SQL01 - Reporting Services di SQL Server con DAX

Marco Russo
marco@sqlbi.com - @marcorus
http://www.sqlbi.com/

Grazie a

Sponsor

Why using DAX with Reporting Services

- You can still use MDX
  - There is an MDX editor, no DAX editor by now
- But DAX is more table-friendly
- Easier to create calculated columns
- More similar to SQL than to MDX
- It can be used for Tabular and Multidimensional
- Pros and Cons
  - You are here to learn them

Agenda

- Basic constructs in DAX queries (SUMMARIZE, ADDCOLUMNS)
- Querying Tabular
  - DAX and MDX
  - Parameters in DAX and MDX
    - Difference between Report Builder and Visual Studio
- Querying Multidimensional
  - DAX and MDX
  - Handling Hierarchies
- Naming Convention for Multidimensional in DAX
  - Key values, multi-part keys
- DAX vs MDX: Performance Comparison in SSRS

Basic Constructs in DAX Queries

Start writing queries in DAX

EVALUATE Syntax

EVALUATE is the statement to use to write DAX queries

```
[DEFINE
    { MEASURE <table>[<col>] = <expression> }]
EVALUATE <Table Expression>
[ORDER BY {<expression> [{ASC | DESC}]} [, …]
    [START AT {<value>|<parameter>} [, …]] ]
```

CALCULATETABLE

CALCULATETABLE is often used to apply filters to the query

```
EVALUATE

CALCULATETABLE (
    'Sales Territory',
    [Sales Territory Country] = "United States"
)

ORDER BY 'Sales Territory'[Sales Territory Country]
```

CommunityDays.it

SUMMARIZE

Performs GROUP BY in DAX and optionally computes subtotals

```
EVALUATE

SUMMARIZE (
    'Internet Sales',
    'Internet Sales'[Order Date],
    "Sales Amount", SUM ( 'Internet Sales'[Sales Amount] )
)
```

Source Table

GROUP BY Column(s)

Result Expression(s) added to the source table

CommunityDays.it

ADDCOLUMNS

Adds one or more columns to a table expression.
Each column is computed in a row context

```
EVALUATE

ADDCOLUMNS (
    'Internet Sales',
    "Margin",
    [Sales Amount] - [Total Product Cost]
)
```

Remember the automatic CALCULATE which surrounds any measure calculation

CommunityDays.it

Using CROSSJOIN

CROSSJOIN does what its name suggests: performs a crossjoin between two tables, very useful in querying Tabular

```
EVALUATE

ADDCOLUMNS (
    CROSSJOIN (
        DISTINCT ( 'Product'[Color] ),
        DISTINCT ( 'Product'[Size] )
    ),
    "Products",
        COUNTROWS ( RELATEDTABLE ( Product ) )
)
```

CommunityDays.it

Using GENERATE

GENERATE is the equivalent of APPLY in SQL, will come handy when using many-to-many relationships

```
EVALUATE

GENERATE (
    'Product Category',
    RELATEDTABLE ( 'Product Subcategory' )
)
```

CommunityDays.it

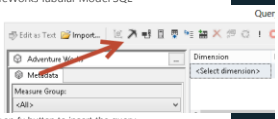
Querying Tabular

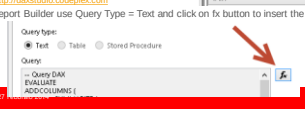
Create reports in MDX and DAX on a SSAS Tabular (TAB) model in Visual Studio and Report Builder

CommunityDays.it

Data Source and Datasets

- Data Source: same connection string for MDX and DAX
  Data Source=.\tabular;Initial Catalog="AdventureWorks Tabular Model SQL 2012";
- Dataset
  - MDX: can use the MDX Query Builder
  - DAX: must use DMX Editor in Visual Studio
    - No support for DAX queries in the editor
    - Use SSMS or DAX Studio instead
      - http://daxstudio.codeplex.com
  - In Report Builder use Query Type = Text and click on fx button to insert the query





Compare Queries on Tabular Model

MDX

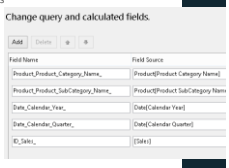
```
SELECT

[Measures].[Internet Total Sales]
   ON COLUMNS,

NON EMPTY
   [Product].[Category].[SubCategory]
   * [Date].[Calendar].[Year]
   * [Date].[Calendar Quarter].[Calendar Quarter]
   ON ROWS

FROM [Model]

CELL PROPERTIES VALUE
```

DAX

```
EVALUATE
ADDCOLUMNS (
   SUMMARIZE (
      'Internet Sales',
      'Product'[Product Category Name],
      'Product'[Product SubCategory Name],
      'Date'[Calendar Year],
      'Date'[Calendar Quarter]
   ),
   "Sales", [Internet Total Sales]
)
ORDER BY
   'Product'[Product Category Name],
   'Product'[Product SubCategory Name],
   'Date'[Calendar Year],
   'Date'[Calendar Quarter]
```


Visual Studio Issues in DAX Queries

- Dataset Field Names
  - Each field name include table and column names
  - Added columns are recognized as ID and not as measures
  - Change names of the fields after importing the data set
    - Better not using the Wizard
    - In Report Builder you can create the Dataset first, then use the Matrix Wizard
- Default aggregation is wrong
  - In Visual Studio, First is used for measure results instead of Sum
  - In Report Builder no aggregation is recognized and you have to set one (usually Sum)



Querying Multidimensional

Create reports in MDX and DAX on a SSAS Multidimensional (MD) model

Connection String

- MDX: Just the server and the database names
  - Cube name can be part of the query
- DAX: Server, Database **and Cube** names
  - Without cube name, DAX queries will be interpreted as DMX, not DAX
  - Use Dialog Box in SSMS and DAX Studio
- Sample connection string:
  Data Source=localhost; Initial Catalog=AdventureWorksDW2012; Cube=Adventure Works
  Cube parameter not supported in SSRS Advanced Properties – put it directly in connection string

Compare Queries on Multidimensional Model

MDX

```
SELECT

[Measures].[Internet Sales Amount] ON
COLUMNS,

NON EMPTY
   [Product].[Product Categories].[Subcategory]
   * [Date].[Calendar Year].[Calendar Year]
   * [Date].[Calendar Quarter of Year].[Calendar
Quarter of Year] ON ROWS

FROM [Adventure Works]

CELL PROPERTIES VALUE
```

DAX

```
EVALUATE
ADDCOLUMNS (
   SUMMARIZE (
      'Internet Sales',
      'Product'[Category.Key0], 'Product'[Category],
      'Product'[Subcategory.Key0], 'Product'[Subcategory],
      'Date'[Calendar Year.Key0], 'Date'[Calendar Year],
      'Date'[Calendar Quarter of Year]
   ),
   "Sales", [Internet Sales Amount]
)
ORDER BY
   'Product'[Category],
   'Product'[Subcategory],
   'Date'[Calendar Year],
   'Date'[Calendar Quarter of Year]
```

Parameters in DAX Queries

How to apply parameters to DAX queries in Reporting Services

Parameters in DAX Queries

- Not supported in MDX Editor, requires either:
  - DMX Editor
    - Not available in Report Builder
  - Dataset properties
    - Parameters always string in field refresh – use cast to INT for numeric columns
- Write each parameter using @ prefix
  - Multiple values not supported (no IN clause in DAX)
- http://sqlblog.com/blogs/marco_russo/archive/2012/01/05/use-parameters-in-your-dax-queries.aspx

From MDX to DAX

Compare the syntax in the two languages

Multidimensional (MDX)	Tabular (DAX)
Cube	Model
Cube Dimension	Table
Attribute Keys and Names	Columns
Measure Group	Table
Measure	Measure
Measure with No Measure Group	Measure in table «Measures»
Measure Group → Dimension Relationship	Relationship
Perspective	Perspective
KPI	KPI
User or Parent/Child Hierarchy	Hierarchy

Columns translation

- You see a column in DAX for each:
  - Attribute Name
  - Attribute Key Column
    - One column for single key attribute, two columns if attribute key made by 2 columns
  - Attribute Value
- Example, attribute [Year Month] shown as:
  - [Year Month] – name – «2013 January»
  - [Year Month.Key0] – Year Number – 2013
  - [Year Month.Key1] – Month Number – 1
  - [Year Month.Value] – The value attribute (if defined) – 201301

User and Parent-Child Hierarchies

- Parent-child hierarchies are flattened
  - All values, keys and properties repeated for every level
  - Original unique key not available in a single field
    - Add a separate field in the dimension if necessary, repeating the key value
- Complete attribute keys for hierarchy levels
  - Specify the complete path of keys in a SUMMARIZE query if the attribute key for the level is not unique and depends on parent values

Performance Comparison

Find when DAX is faster than MDX

DAX Might Be Faster

- DAX does less than MDX
  - No sort unless requested in ORDER BY
  - No formatting
    - Use CELL PROPERTIES VALUE in MDX to get only values like DAX
  - No hierarchies
- DAX might be faster than MDX
  - Maybe because DAX does less than MDX
  - Additional features of MDX might be not required in Reporting Services
    - Do you really need the format?
- Do your own tests…

Top 10 Sales by Customer/Date

MDX – 187/124 msec (*)

```
SELECT
    [Measures].[Internet Sales Amount]
        ON COLUMNS,

    TOPCOUNT (
        NONEMPTY (
            [Customer].[Customer].[Customer]
            * [Date].[Date].[Date],
            [Measures].[Internet Sales Amount]
        ),
        10,
        [Measures].[Internet Sales Amount]
    ) ON ROWS

FROM [Adventure Works]

CELL PROPERTIES VALUE
```

DAX – 187/124 msec (*)

```
EVALUATE
TOPN (
    10,
    ADDCOLUMNS (
        FILTER (
            CROSSJOIN (
                SUMMARIZE (
                    Customer,
                    Customer[Customer],
                    Customer[Customer.Key0]
                ),
                VALUES ( 'Date'[Date.Key0] )
            ),
            NOT ISBLANK ( [Internet Sales Amount] )
        ),
        "Sales", [Internet Sales Amount]
    ),
    [Sales], 0,
    'Date'[Date.Key0], 0,
    Customer[Customer], 1
),
ORDER BY
    'Date'[Date.Key0] DESC,
    Customer[Customer] ASC
```

(*) average time cold cache / warm cache

SUMMARIZE vs. FILTER (CROSSJOIN (), NOT ISBLANK())

DAX equivalent to MDX should use:

```
FILTER (
    CROSSJOIN (
        VALUES ( Dimension1[Attribute1] )
        VALUES ( Dimension2[Attribute2] )
    ),
    NOT ISBLANK( [Measure] )
)
```

Optimization with SUMMARIZE for implicit measures (**don't use with arbitrary measures**!):

```
ADDCOLUMNS (
    SUMMARIZE (
        Fact,
        Dimension1[Attribute1],
        Dimension2[Attribute2],
    ),
    "Measure", [Measure]
)
```


Top 10 Sales by Customer/Date with DAX Shortcut

MDX – 187/124 msec (*)

```
SELECT

    [Measures].[Internet Sales Amount]
        ON COLUMNS,

    TOPCOUNT (
        NONEMPTY (
            [Customer].[Customer].[Customer]
            * [Date].[Date].[Date],
            [Measures].[Internet Sales Amount]
        ),
        10,
        [Measures].[Internet Sales Amount]
    ) ON ROWS

FROM [Adventure Works]
```

DAX – 124/31 msec (*)

```
EVALUATE
TOPN (
    10,
    ADDCOLUMNS (
        SUMMARIZE (
            'Internet Sales',
            Customer[Customer.Key0],
Customer[Customer],
            'Date'[Date.Key0], 'Date'[Date]
        ),
        "Sales", [Internet Sales Amount]
    ),
    [Sales], 0,
    'Date'[Date.Key0], 0,
    Customer[Customer.Key0], 1
),
ORDER BY
    [Sales] DESC,
    'Date'[Date.Key0] DESC,
    Customer[Customer] ASC
```

(*) average time cold cache / warm cache

Large resultset - Sales by Customer (18,000 rows)

MDX – 452/374 msec (*)

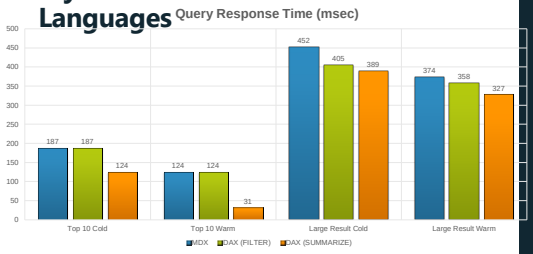
```
SELECT

    Measures.[Internet Sales Amount]
        ON COLUMNS,

    NON EMPTY
        [Customer].[Customer].[Customer]
            ON ROWS

FROM [Adventure Works]

CELL PROPERTIES VALUE
```

DAX – 389/327 msec (*)

```
EVALUATE
ADDCOLUMNS (
    SUMMARIZE (
        'Internet Sales',
        Customer[Customer],
        Customer[Customer.Key0]
    ),
    "Sales", [Internet Sales Amount]
)
ORDER BY Customer[Customer]
```

(*) average time cold cache / warm cache

Try Your Own Critical Queries in Both Languages



Complexity Comparison

Compare readability between DAX and MDX

Flexibility in Relationship Usage

- MDX requires relationships in data model
- Limited flexibility in joining data between unrelated tables in a single result
  - MDX requires FILTER in measure
  - DAX allows generation of a single row context
- DAX might be easier to write and should be usually faster

MDX vs DAX Querying Unrelated Tables

MDX

```
WITH
   MEMBER Measures.[Reseller Sales] AS
      ( [Reseller Sales Amount],
        FILTER (
           Geography.Country.Country,
           Geography.Country.CurrentMember.Name
              = Customer.Country.CurrentMember.Name
        ).Item(0)
      )

SELECT

   { [Internet Sales Amount], [Reseller Sales] } ON COLUMNS,

   Customer.Country.Country  ON ROWS

FROM [Adventure Works]
```

DAX

```
EVALUATE
ADDCOLUMNS (
   GENERATE (
      VALUES ( Customer[Country.Key0] ),
      FILTER (
         ALL ( Geography[Country.Key0] ),
         Geography[Country.Key0] = Customer[Country.Key0]
      )
   ),
   "Internet Sales", [Internet Sales Amount],
   "Reseller Sales", [Reseller Sales Amount]
)
```


Conclusions

Choose between MDX and DAX writing your reports

Q&A

Tutto il materiale di questa sessione su
http://www.communitydays.it/

Lascia il feedback su questa sessione,
potrai essere estratto per i nostri premi!

Seguici su
Twitter @CommunityDaysIT
Facebook http://facebook.com/cdaysit
#CDays14