

Analysis Services Feature Comparison

Charlotte Business Intelligence Group | March 6, 2017

Vincent Staropoli

@vstaropoli | [linkedin.com/in/vincentstaropoli](https://www.linkedin.com/in/vincentstaropoli)

Agenda

- Introduction
- Why have this discussion?
- Primer and key concepts
- Many to many relationships
- Row level security
- Scoped metrics

Introduction

- Solution Architect with BlueGranite, Inc
- Data Professional focusing on Data Warehousing and Advanced Analytics
- SQL Server and Microsoft Data Platform Enthusiast



Microsoft
CERTIFIED

Solutions Expert

Data Management and
Analytics

Microsoft
CERTIFIED

Solutions Expert

Cloud Platform and
Infrastructure

Microsoft
CERTIFIED

Solutions Expert

Business Intelligence

Microsoft
CERTIFIED

Solutions Expert

Data Platform

Why have this discussion?

- Analysis Services has been a key link between users and data
- The tabular model is built into Excel, Power BI, SharePoint, and O365
- Significant differences between how Multidimensional and Tabular work
- The core business problem we're trying to solve is the same:
 - How can we provide a consistent and easily understood pathway to complex data?
- The concepts we will cover are a natural extension of the models you've been building in tools like Power BI

Where does this apply?

Feature	Multidimensional	Tabular
SQL Server Analysis Services	X	X
Azure Analysis Services		X
Power BI Desktop		X
Excel Power Pivot		X
Oracle Essbase	X	
SAP Business Warehouse	X	
Microstrategy Intelligent Cube	X	

Tabular models are based on the xVelocity engine (formerly VertiPaq) which used column-based storage and compression.

Tabular models are not the same as columnar databases, however they compress data similarly

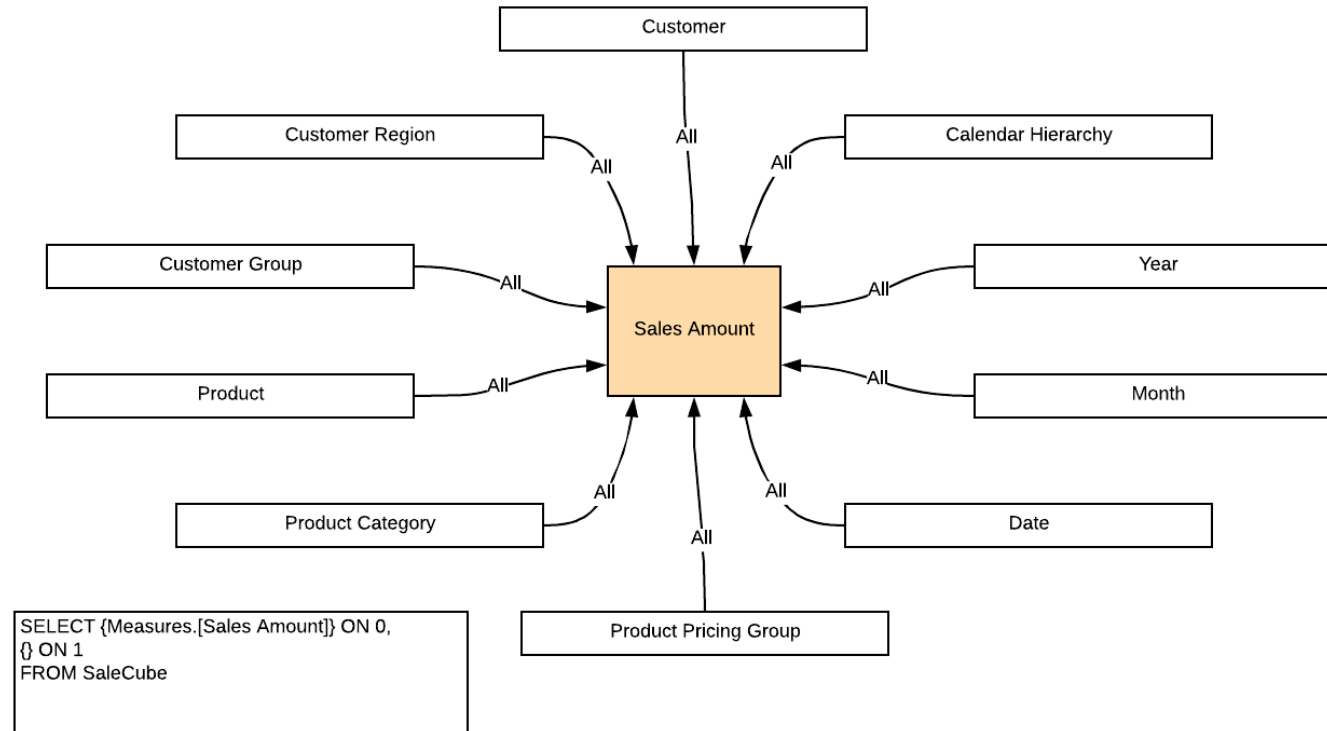
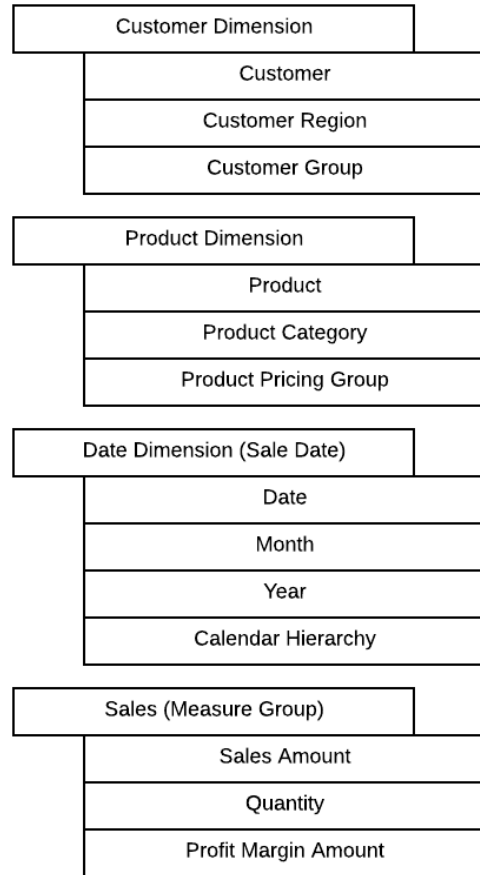
Core Concepts

- Analysis Services Multidimensional
 - Traditional OLAP database
 - Facts = Measures, Dimensions = Dimensions
 - Key Attribute and Dimension Attribute Relationships
 - MDX query (Multi-dimensional Expressions)
- Analysis Services Tabular
 - Tabular database powered by columnar storage and compression
 - All tables can have calculated columns or measures
 - Faster time to deliver as compared to Multidimensional, more flexibility
 - DAX (Data Analytics Expressions)

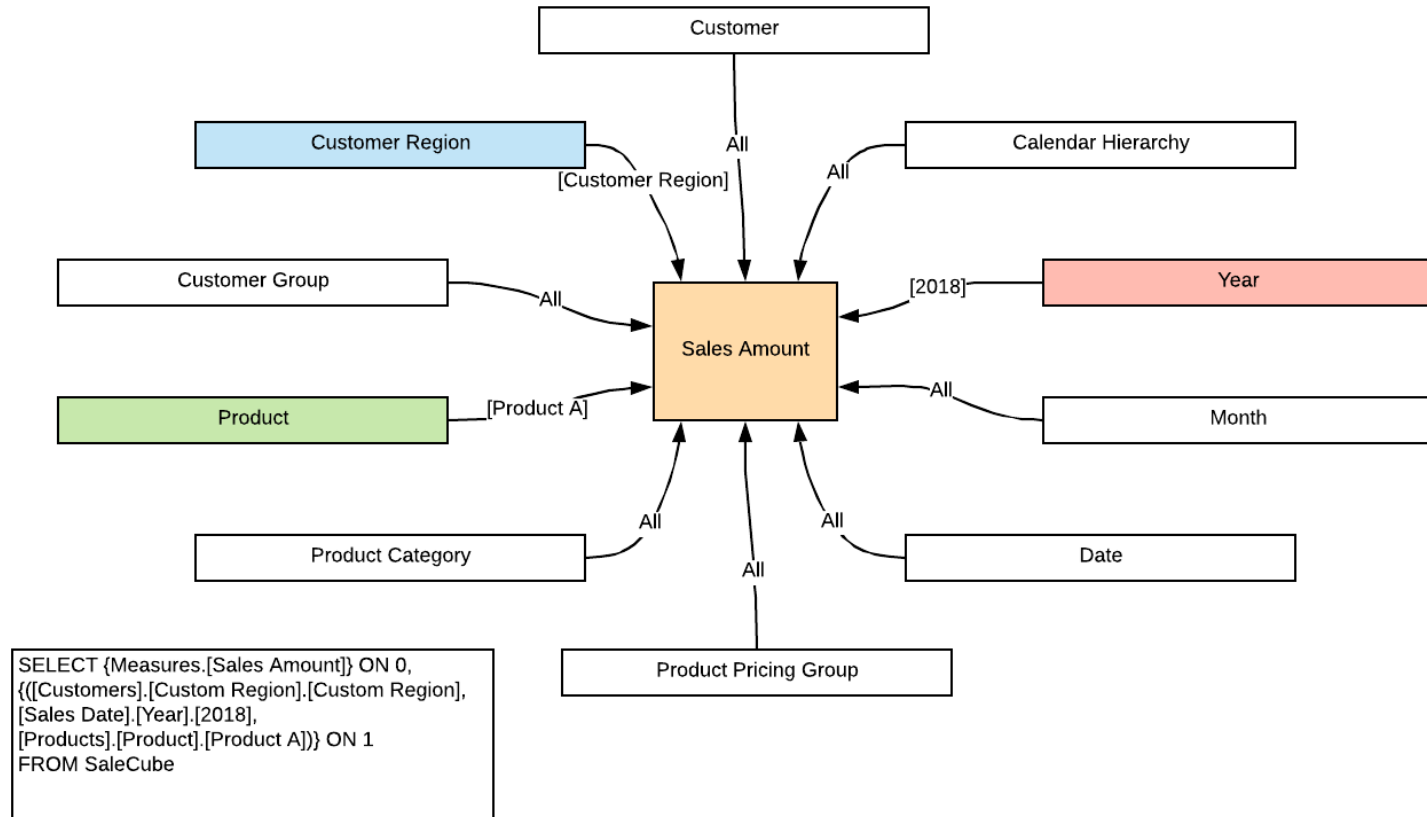
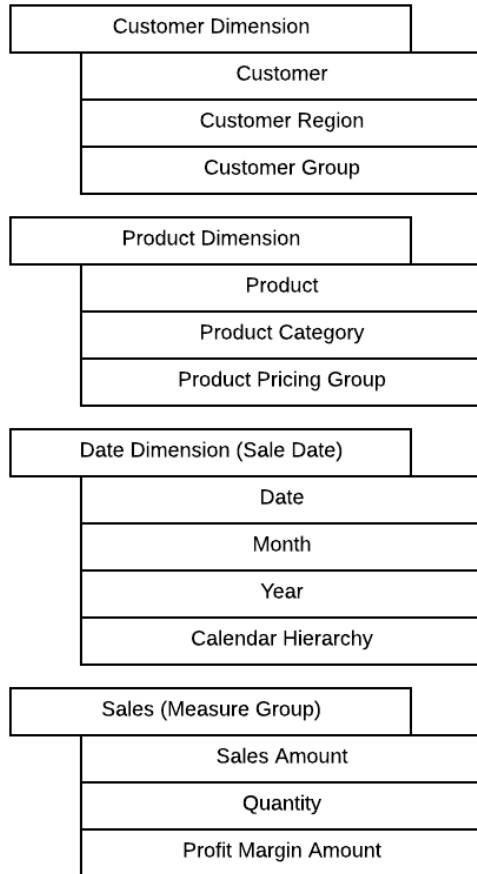
MDX vs DAX

- MDX
 - "Give me the values that exist at this intersection"
 - Cubes, sub-cubes, and slicers
 - All dimensions are involved with each query
 - There is ALWAYS a measure involved
- DAX
 - "What rows exist after applying these filters"
 - Tables, filters, and slicers
 - Queries only access necessary tables and calculations
 - Cross or multi-table relationships can get complicated

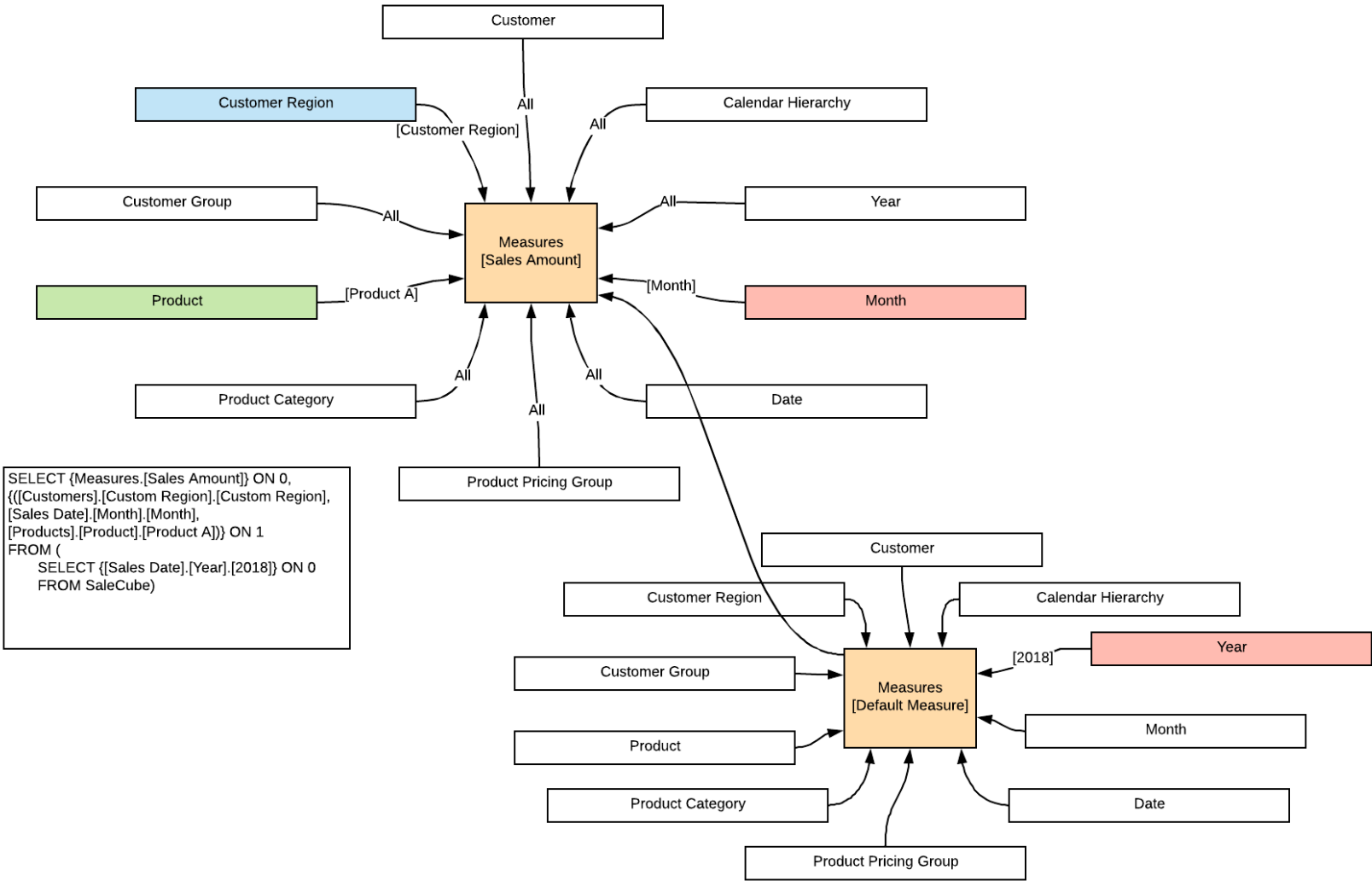
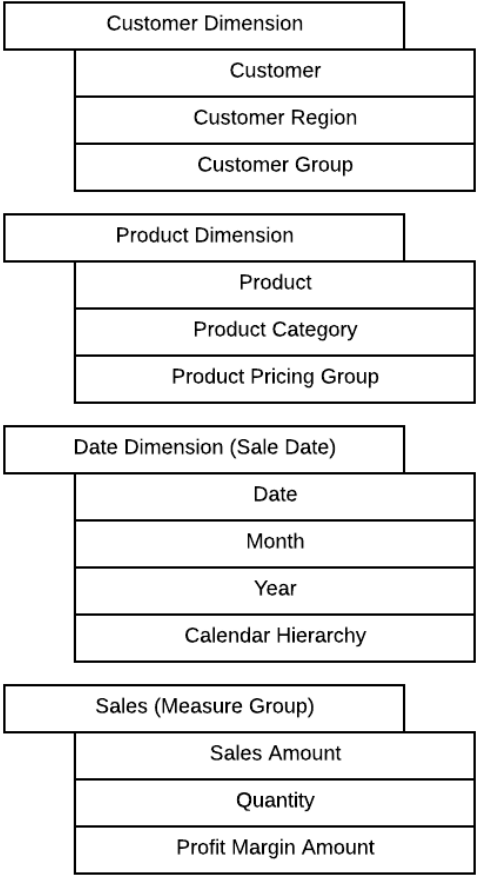
Primer - MDX Query



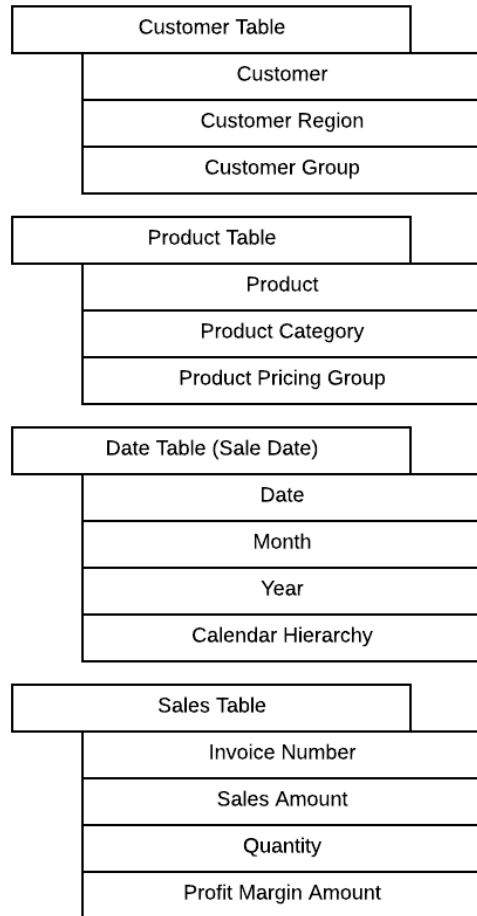
Primer - MDX Query



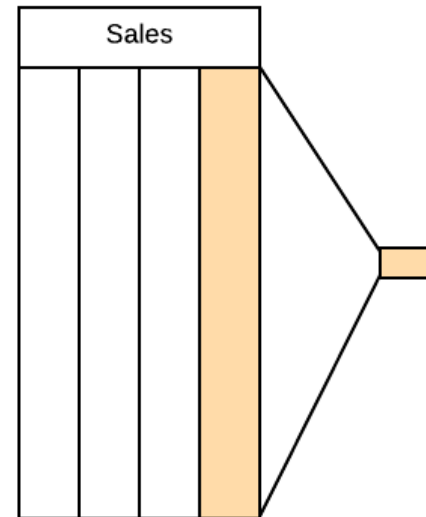
Primer - MDX Query



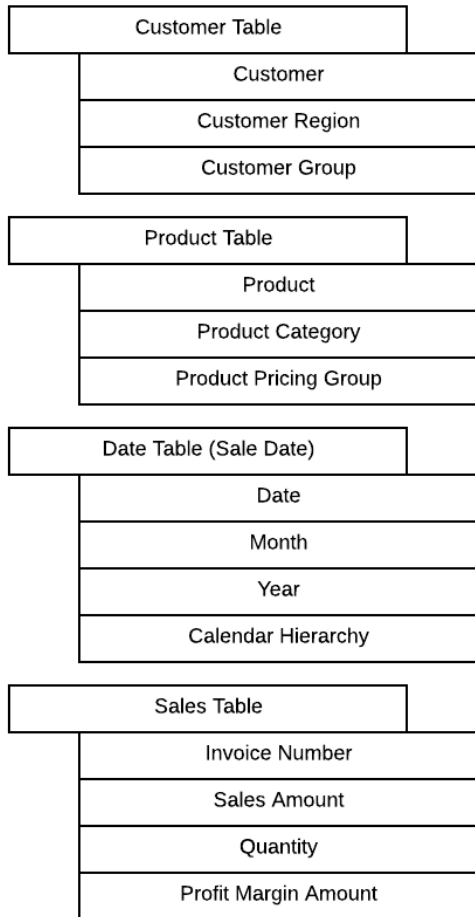
Primer - DAX Query



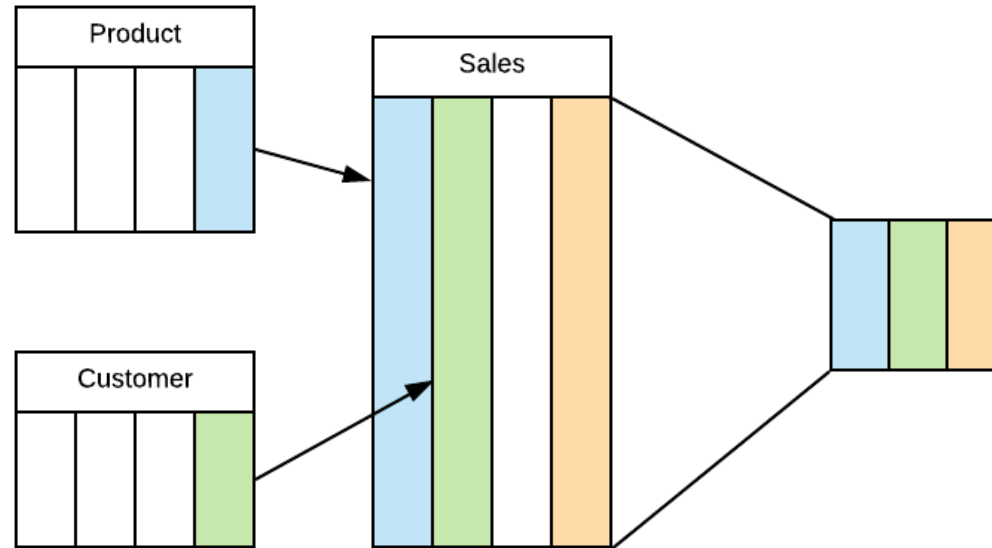
```
EVALUATE  
SUMMARIZECOLUMNS(  
    "Total Sales",  
    SUM('Sales'[Sales Amount])  
)
```



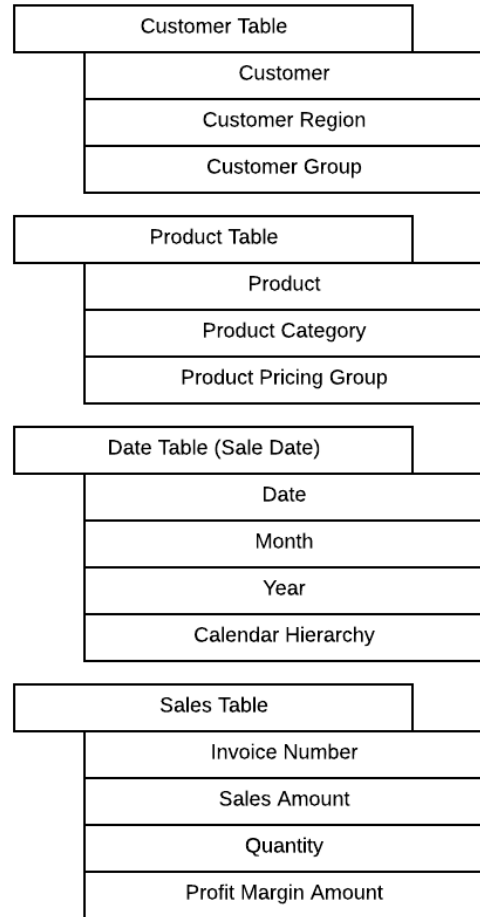
Primer - DAX Query



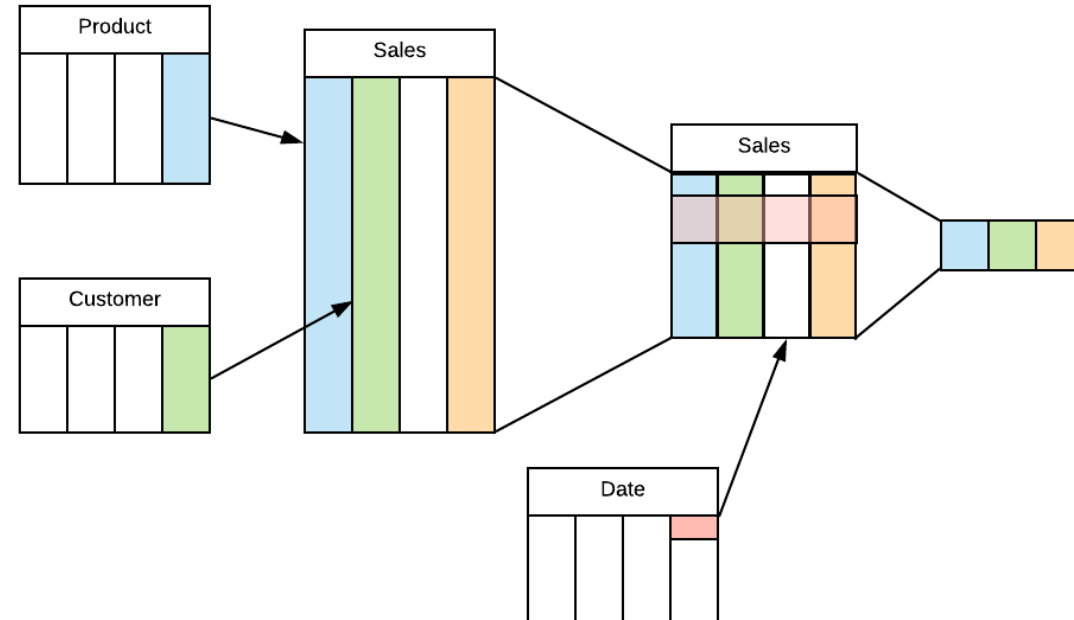
```
EVALUATE  
SUMMARIZECOLUMNS(  
Product[Product Category],  
Customer[Customer Region]  
"Total Sales",  
SUM('Sales'[Sales Amount])  
)
```



Primer - DAX Query



```
EVALUATE  
SUMMARIZECOLUMNS(  
    Product[Product Category],  
    Customer[Customer Region]  
    "Total Sales",  
    CALCULATE(  
        SUM('Sales'[Sales Amount]),  
        Date[Year] = 2018  
    )  
)
```



Many to Many Relationships

- Commonly used advanced modeling approach
 - Dates to Time Periods
 - Sales People to Regions
 - Users to Data Security Criteria
- Model the User Access Dimension and M2M Table
 - Fact-less Fact facilitates Relationship
 - Also called a bridge or relationship table
 - Leaf level values are best for key relationships

Attribute Relationships

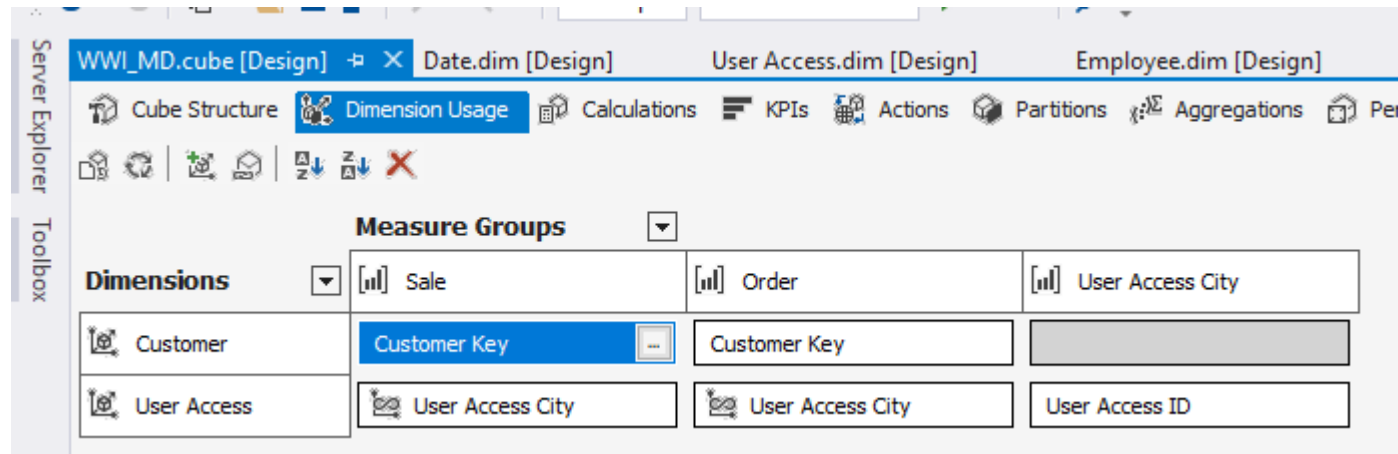
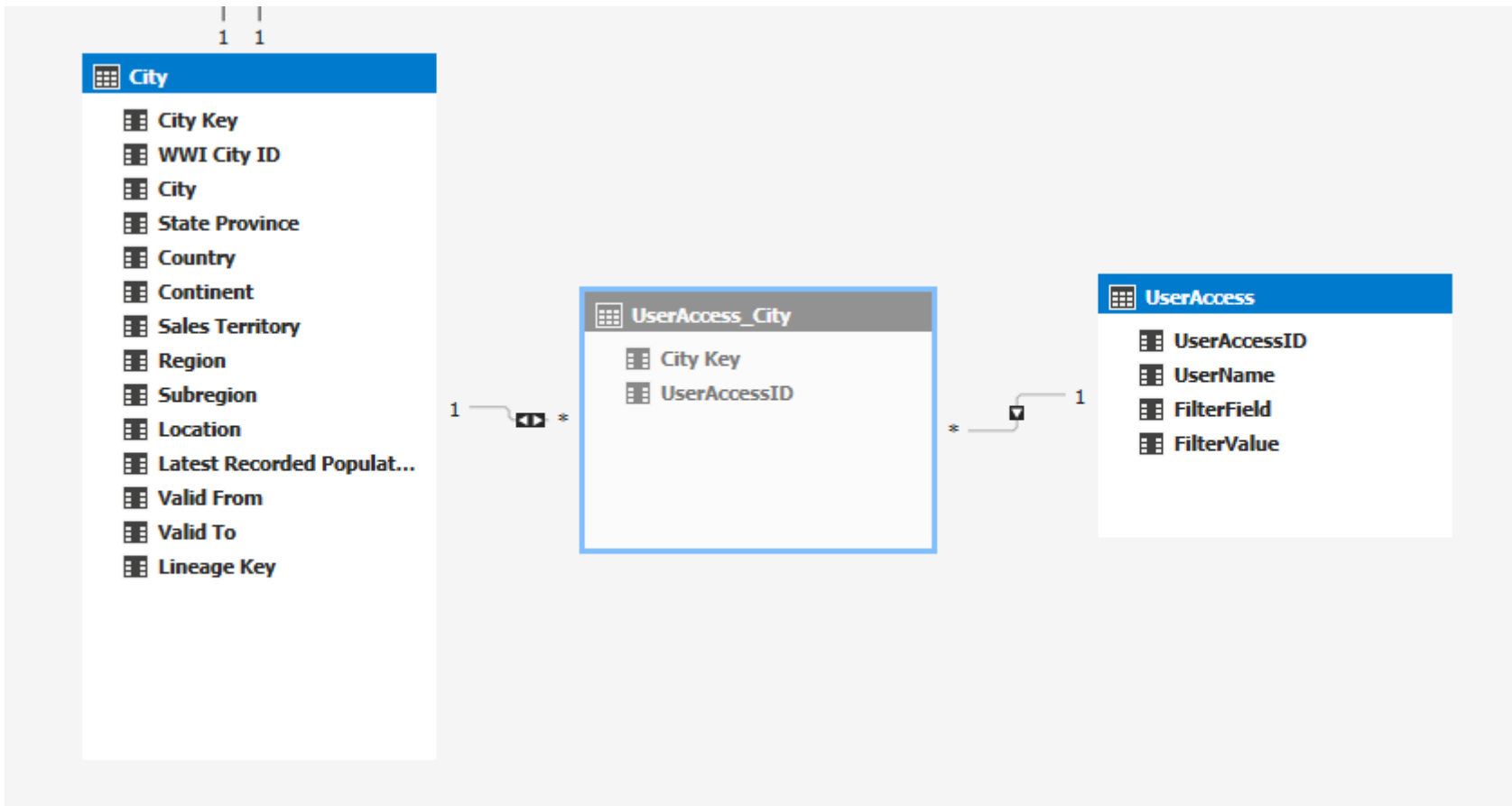


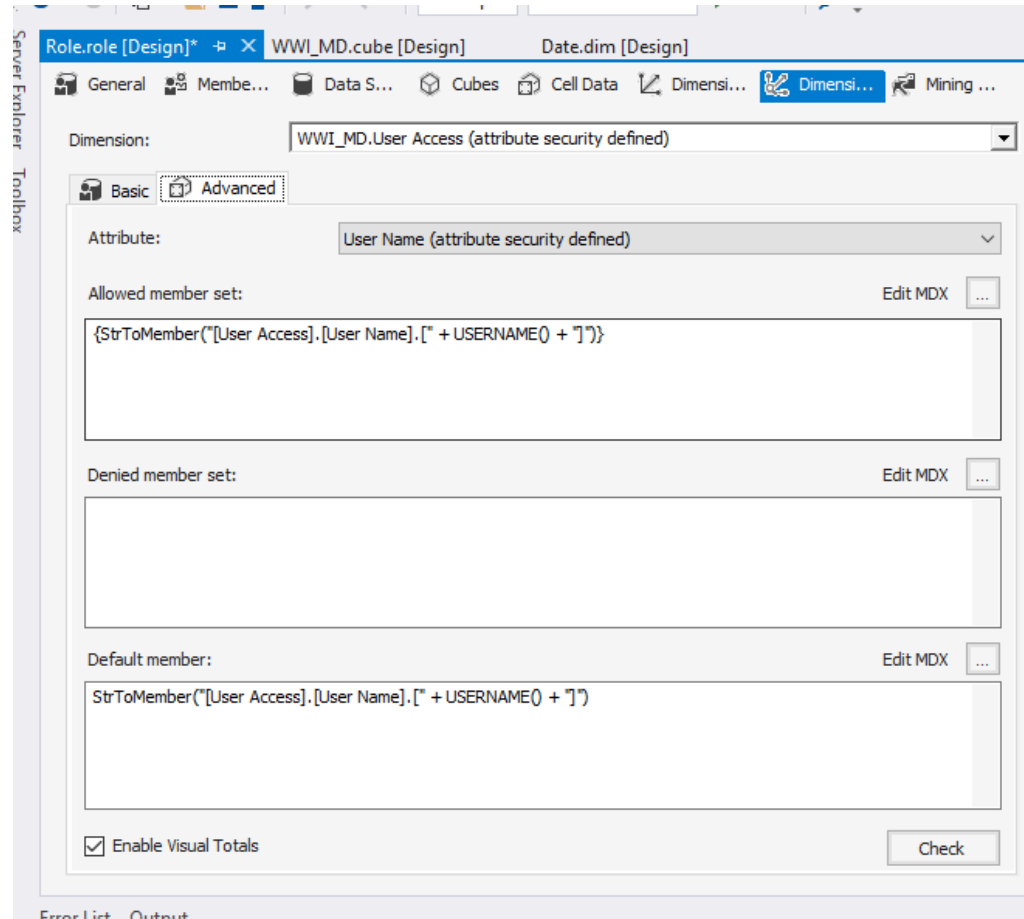
Table Relationships



Implementing Row Level Security

- Use the User Access dimension for filtering data
 - Happens without user intervention
 - Centralizes security management
 - Principles are similar in each model, implementation varies
- “The Omnipresent Slice”

Multidimensional Role Configuration



Tabular Role Configuration

Role Manager

Specify the roles for the tabular project. Roles define a group of users with a set of permissions on the Analysis Services database.

Name	Permissions	Description
Role	Read	

New Copy Delete

Details - Role

Row Filters Members Tables and Columns

Specify DAX expressions that return Boolean values. Only rows that match the specified filters are visible to users in this role.

Table	DAX Filter
Stock Holding	
Transactions	
UserAccess_City	
UserAccess	=UserAccess[UserName]=USERNAME()
TimeMeasures	

OK Cancel

Scoping Metrics

- Multidimensional cubes and MDX provide a SCOPE command
 - Changes the behavior of a member or cell value
 - Very powerful, can be confusing to troubleshoot
 - No **real** equivalent in Tabular database
- Tabular cubes have functions to apply row filter and context changes
 - Check for presence of either an attribute member or value
 - Use AggregateX(table, expression) calculations to emulate behavior
 - First you create the context, then you aggregate with the desired calculation

Multidimensional Calculation Script

```
[-] Create Member
    CurrentCube.[Date].[Calendar Date Calculations].[Prior Month]
    As "NA";
```

```
Scope(
{
    [Measures].[Total Excluding Tax],
    [Measures].[Total Including Tax],
    [Measures].[Quantity],
    [Measures].[Total Excluding Tax - Sale],
    [Measures].[Profit],
    [Measures].[Sale Count],
    [Measures].[Ordered Quantity],
    [Measures].[Purchase Count],
    [Measures].[Quantity - Order],
    [Measures].[Total Excluding Tax - Order],
    [Measures].[Total Including Tax - Order],
    [Measures].[Order Count]
}
);
```

```
/*Prior Month*/
(
    [Date].[Calendar Date Calculations].[Prior Month],
    [Date].[Calendar Year].[Calendar Year].Members,
    [Date].[Date].Members
)
=
Aggregate(
    { [Date].[Calendar Date Calculations].[Current Date] }
    *
    ParallelPeriod(
        [Date].[Calendar].[Calendar Month],
        1,
        [Date].[Calendar].CurrentMember
    )
);
```

Tabular Measure

```
OrderCountCompare:=  
IF(SELECTEDVALUE(TimeMeasures[TimeMeasureName], "Current")="Prior Month"  
    ,SUMX(PREVIOUSMONTH('Date'[Date]),[OrderCount])  
    ,IF(SELECTEDVALUE(TimeMeasures[TimeMeasureName], "Current")="Prior Month % Change"  
        ,FORMAT(DIVIDE([OrderCount]-SUMX(PREVIOUSMONTH('Date'[Date]),[OrderCount]),SUMX(PREVIOUSMONTH('Date'[Date]),[OrderCount])), "Percent")  
        ,IF(SELECTEDVALUE(TimeMeasures[TimeMeasureName], "Current")="Month Over Month Growth"  
            ,[OrderCount]-SUMX(PREVIOUSMONTH('Date'[Date]),[OrderCount])  
            ,IF(SELECTEDVALUE(TimeMeasures[TimeMeasureName], "Current")="Same Month Last Year"  
                ,SUMX(SAMEPERIODLASTYEAR('Date'[Date]),[OrderCount])  
                ,IF(SELECTEDVALUE(TimeMeasures[TimeMeasureName], "Current")="Same Month Last Year % Change"  
                    ,FORMAT(DIVIDE([OrderCount]-SUMX(SAMEPERIODLASTYEAR('Date'[Date]),[OrderCount]),SUMX(SAMEPERIODLASTYEAR('Date'[Date]),[OrderCount])), "Percent")  
                    ,IF(SELECTEDVALUE(TimeMeasures[TimeMeasureName], "Current")="Same Month Last Year Growth"  
                        ,[OrderCount]-SUMX(SAMEPERIODLASTYEAR('Date'[Date]),[OrderCount])  
                        ,Orders[OrderCount]))))))))
```

Recap

Feature	Multidimensional	Tabular
Many to Many Dimensions	More involved to setup, more complexity in the solution, but once it's solved, it works.	Less complex to configure, more involved to generate correct calculations

Recap

Feature	Multidimensional	Tabular
Many to Many Dimensions	More involved to setup, more complexity in the solution, but once it's solved, it works.	Less complex to configure, more involved to generate correct calculations
Row Level Security	Makes sense conceptually, but complexity arises in getting the correct allowed member set AND visual totals	Filter is overall less complex to model and easier to understand

Recap

Feature	Multidimensional	Tabular
Many to Many Dimensions	More involved to setup, more complexity in the solution, but once it's solved, it works.	Less complex to configure, more involved to generate correct calculations
Row Level Security	Makes sense conceptually, but complexity arises in getting the correct allowed member set AND visual totals	Filter is overall less complex to model and easier to understand
Scoping Metrics	The "Dark Art" of scoping has it's benefits and pitfalls, but it reliable when you understand the rules	No true scoping, but changing the filter context on the fly makes this behave similarly and is very useful

Additional Reading

- JAVIER!
 - <https://javierguillen.wordpress.com/2012/05/02/scoping-at-different-granularities-in-dax-part-i/>
 - <https://javierguillen.wordpress.com/2012/05/04/scoping-at-different-granularities-part-ii/>
- <https://www.daxpatterns.com/>
- Many to Many Revolution
 - <https://www.sqlbi.com/topics/many-to-many/>
- Securing the Tabular Model
 - <https://msdn.microsoft.com/en-us/library/jj127437.aspx>
- White Papers
 - SQL 2012 - [https://msdn.microsoft.com/en-us/library/403491\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/403491(v=sql.110).aspx)
 - SQL 2014 - [https://msdn.microsoft.com/en-us/library/dn609810\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/dn609810(v=sql.120).aspx)