Berlin SQL Server User Group | 10 February, 2015

# Mastering the Excel CUBE Functions

Bitwise Solutions

people + technology + **training** = success

# Presenter Introduction

Peter Myers

BI Expert – Bitwise Solutions

BBus, SQL Server MCSE, MCT, SQL Server MVP (since 2007)

Experienced in designing, developing and maintaining Microsoft database and application solutions, since 1997

Focuses on education and mentoring

Based in Melbourne, Australia

✉ peter.myers@bitwisesolutions.com.au

in http://www.linkedin.com/in/peterjsmyers

Microsoft® MVP Most Valuable Professional

# Session Objectives

To introduce each the Excel CUBE functions

To describe when to use, and not to use, the CUBE functions

To provide recommended practices

To suggest advanced scenarios

To demonstrate how to create compelling and interactive reports based on the CUBE functions

# Session Outline

Introducing the CUBE Functions

When to Use the CUBE Functions

When Not to Use the CUBE Functions

Recommended Practices

Advanced Scenarios

Demonstrations

# Introducing the CUBE Functions

Excel 2013 includes seven CUBE functions – available since Excel 2007

CUBE functions allow creating Excel report layouts cell-by-cell based on an Analysis Services BI Semantic Model (BISM)

- Only works for data models that can be queried with Multidimensional Expressions (MDX)
- Tabular data models in DirectQuery mode cannot be queried by using MDX

Reports can allow user input by using slicers, timelines and cell values (simulating parameters)

CUBE function arguments can be defined by using expressions – both by using Excel or MDX expressions

# Introducing the CUBE Functions

## Continued

Data model metadata, members, sets, tuples and formats are cached to minimize impact on the database

Using the CUBE functions can work well for some report designs, and not so well for others

CUBE function report layouts can be created by converting a PivotTable to formulas – it is a one-way conversion, and can be helpful for beginners learning to write CUBE formulas

# Demonstrations

Converting a PivotTable to Formulas

# Introducing the CUBE Functions

## Continued

Excel includes seven CUBE functions:

- CUBEMEMBER
- CUBESET
- CUBESETCOUNT
- CUBERANKEDMEMBER
- CUBEVALUE
- CUBEKPIMEMBER
- CUBEMEMBERPROPERTY

# Introducing the CUBE Functions

## Continued

Most CUBE functions require Connection and Caption parameters

Connection: Text string of the name of the workbook connection to the data model

- Create workbook connections to Analysis Services data models, or use the PowerPivot data model workbook connection “ThisWorkbookDataModel”
- Tip: Name the workbook connections appropriately – and concisely – by using the Friendly Name property

Caption: Text string that is displayed in the cell instead of the default caption, if one is defined, from the data model

- Use this argument to present a user-friendly label

# Introducing the CUBE Functions

```
CUBEMEMBER(connection, member_expression, caption)
```

Returns a member or tuple from the data model

- member_expression: Text string of an MDX expression that evaluates to a unique member in the data model. Alternatively, it can be a tuple, specified as a cell range or an array constant.
- caption: When a tuple is returned, the caption used is the one for the last member in the tuple

Examples of member_expression inputs:

- [Date].[Calendar].[CY2013]
- A$1
- (A2, B2, C2)
- A2:C2

# Introducing the CUBE Functions

```
CUBESET(connection, set_expression, caption, sort_order, sort_by)
```

Returns a set of members from the data model

- set_expression: A set expression
- sort_order: Integer value specifying the rank to return
- sort_by: Text string of the value by which to sort, typically a measure

Note:

- The sort parameters are ideal for passing in user preference (parameters)
- Excel slicers (representing multi-member selection) can be passed in to the set_expression argument

# Introducing the CUBE Functions

```
CUBESETCOUNT(set)
```

Returns the number of items in a set

- set: Text string or an Excel expression that evaluates to a set defined by the CUBESET function

Note:

- This is the only CUBE function that does not require a connection
  - It does require the result of a CUBESET function to be passed in
- Can be used to determine the number of members selected in an Excel slicer
- Can be used to determine report layout (to simulate dynamic rows/columns)

# Introducing the CUBE Functions

```
CUBERANKEDMEMBER(connection, set_expression, rank, caption)
```

Returns the nth, or ranked, member in a set

- set_expression: A set expression, a CUBESET function, or reference to a CUBESET function
- rank: Integer value specifying the rank to return

Note:

- The first member is rank 1
- Requesting a rank beyond the set member count will produce an error
  - Can work around this by using IFERROR(CUBERANKEDMEMBER(…), 0)
- To return the bottom n values, use the sort_order and sort_by arguments of the CUBESET function to reverse the order of the set
- Works best when cube sets have a fixed, and known, number of members, such as the top sales performer, the top 10 students, or the 12 months of a year
- Use the ROW or COLUMN function to build a “template function” to copy and paste into a range

# Introducing the CUBE Functions

```
CUBEVALUE(connection, member_expression1, member_expression2…)
```

Returns an aggregated value from the data model

- member_expression: Text string of an MDX expression that evaluates to a member or tuple within the data model. Alternatively, it can be a set defined with the CUBESET function.

Note:

- Each member_expression will form a tuple
- If at least one element within the tuple is invalid, CUBEVALUE returns a #VALUE! error value
- Excel slicers and timelines can be used as member expressions

# Introducing the CUBE Functions

```
CUBEKPIMEMBER(connection, kpi_name, kpi_property, caption)
```

## Returns a KPI property from the data model

- kpi_name: Text string of the name of the KPI in the data model
- kpi_property: Integer value specifying the KPI component to be returned, and can be one of the following:

| Integer | Enumerated constant |
|---|---|
| 1 | KPIValue |
| 2 | KPIGoal |
| 3 | KPIStatus |
| 4 | KPITrend |
| 5 | KPIWeight |
| 6 | KPICurrentTimeMember |

Note:

- Status and Trend values represent a normalized value, typically:
  - Status: -1 (off track), 0 (slightly off track) and 1 (on track)
  - Trend: -1 (deteriorating), 0 (no change) and 1 (improving)
- Auto-complete does not list the available status or trend values – you must know them – consult the data model documentation ☺
  - Use Excel conditional formatting to hide values and represent them as color scales or icon sets

# Introducing the CUBE Functions

```
CUBEMEMBERPROPERTY(connection, member_expression, property)
```

Returns the value of a member property from a cube

- member_expression: Text string of an MDX expression of a member within the cube, or a reference to an Excel formula that returns a member (CUBEMEMBER or CUBERANKEDMEMBER)
- property: Text string of the name of the property returned, or a reference to a cell that contains the name of the property

Note:

- Useful for introducing properties like phone number, email address, price, etc.
- Auto-complete does not list the available member properties – you must know them – consult the data model documentation ☺
- Tabular BISM, including Power Pivot, cannot define member properties

# Demonstrations

Producing a Dashboard with the CUBE Functions

# When to Use the CUBE Functions

When dashboard-style layouts are required, especially when published to SharePoint and rendered by Excel Services

To reproduce reports that must look and behave similarly to legacy reports

To overcome limitations of PivotTables and PivotCharts

- Create asymmetric reports (this year vs. last year) – though this could possibly be achieved with a custom set
- GETPIVOTDATA is not an elegant or easy way to retrieve values from a PivotTable

When customized MDX is required

To integrate data from multiple data models

To produce easier logic by using Excel formula, instead of MDX or DAX in the data model

# When Not to Use the CUBE Functions

When a PivotTable or PivotChart, or Power View, could deliver the same result

- These report types provide structured exploration and rich interactivity
- These report types are easier to create

When interactivity is required: sorting, filtering, etc.

- Some of these can be simulated with CUBE formulas by using expressions, but this takes more development time and effort

When sets with variable member counts need to be displayed – the CUBERANKEDMEMBER function will return an error if requesting a rank beyond the set size

- You can work around this by testing with the IFERROR function

# Recommendations

Learn the CUBE functions by starting with a PivotTable and converting it to formulas

Know MDX to customize beyond the basics – you can work beyond the Auto-complete options

Adopt a methodical design approach to help implement change and to troubleshoot errors

- Create a “Globals” worksheet to define CUBE formulas and annotate them
- Hide rows, columns or worksheets that contain CUBE formulas
- Avoid complex CUBE formulas – break them down to smaller formulas and nest them

Strive for parameter-driven designs to allow for variability and user interactivity

- Define Excel named ranges to simplify building member and set expressions
- Use “dynamic” expressions based on Excel formula
- Use slicers and timelines to allow users to interactively filter the report

# Recommendations

## Continued

Use template functions to copy and paste into ranges

- This is especially useful for the CUBERANKEDMEMBER and CUBEVALUE functions

Define captions to allow the CUBE formulas to blend into the report

If the data model definition changes, be sure to refresh the workbook connection to update the cache

- This is usually important when modifying the embedded PowerPivot data model

Expressions cannot be greater than 255 characters, which is the limit for an argument to a function in Excel

- To overcome the text length limit, use references to cells that contain the expression, which can have a text length size of up to 32,767 characters

# Advanced Scenarios

Produce dynamic layouts by using macros

- However, these reports cannot be rendered by Excel Services

Produce reports that integrate members and values from multiple data models

- Combine the enterprise data model with your PowerPivot data model
- Extend the corporate data stores with custom data and/or calculations
- Tip: It pays off to design data models with a consistent structure (dimension, hierarchy, member names, etc.) – but this is not necessary, just easier

Use the Table Analysis Tools Excel data mining add-in based on an Excel table of CUBE formulas

Create an Excel table based on CUBE formulas as a source for other visualizations:

- Power Map
- Visio Services
- Apps for Office

# Demonstrations

Advanced Scenarios with the CUBE Functions

# Resources

Book: “Microsoft SQL Server 2008 MDX Step by Step”

- Publisher: Microsoft Press
- Author: Bryan Smith

Channel 9: TechEd New Zealand 2013

- Video: “Mastering the Excel CUBE Functions in Excel 2013”
- http://channel9.msdn.com/Events/TechEd/NewZealand/2013/DBI304

# Thank You

Bitwise Solutions

people + technology + **training** = success