3. In Excel, in the **PivotTable Fields** pane, in the **Internet Sales** table, select the **Profit** and **Margin** measures so that they are summarized in the **PivotTable**.
4. In the **PivotTable Fields** pane, in the **Customer** table, select **Full Name** so that the PivotTable shows profit and margin by customer.
5. In the **PivotTable Fields** pane, expand **KPIs**, expand **Margin**, and select **Status** so that an indicator shows the KPI status for each customer.

(More notes on the next slide)

6. Close Excel without saving the workbook.

7. Keep Visual Studio open for the next demonstration.

Multiple Relationships

- Look up a value in an unrelated table:

```
LOOKUPVALUE('Date'[Fiscal Year], [Full Date], [Registration Date])
```

- Look up a value in a related table:

```
RELATED('Product Category'[Category Name])
```

- Aggregate values in multiple rows in a related table:

```
RELATED('Product Category'[Category Name])
```

- Use a non-active relationship to aggregate data:

```
CALCULATE(SUM(Reseller Sales[Sales Amount]),
  USERELATIONSHIP(Reseller Sales[ShipDateKey],Date[DateKey]))
```

Demonstration: Using a Specific Relationship in a Measure

In this demonstration, you will see how to:
- Retrieve a value from an unrelated table
- Retrieve a value from a related table
- Aggregate related values
- Analyze related data

Preparation Steps

Complete the previous demonstration in this module.

Demonstration Steps

Retrieve a Value from an Unrelated Table

1. Ensure that you have completed the previous demonstration.
2. In Visual Studio, with the **Model.bim** pane open in data view, on the **Customer** tab, after the existing columns, double-click the **Add Column** header and enter the column name **First Fiscal Year**.
3. With the **First Fiscal Year** column selected, in the formula bar, enter the following DAX expression:

   =LOOKUPVALUE('Order Date'[FiscalYear], [Date], [DateFirstPurchase])

   Wait for the table to finish updating, and note the calculated values in the new column.

Retrieve a Value from a Related Table

1. On the **Customer** tab, after the existing columns, double-click the **Add Column** header and enter the column name **Country-Region**.
2. With the **Country-Region** column selected, in the formula bar, enter the following DAX expression:

   =RELATED('Geography'[CountryRegionCode])

   Wait for the table to finish updating, and note the calculated values in the new column.

Aggregate Related Values

1. On the **Customer** tab, after the existing columns, double-click the **Add Column** header and enter the column name **Order Count**.
2. With the **Order Count** column selected, in the formula bar, enter the following DAX expression:

   =COUNTAX(RELATEDTABLE('Internet Sales'), [SalesOrderNumber])

(More notes on the next slide)

Wait for the table to finish updating, and note the calculated values in the new column.

3. On the **File** menu, click **Save All**.

Analyze Related Data

1. On the **Model** menu, click **Analyze in Excel**.
2. In the **Analyze in Excel** dialog box, ensure that **Current Windows User** is selected and that the **(Default)** perspective is specified, and click **OK**.
3. In Excel, in the **PivotTable Fields** pane, in the **Internet Sales** table, select the **Revenue** measure.
4. In the **PivotTable Fields** pane, in the **Customer** table, select **First Fiscal Year** so that the PivotTable shows sales revenue by customer, based on the first year in which customers made a purchase.
5. In the **PivotTable Fields** pane, in the **Customer** table, clear **First Fiscal Year** and select **Country-Region** so that the PivotTable shows sales revenue by customer country or region.
6. In the **PivotTable Fields** pane, in the **Customer** table, clear **Country-Region** and select **Order Count** so that the PivotTable shows sales revenue by customer, based on the number of orders placed.
7. Close Excel without saving the workbook.
8. Leave Visual Studio open for the next demonstration.

Introduction to Time Intelligence

- Compare data from one time period against equivalent data from a different time period
- The tabular model should include a separate table that contains only date information
  - The date table should have a continuous range of dates without gaps
  - The column in the date table that uses the date data type should use "day" as the lowest level of granularity
  - Mark the table as Date Table to use time intelligence functions against that table

Demonstration: Using Time Intelligence

In this demonstration, you will see how to:
- Mark a table as a date table
- Create a measure that uses time intelligence
- Analyze data using time intelligence

Preparation Steps

Complete the previous demonstration.

Demonstration Steps

Mark a Table As a Date Table

1. Ensure you have completed the previous demonstration.
2. In Visual Studio, with the **Model.bim** pane open in data view, click the **Order Date** tab.
3. On the **Table** menu, point to **Date** and click the **Mark As Date Table**.
4. In the **Mark As Date Table** dialog box, select the **Date** column, and click **OK**.

Create a Measure That Uses Time Intelligence

1. In the model designer, click the **Internet Sales** tab.
2. Click the first empty cell in the measure grid, then in the formula bar, enter the following DAX expression:

```
Previous Year Revenue:=CALCULATE(SUM([SalesAmount]),
SAMEPERIODLASTYEAR('Order Date'[Date]))
```

3. Select the cell containing the **Previous Year Revenue** measure, and press F4 to display the properties pane. Then set the **Format** property to **Currency**.
4. On the **File** menu, click **Save All**.

Analyze Data Using Time Intelligence

1. On the **Model** menu, click **Analyze in Excel**.
2. In the **Analyze in Excel** dialog box, ensure that **Current Windows User** is selected and that the **(Default)** perspective is specified, and click **OK**.
3. In Excel opens, in the **PivotTable Fields** pane, in the **Internet Sales** table, select the **Revenue** and **Previous Year Revenue** measures.

(More notes on the next slide)

4. In the **PivotTable Fields** pane, in the **Order Date** table, drag **Calendar Date** to the **Rows** area so that the PivotTable shows sales revenue and previous year revenue for each year.

5. Expand **2007** and **2008**, and note that the revenue for each quarter in 2007 is shown as the previous year revenue for the same quarters in 2008.

6. Close Excel without saving the workbook.

7. Close Visual Studio.

Implementing Parent-Child Hierarchies

- Requires calculated measures based on DAX functions
  - PATH
  - LOOKUPVALUE
  - PATHITEM
- Create levels in a table
- Create the hierarchy in the data model

Point out that students will have an opportunity to create a parent-child hierarchy for themselves in the lab.

Question

You are creating a calculated column in a table, based on the values in a column in another table. The two tables have a one-to-many relationship, based on a key column. Which DAX function should you use to perform the lookup in the most efficient way?

( )Option 1: USERELATIONSHIP

( )Option 2: RELATEDTABLE

( )Option 3: RELATED

( )Option 4: LOOKUPVALUE

Answer

(√) Option 3: RELATED

Lab: Using DAX to Enhance a Tabular Data Model

- Exercise 1: Creating Calculated Columns
- Exercise 2: Creating Measures
- Exercise 3: Creating a KPI
- Exercise 4: Implementing a Parent-Child Hierarchy

Logon Information
Virtual machine: **20768C-MIA-SQL**
User name: **ADVENTURWORKS\Student**
Password: **Pa55w.rd**

Estimated Time: 75 minutes

Point out that the lab instructions are deliberately designed to be high level so that students need to think carefully about what they are trying to accomplish and work out how best to proceed. Encourage students to read the scenario information carefully and collaborate with each other to meet the requirements. Remind students that, if they find a particular task or exercise too challenging, they can find step-by-step instructions in the lab answer key.

Point out that, like all other labs in this course, students must start by running a setup script to prepare the lab environment.

Exercise 1: Creating Calculated Columns

You have the raw data that you require, but this data would be more useful if it could be adapted by using calculations.

To improve the presentation of data, you want to concatenate your employees' first name, middle name, and last name fields. You want to make profit more visible by creating a calculated value that subtracts a product's cost from its selling price. Furthermore, you want to create a calculation to improve the categorization of products.

Exercise 2: Creating Measures

Business users have requested the ability to view the following aggregated measures:

- Profit
- Margin
- Previous Year Revenue

Exercise 3: Creating a KPI

Adventure Works wants to achieve year-on-year revenue growth of 20 percent. To track this goal, you plan to create a KPI.

Exercise 4: Implementing a Parent-Child Hierarchy

The CEO at Adventure Works wants to view sales results aggregated by each sales manager, and then drill down to view individual employee sales results.

8: Introduction to Data Analysis Expressions (DAX)

Lab Scenario

A senior business analyst at Adventure Works Cycles wants to perform some detailed analysis of the data in a tabular data model. To accomplish this, you will need to use DAX to enhance the data model.

Lab Review

- Which DAX time intelligence function would you use to calculate a measure based on values for the end of the next quarter?

Question

Which DAX time intelligence function would you use to calculate a measure based on values for the end of the next quarter?

( )Option 1: ENDOFQUARTER

( )Option 2: NEXTQUARTER

( )Option 3: OPENINGBALANCEQUARTER

( )Option 4: PREVIOUSQUARTER

Answer

(√) Option 2: NEXTQUARTER

Module Review and Takeaways

- Review Question(s)

Review Question(s)

Question

Will time intelligence functionality be useful to you? What kinds of time-based analyses might you want to perform? Which are the relevant time intelligence functions that you might use to achieve this?

Answer

Answers will vary, depending on individuals and their experiences.