



sqlbits

SPEAKEASY

Enterprise BI with Power BI and SSAS

Part 1 & 2

Christian Wade, Senior Program Manager at Microsoft



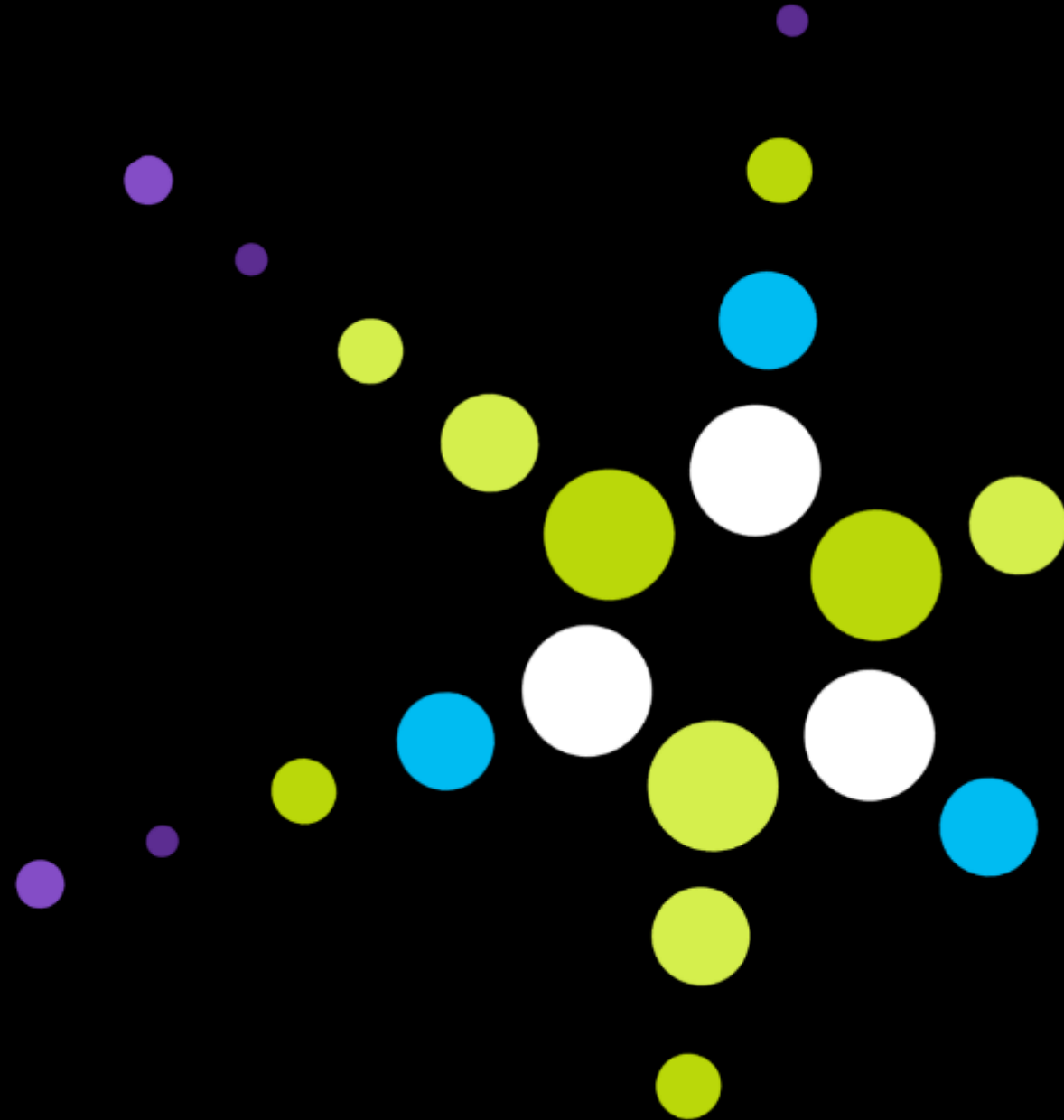
@_christianWade

We'd love your feedback

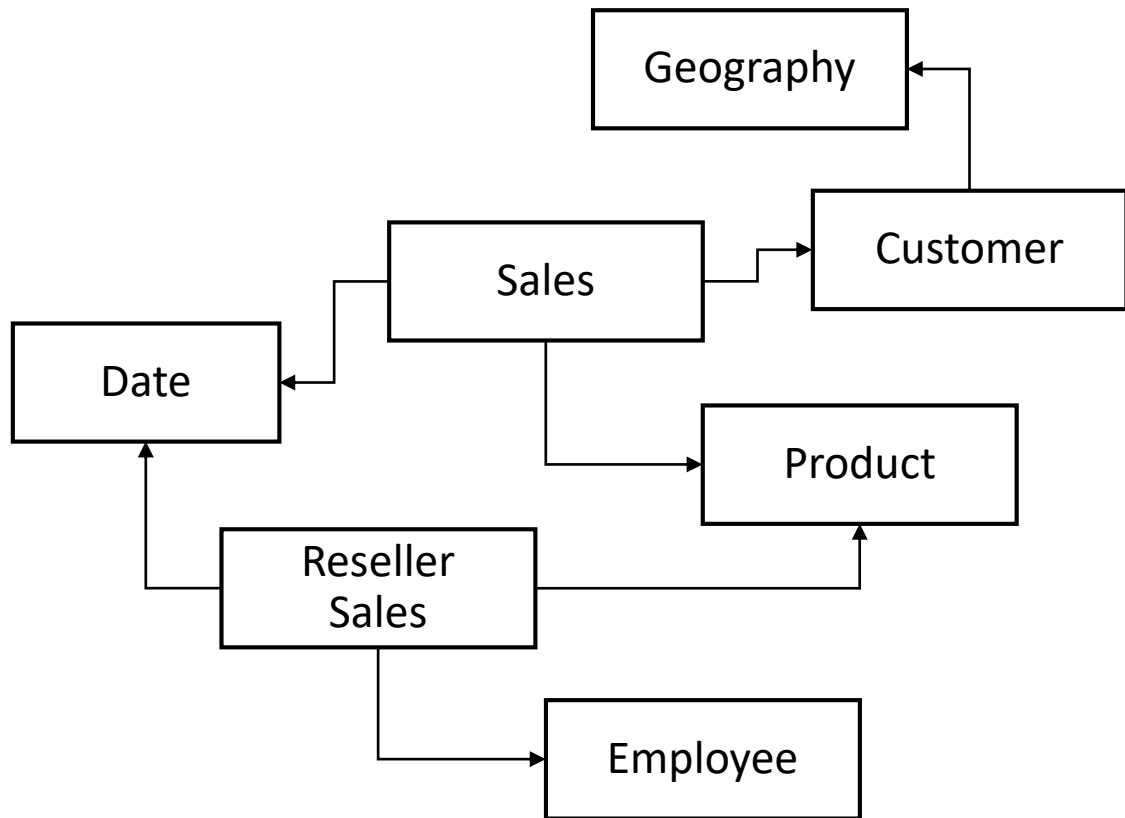
[Aka.ms/SQLBits19](https://aka.ms/SQLBits19)

Aggregations

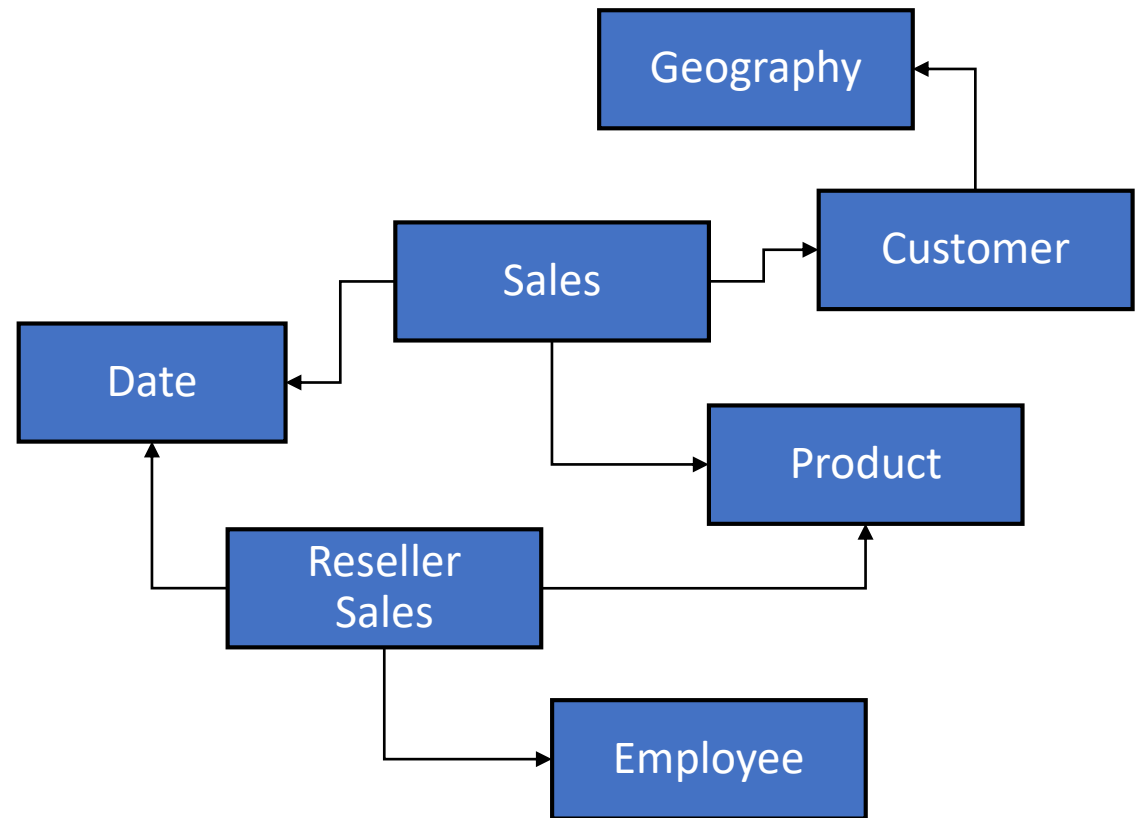
Demo: <https://aka.ms/TrillionRowDemo>



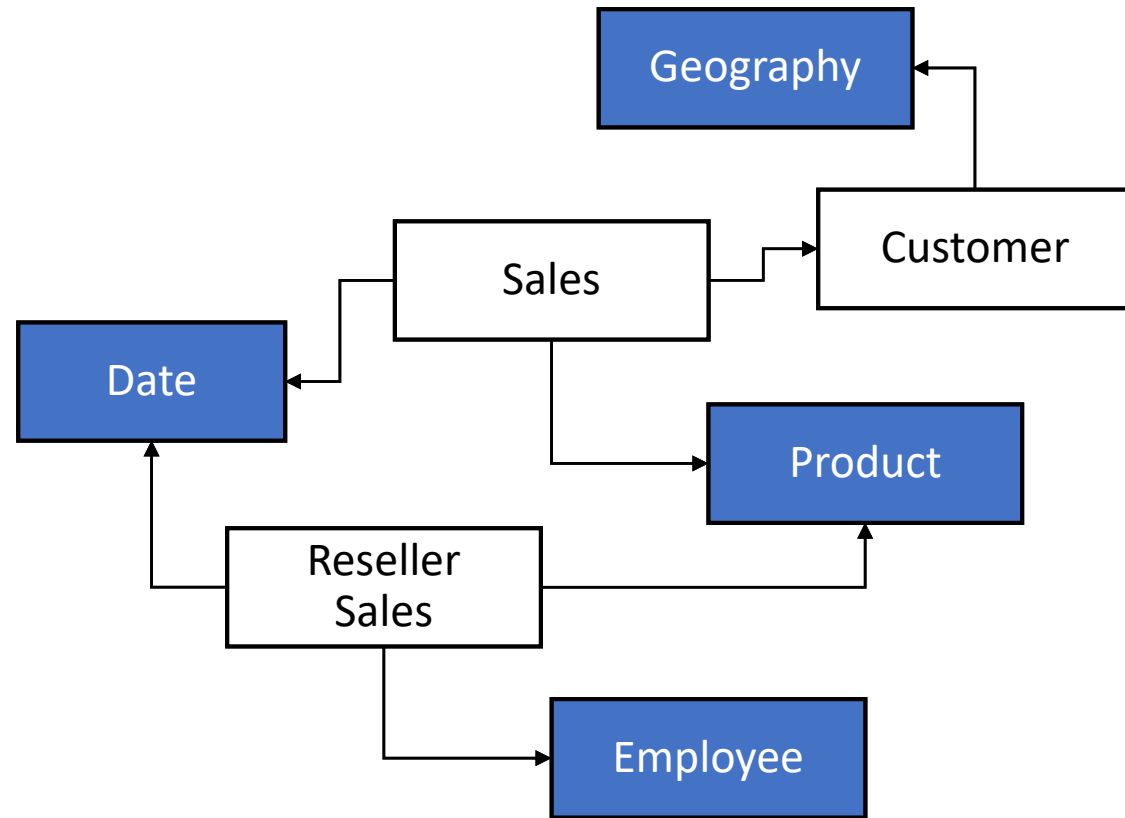
DirectQuery



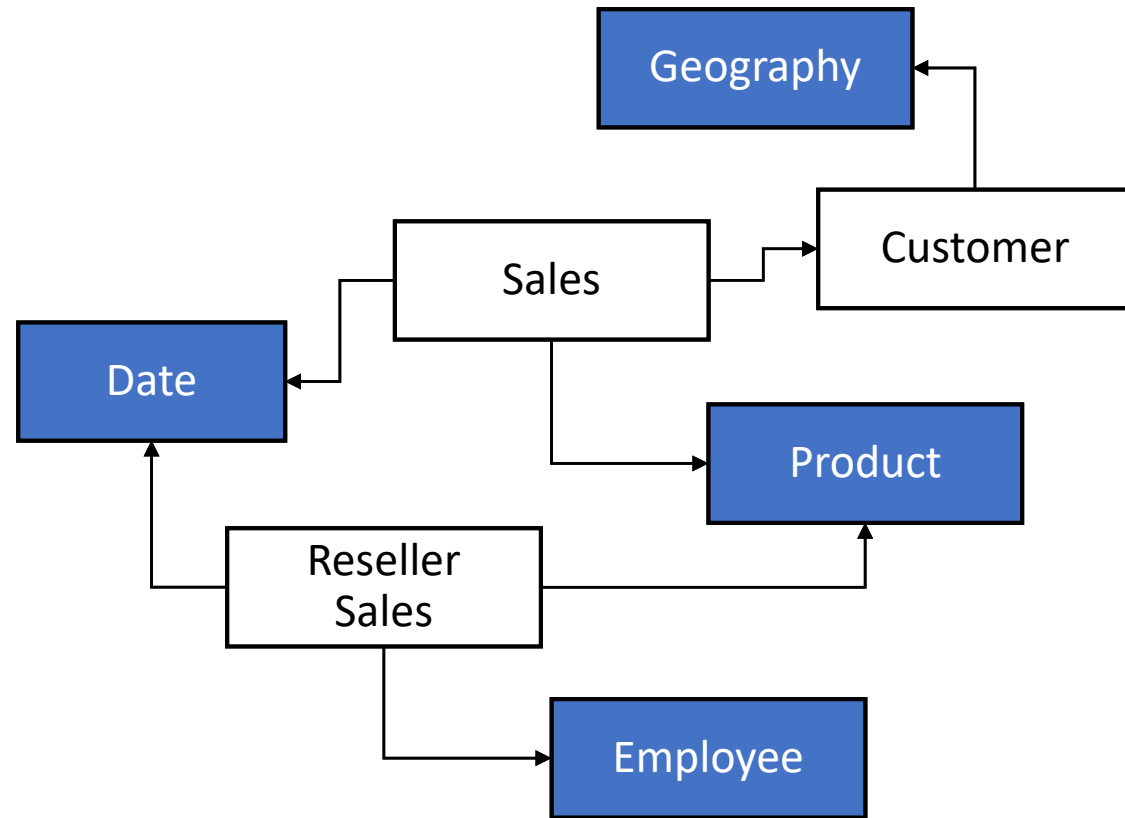
Import



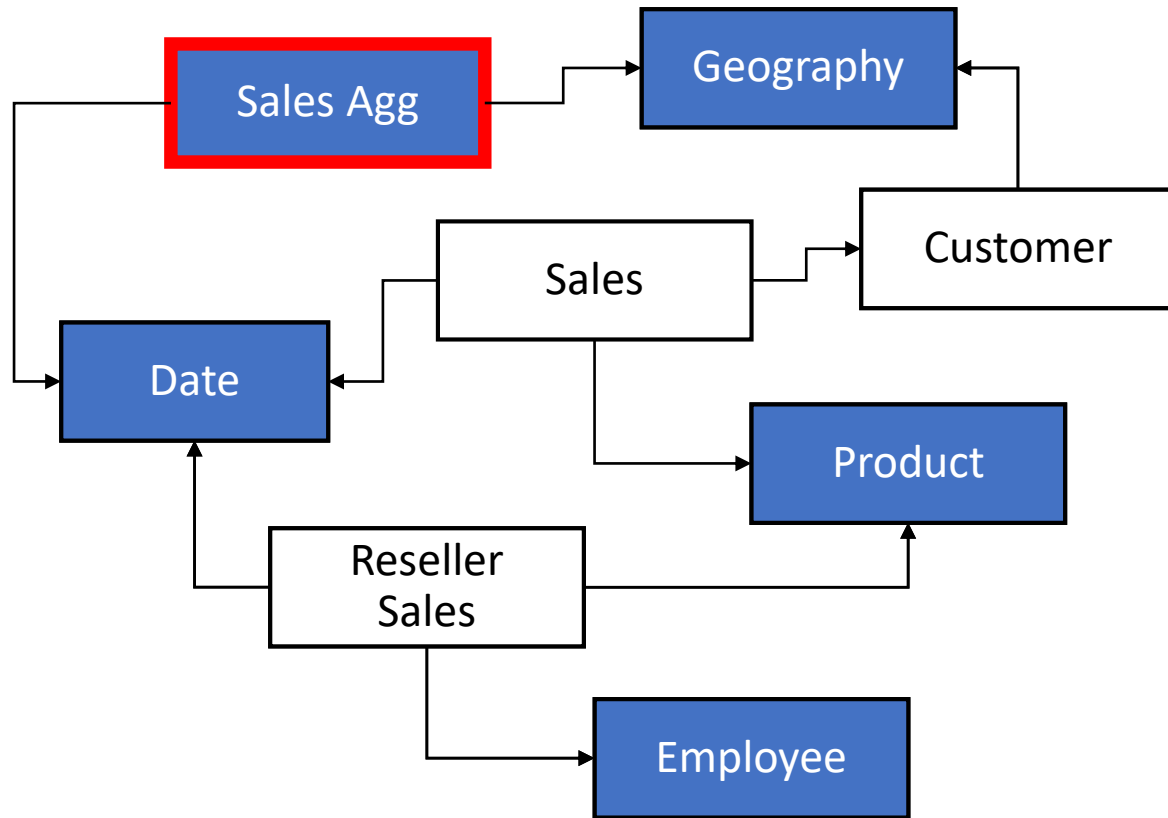
DirectQuery & Import



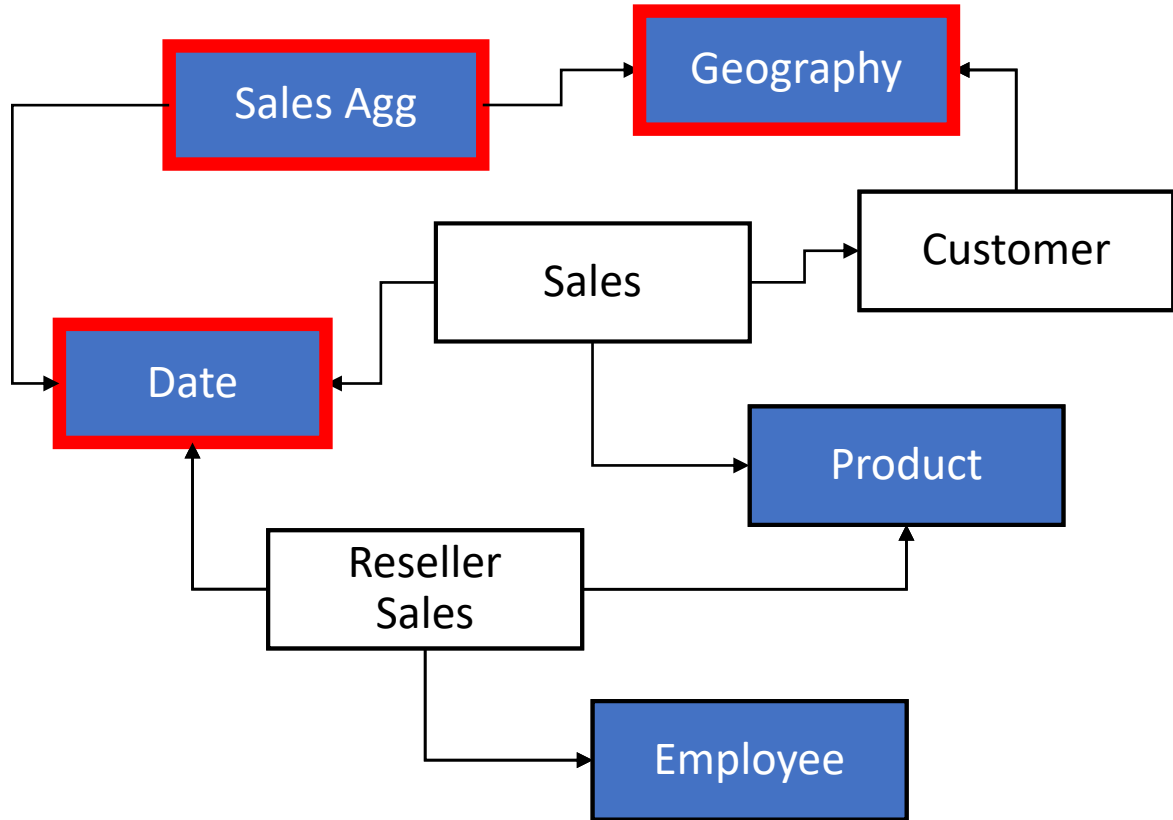
DirectQuery & Import



Aggregations



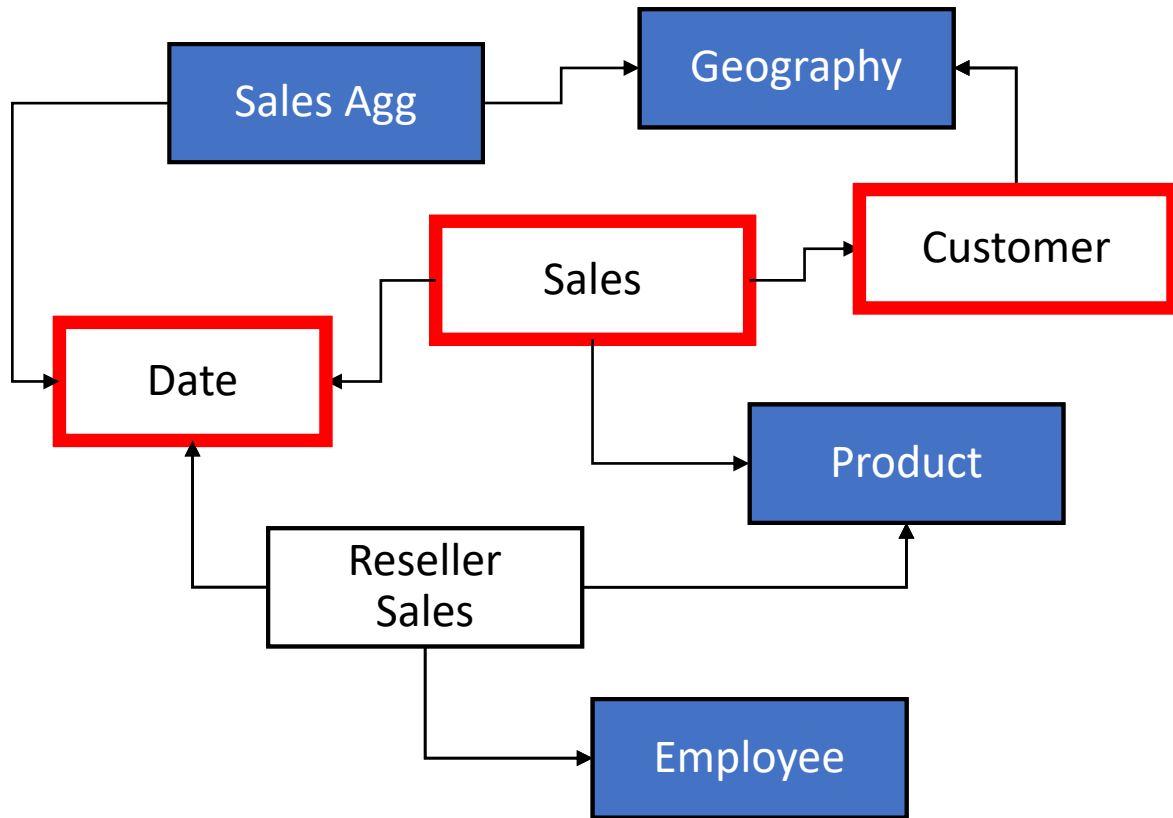
Aggregations



```
SummarizeColumns(  
  Date[Year],  
  Geography[City],  
  "Sales", Sum(Sales[Amount])  
)
```

Hits in-memory cache

Aggregations



```
SummarizeColumns(  
  Date[Year],  
  Customer[Name],  
  "Sales", Sum(Sales[Amount])  
)
```

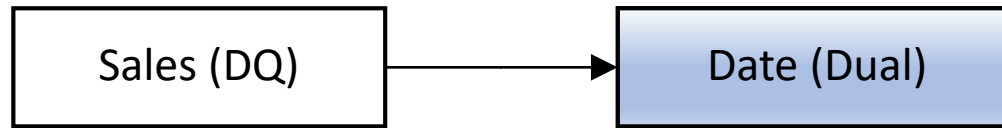
DirectQuery

Aggregations: dimensional models

Demo



Strong vs. weak relationships



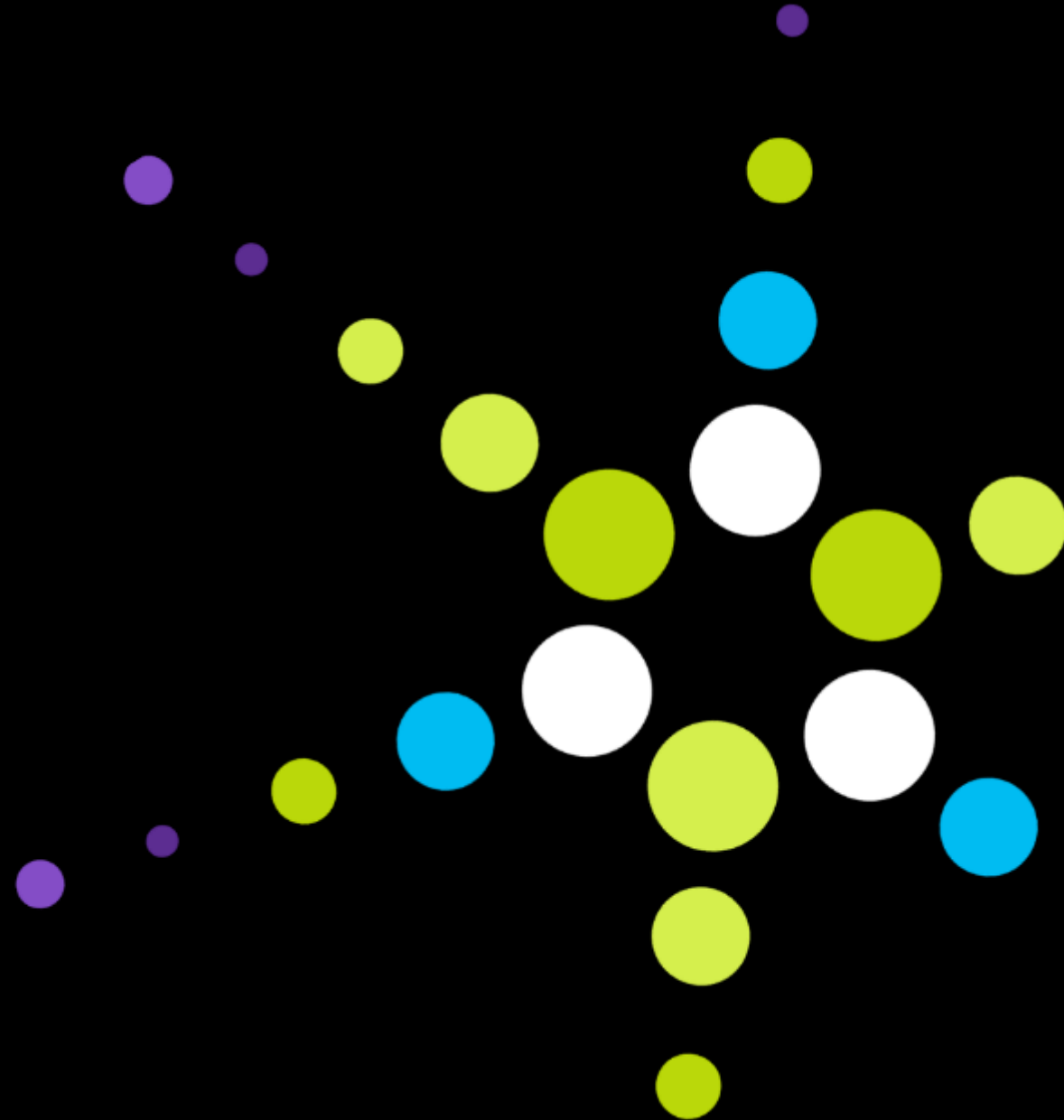
- Able to push the “join” to the source
- Considered for aggregation hits

Strong relationship rules (both sides from a single source)

“Many side”	“One side”
Dual	Dual
Import	Dual or Import
DQ	Dual or DQ

Aggregations: big-data models

Demo



Enterprise BI



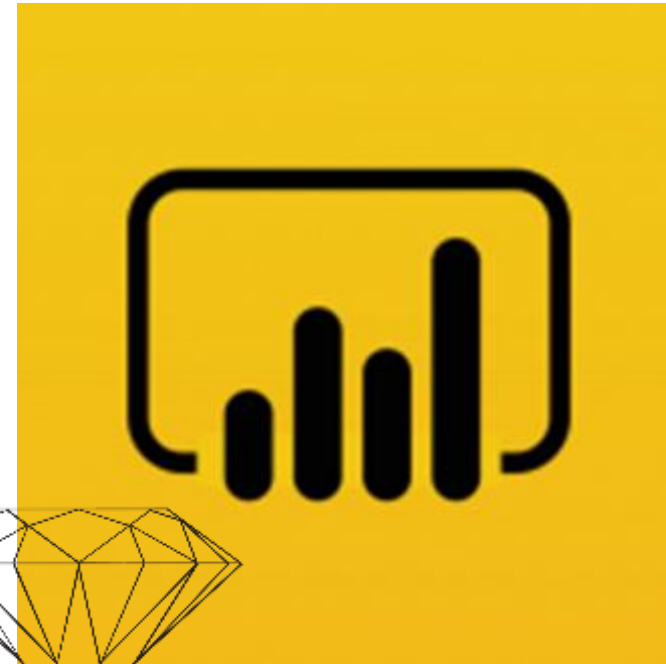
Azure
Analysis Services

All BI users



Power BI
Premium

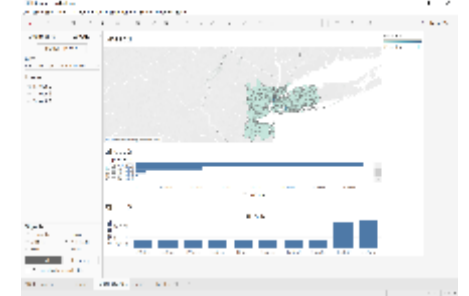
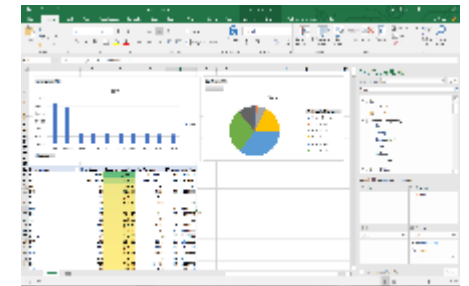
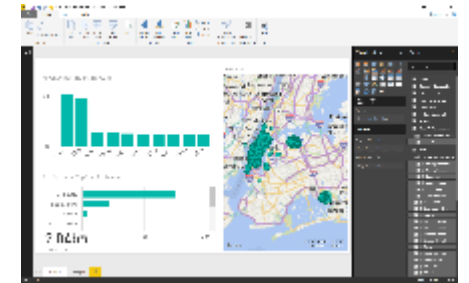
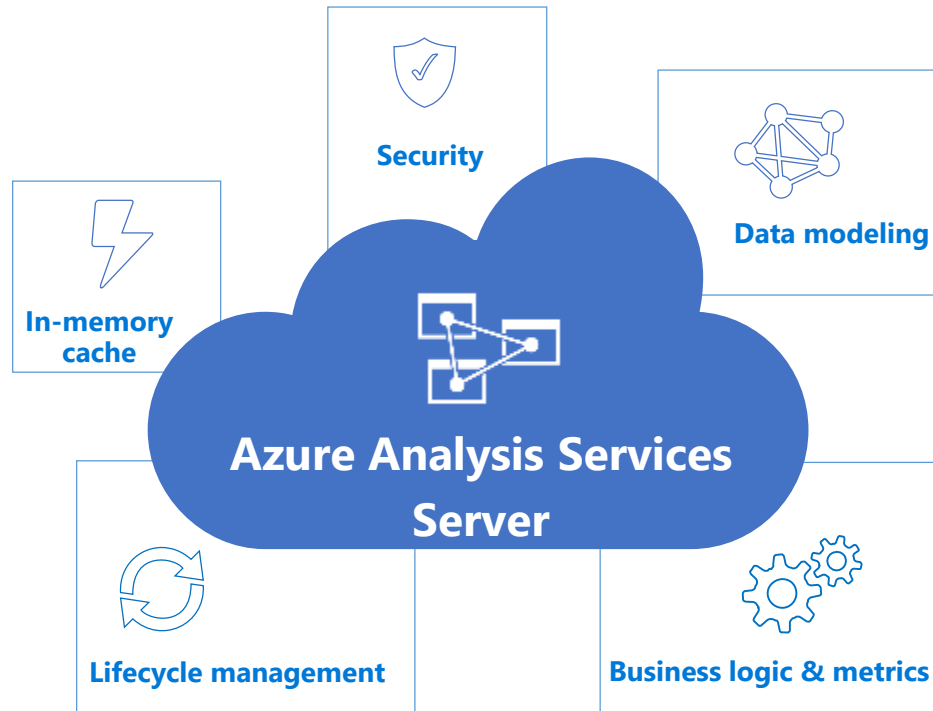
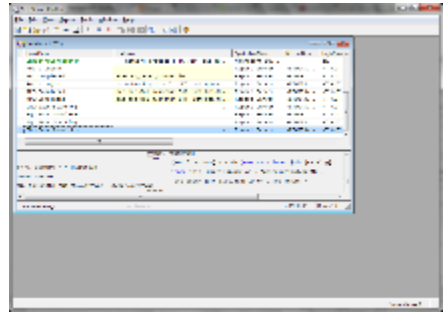
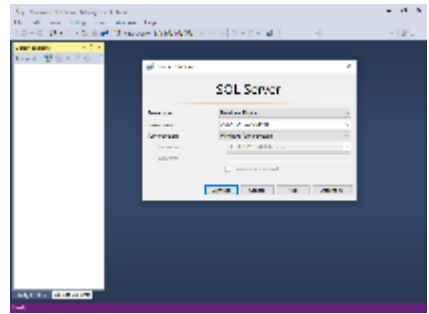
Self-service BI users



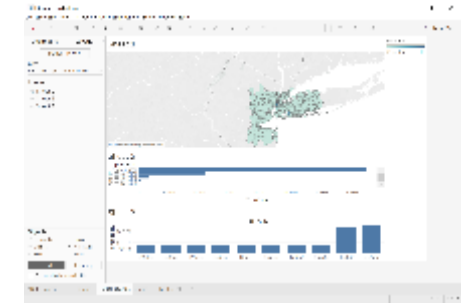
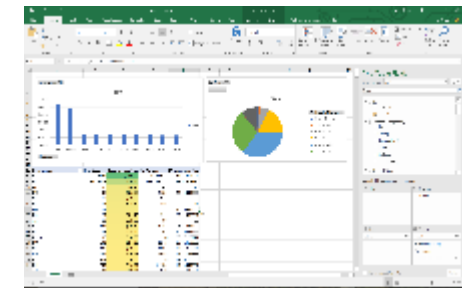
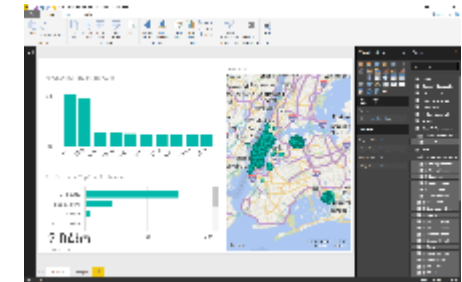
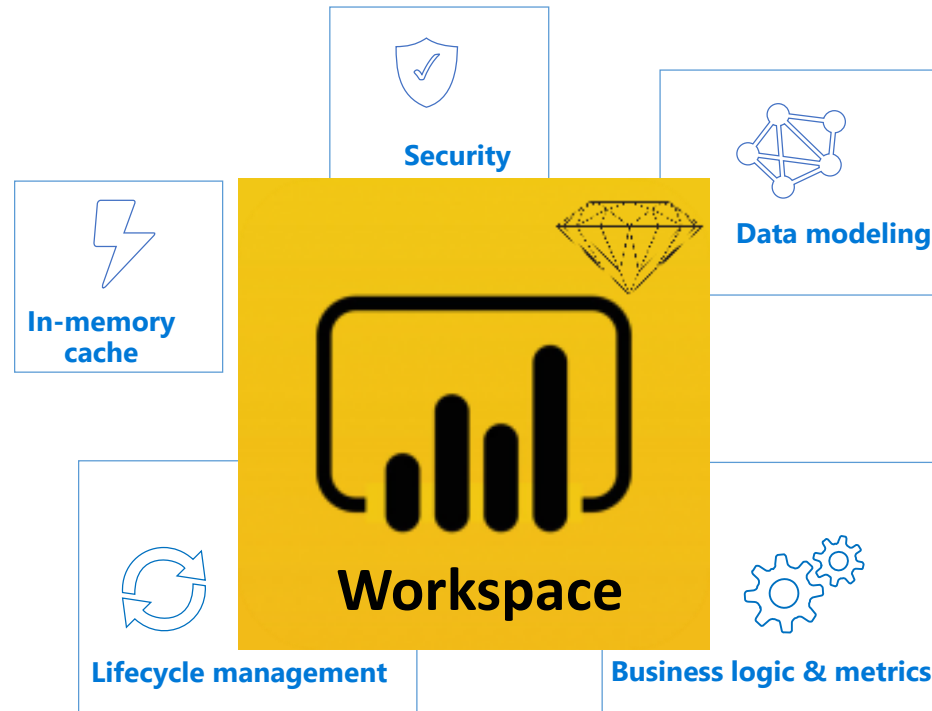
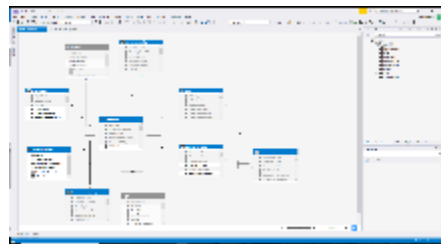
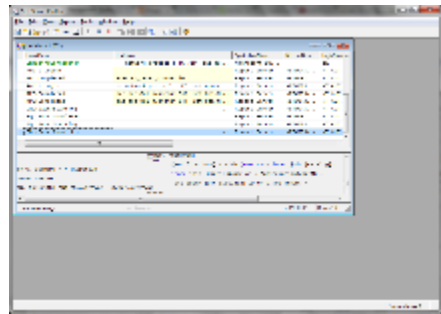
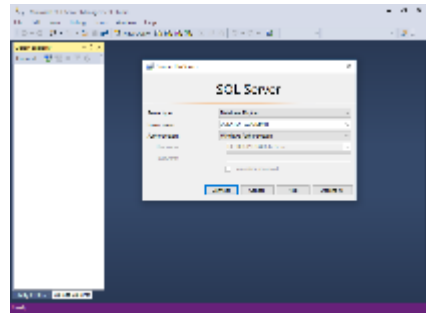
Power BI

XMLA Endpoints

Connectivity: Analysis Services

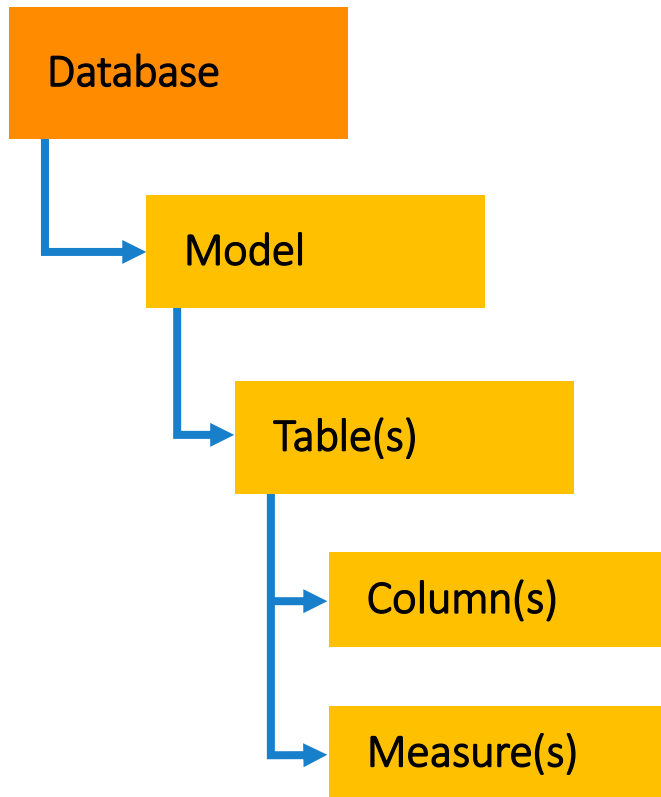


Connectivity: Analysis Services



Tabular Object Model (TOM)

Community tools, partition management, automation, ...



```
public void RefreshTable(...)
{
    var server = new Server();
    server.Connect(cnnString);

    // Connect to the server
    Database db = server.Databases[dbName];

    // Connect to the database
    Model = db.Model;

    // Reprocess the table
    model.Tables[tableName].RequestRefresh(RefreshType.Full);
    model.SaveChanges(); // Commit the changes
}
```

Tabular Model Scripting Language (TMSL)

- Admin functions: Create, Alter, Refresh
- SSAS PowerShell cmdlet Invoke-AsCmd accepts TMSL commands

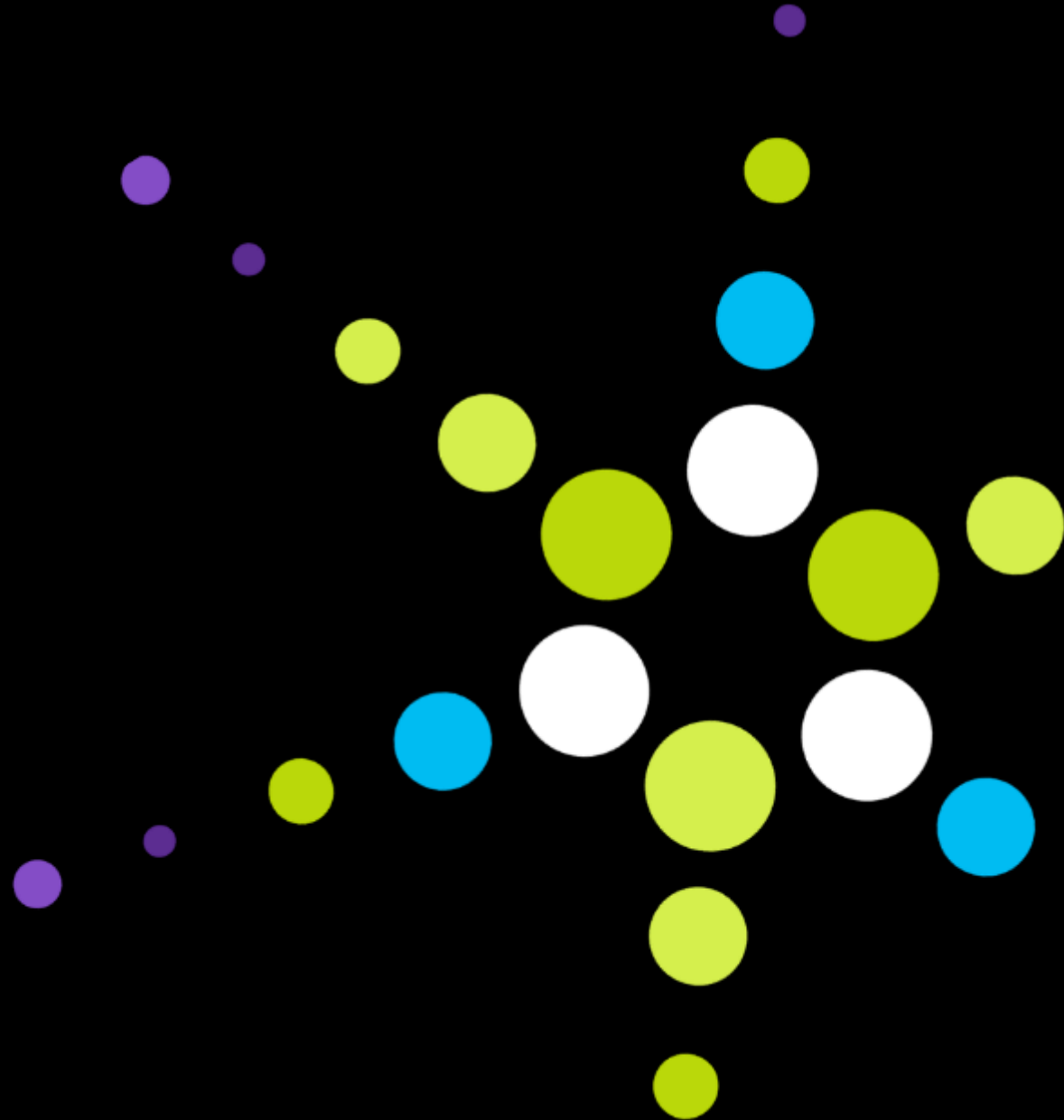
```
{
  "refresh": {
    "type": "full",
    "objects": [
      {
        "database": "Sales Analysis",
        "table": "Reseller Sales"
      }
    ]
  }
}
```

```
{
  "createOrReplace": {
    "object": {
      "database": "AdventureWorks"
    },
    "database": {
      "name": "AdventureWorks",
      ...
    }
  }
}
```

Incremental refresh

XMLA endpoints & incremental refresh

Demo

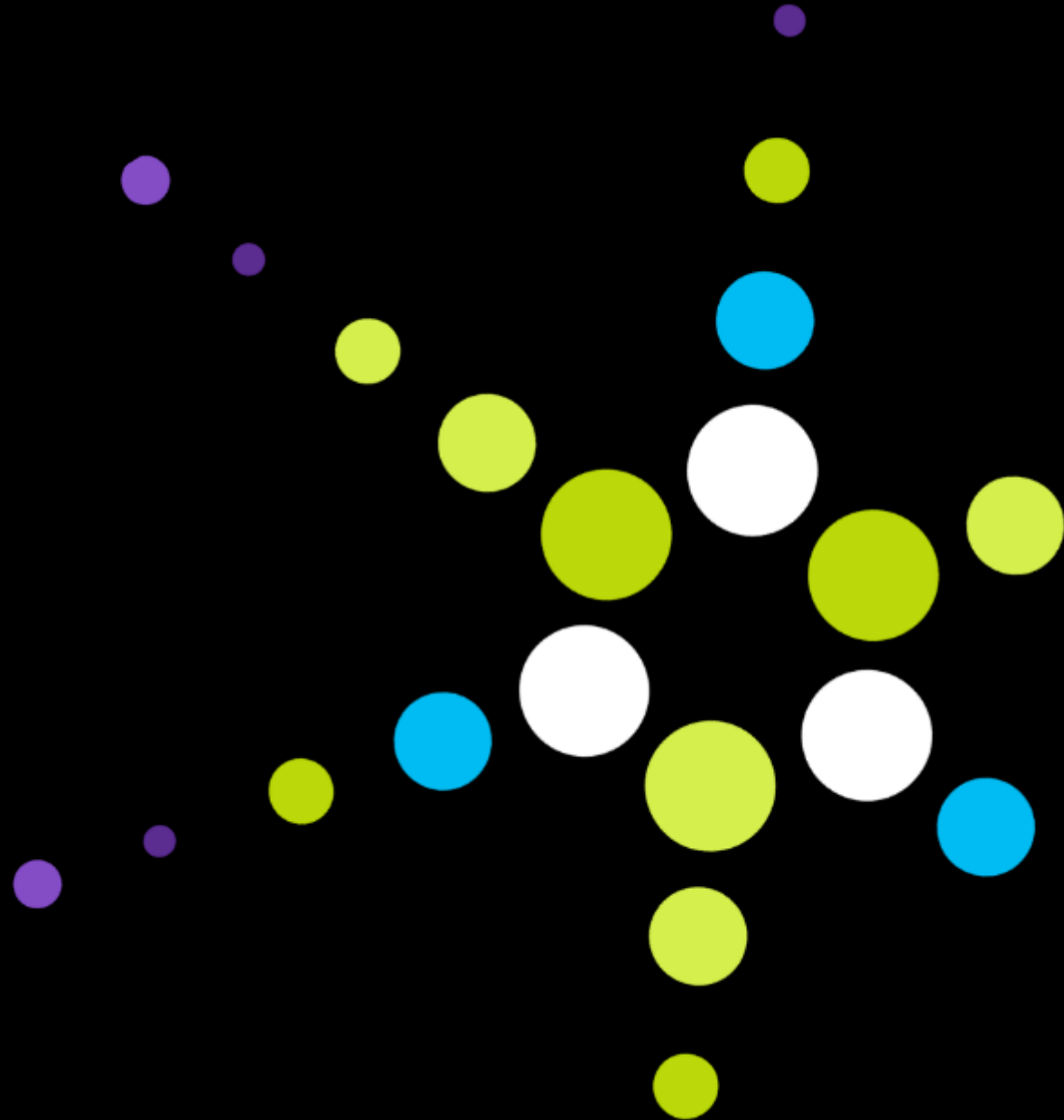


Incremental refresh

- Enable large models in Power BI
 - Faster refresh
 - More reliable
 - Lower CPU and Memory usage
- Define policy in Power BI Desktop; apply it in the service
- Policy doesn't affect data in Power BI Desktop

Modeling view

Demo



Modeling enhancements in Analysis Services

Multidimensional vs. tabular

- Recent tabular modeling recent features
 - Detail rows
 - OLS
 - Ragged hierarchies
 - Aggregations/composite models

Sales	Orders
Sales YTD	Orders YTD
Sales MTD	Orders MTD
Sales QTD	Orders QTD
Sales 3 Month Avg	Orders 3 Month Avg
Sales 12 Month Avg	Orders 12 Month Avg
Sales Prev Year	Orders Prev Year
Sales YoY %	Orders YoY %
...	...

Calculation groups

- Reuse DAX calculations to reduce complexity
- Will ship in
 - SSAS 2019 **TODAY (CTP 2.3)!**
 - Azure AS
 - Power BI Premium (XMLA endpoint enablement initially)
- Require new 1470 compat level

Calculation groups

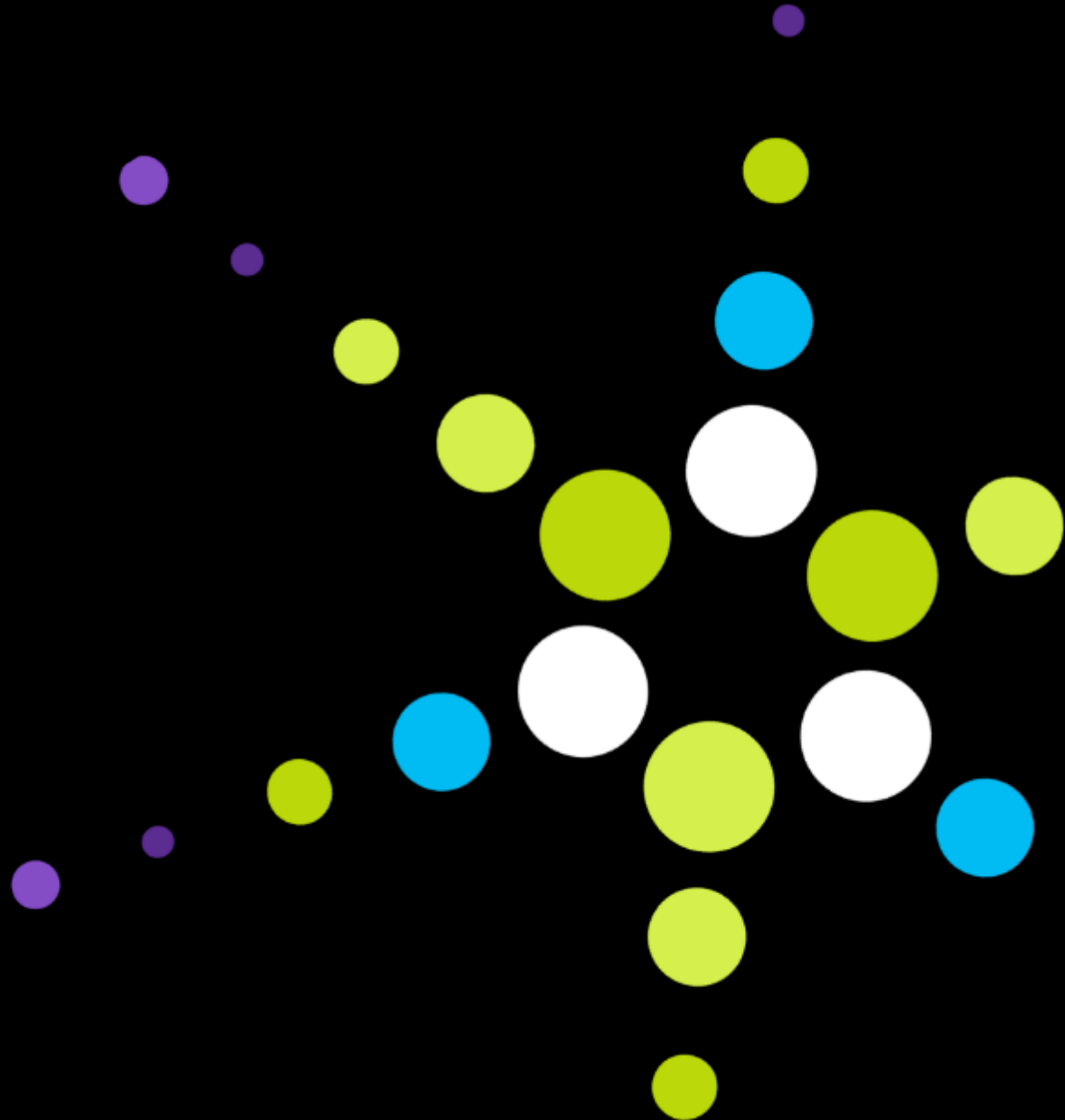
Calculation Item	Expression
"Current"	<code>SELECTEDMEASURE()</code>
"MTD"	<code>CALCULATE(SELECTEDMEASURE(), DATESMTD(DimDate[Date]))</code>
"QTD"	<code>CALCULATE(SELECTEDMEASURE(), DATESQTD(DimDate[Date]))</code>
"YTD"	<code>CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))</code>
"PY"	<code>CALCULATE(SELECTEDMEASURE(), SAMEPERIODLASTYEAR(DimDate[Date]))</code>
"PY MTD"	<code>CALCULATE(SELECTEDMEASURE(), SAMEPERIODLASTYEAR(DimDate[Date]), 'Time Intelligence'[Time Calculation] = "MTD")</code>
"PY QTD"	<code>CALCULATE(SELECTEDMEASURE(), SAMEPERIODLASTYEAR(DimDate[Date]), 'Time Intelligence'[Time Calculation] = "QTD")</code>
"PY YTD"	<code>CALCULATE(SELECTEDMEASURE(), SAMEPERIODLASTYEAR(DimDate[Date]), 'Time Intelligence'[Time Calculation] = "YTD")</code>
"YOY"	<code>SELECTEDMEASURE() - CALCULATE(SELECTEDMEASURE(), 'Time Intelligence'[Time Calculation] = "PY")</code>
"YOY%"	<code>DIVIDE(CALCULATE(SELECTEDMEASURE(), 'Time Intelligence'[Time Calculation]="YOY"), CALCULATE(SELECTEDMEASURE(), 'Time Intelligence'[Time Calculation]="PY"),)</code>

Table	Time Intelligence
Column	Time Calculation
Precedence	20

Calculation groups

Demo:

- Power BI consumption experience
- Tabular Editor
- Queries



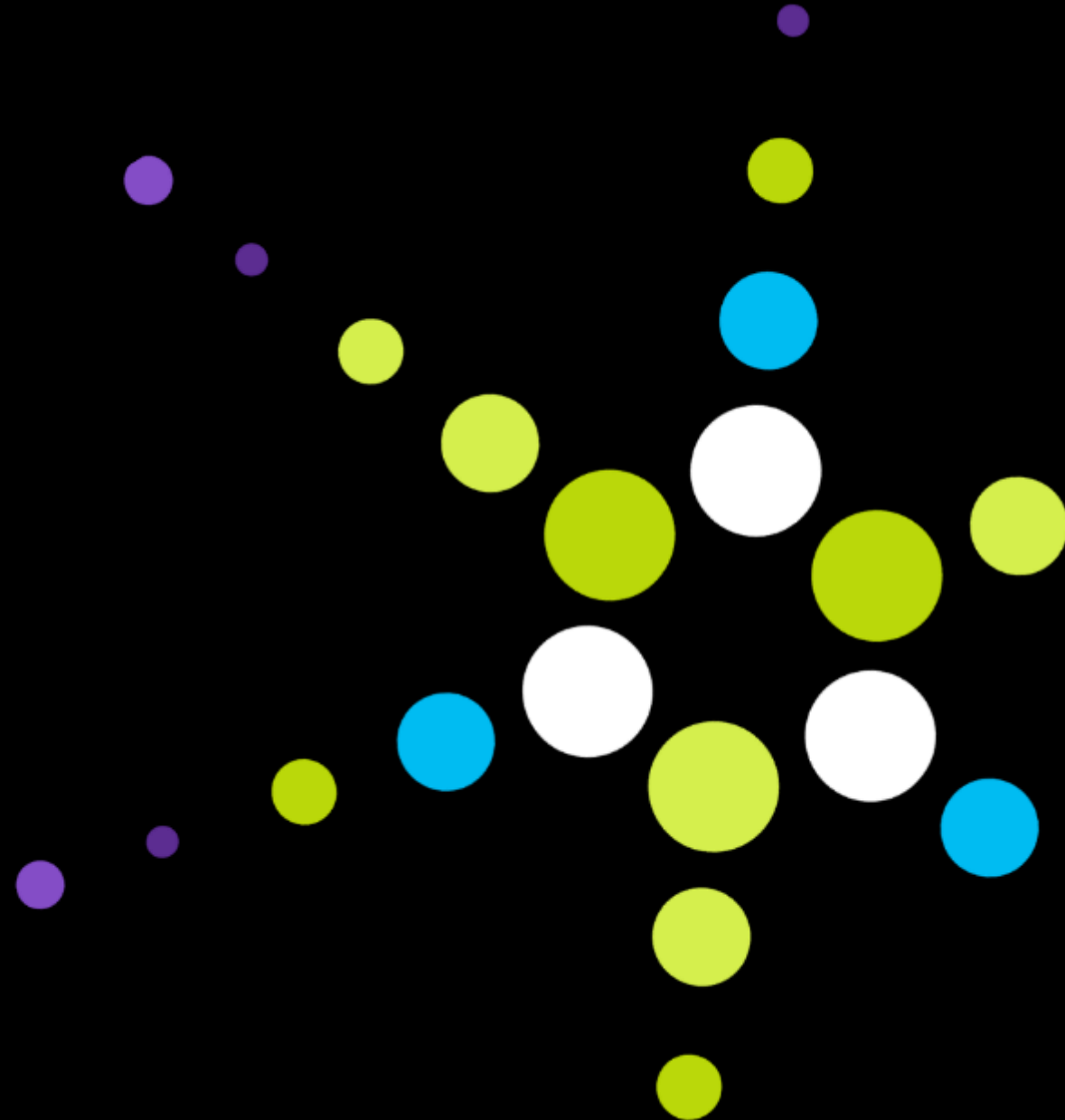
Multiple calculation groups w/ precedence

Table	Averages
Column	Average Calculation
Precedence	10

Calculation Item	Expression
"No Average"	SELECTEDMEASURE()
"Daily Average"	DIVIDE(SELECTEDMEASURE(), COUNTROWS(DimDate))

Calculation groups: precedence

Demo



Precedence

"YTD"	<pre>CALCULATE(SELECTEDMEASURE(), DATESYTD(DimDate[Date]))</pre>	Precedence=20
"Daily Average"	<pre>DIVIDE(SELECTEDMEASURE(), COUNTROWS(DimDate))</pre>	Precedence=10
"YTD" "Daily Average"	<pre>CALCULATE(DIVIDE(SELECTEDMEASURE(), COUNTROWS(DimDate)), DATESYTD(DimDate[Date]))</pre>	

New DAX functions

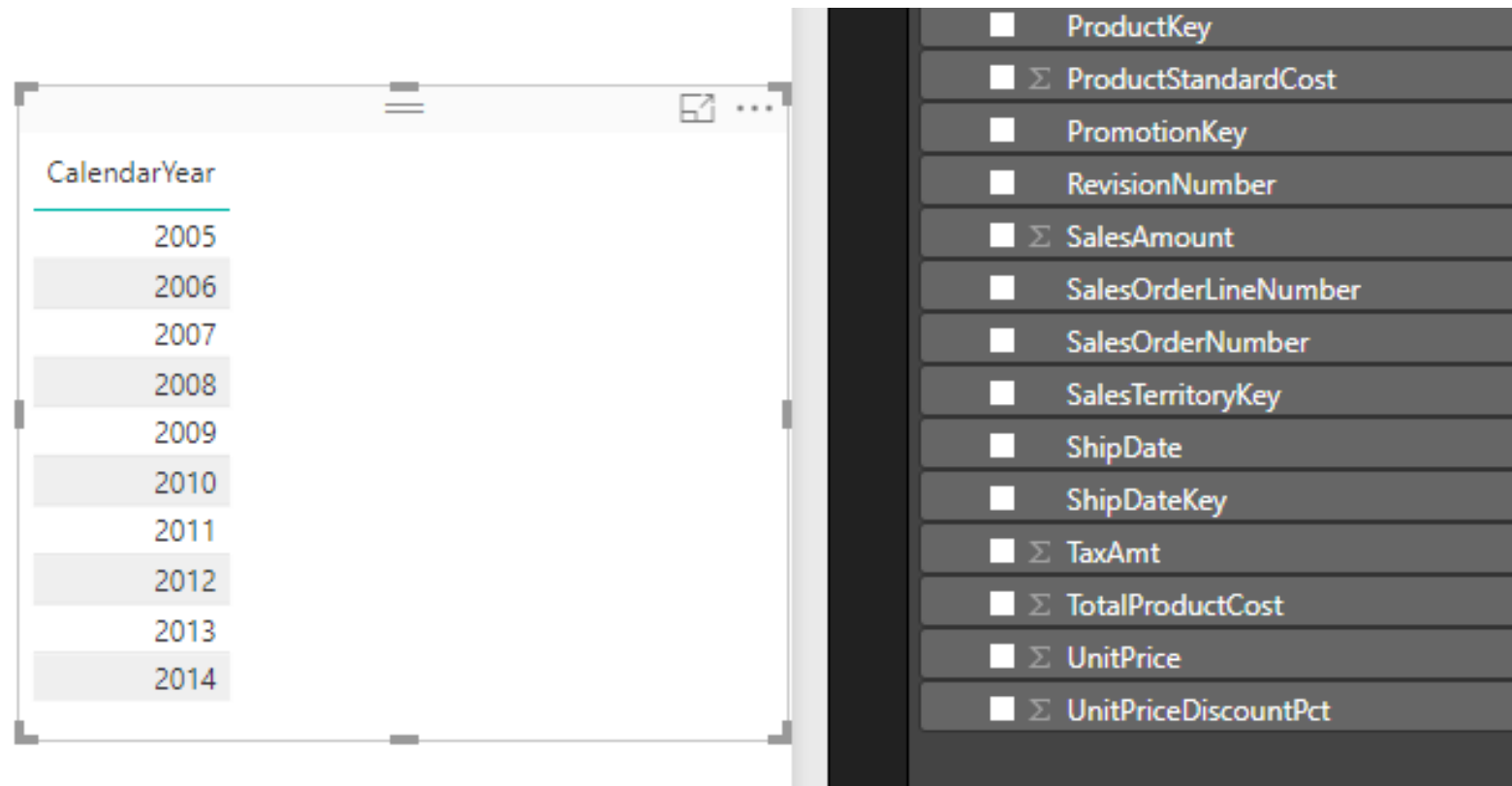
Function name	Description
<code>SELECTEDMEASURE()</code>	Returns a reference to the measure currently in context.
<code>SELECTEDMEASURENAME()</code>	Returns a string containing the name of the measure currently in context.
<code>ISSELECTEDMEASURE(M1, M2, ...)</code>	Returns a Boolean indicating whether the measure currently in context is one of those specified as an argument.

```
IF (
    ISSELECTEDMEASURE ( [Expense Ratio 1], [Expense Ratio 2] ),
    SELECTEDMEASURE (),
    DIVIDE ( SELECTEDMEASURE (), COUNTROWS ( DimDate ) )
)
```


Not working yet

- MDX support – **coming soon**
- Dynamic format strings – **coming soon**
- OLS/RLS
- Implicit measures

DiscourageImplicitM

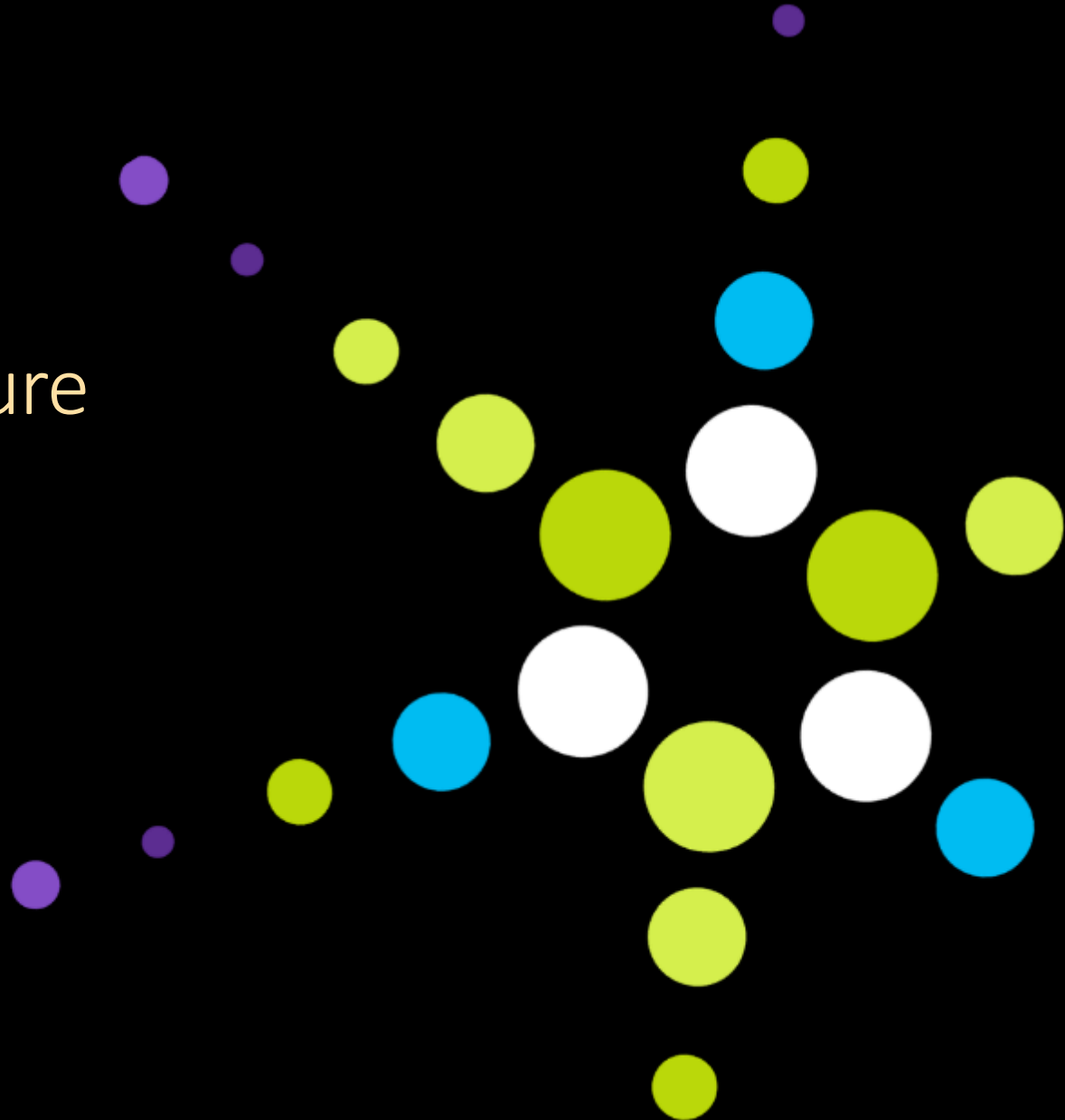


The screenshot shows a BI tool interface. On the left, a 'CalendarYear' filter is applied, showing a list of years from 2005 to 2014. On the right, a list of fields is displayed, including keys, measures, and other attributes.

Field Name	Measure Type
ProductKey	Key
ProductStandardCost	Measure (Σ)
PromotionKey	Key
RevisionNumber	Key
SalesAmount	Measure (Σ)
SalesOrderLineNumber	Key
SalesOrderNumber	Key
SalesTerritoryKey	Key
ShipDate	Key
ShipDateKey	Key
TaxAmt	Measure (Σ)
TotalProductCost	Measure (Σ)
UnitPrice	Measure (Σ)
UnitPriceDiscountPct	Measure (Σ)

Calculation groups: implicit measure syntax

Demo



Recursion safeguards not required

```
(  
    [Date].[Calendar Date Calculations].[Year to Date],  
    [Date].[Calendar Year].[Calendar Year].Members,  
    [Date].[Date].Members  
) =
```

```
Aggregate(  
    { [Date].[Calendar Date Calculations].[Current Date] } *  
    PeriodsToDate(  
        [Date].[Calendar].[Calendar Year],  
        [Date].[Calendar].CurrentMember  
    )  
);
```

Calculation items cannot be created on regular columns

```
WITH  
MEMBER [Geography].[State-Province].[Pacific Northwest] AS  
SUM({[Geography].[State-Province].&[WA]&[US],  
      [Geography].[State-Province].&[OR]&[US],  
      [Geography].[State-Province].&[ID]&[US]})  
)  
SELECT  
{[Measures].[Internet Sales Amount]}  
ON COLUMNS,  
{[Geography].[State-Province].[Pacific Northwest]}  
ON ROWS  
FROM [Adventure Works]
```

SQL Server Analysis Services 2019

- **Modeling & Analytics**
 - Calculation groups (aka calc members) **TODAY (CTP 2.3)!**
 - Many-to-many relationships
 - SuperDAXMD (Multidimensional)
- **Resource governance**
 - New memory management settings (already shipped in Azure AS):
 - QueryMemoryLimit, RowsetSerializationLimit
- **Data Connectivity (to be confirmed)**
 - Will prioritize from new Power Query enterprise data sources
- **New DAX functions including:**
 - IsInScope(), SelectedValue() ...

Session resources

- Announcement blog post at: <https://aka.ms/ASTeamBlog>
 - Blog to be migrated soon likely to Power BI blog
- SSAS 2019 CTP 2.3 can be downloaded : <https://www.microsoft.com/en-us/sql-server/sql-server-2019>
- Aggregations documentation: <https://aka.ms/Aggregations>
- Incremental refresh demo: <https://aka.ms/IncrementalRefreshDemo>
- Incremental refresh documentation: <https://aka.ms/PBIIncrementalRefresh>
- Trillion row demo: <https://aka.ms/TrillionRowDemo>
- Microsoft Ready content can be found at <https://digital.microsoftready.com/>



FRIDAY NIGHT, SQLBITS

PROHIBITION PARTY



FEEDBACK FORMS

PLEASE FILL OUT AND PASS TO YOUR ROOM
HELPER BEFORE YOU LEAVE THE SESSION



sqlbits

SPEAKEASY