

Microsoft



Power BI

Sales Dashboard

Ask a question about the data on this dashboard.

How to ask

Add widget

Share

...

?

Help

Agenda

- Enterprise BI with Power BI
- Power BI and Analysis Services
- Incremental refresh
- Aggregations
- Reusing measures with Calculation groups
- Query interleaving ..



Enterprise BI Vision

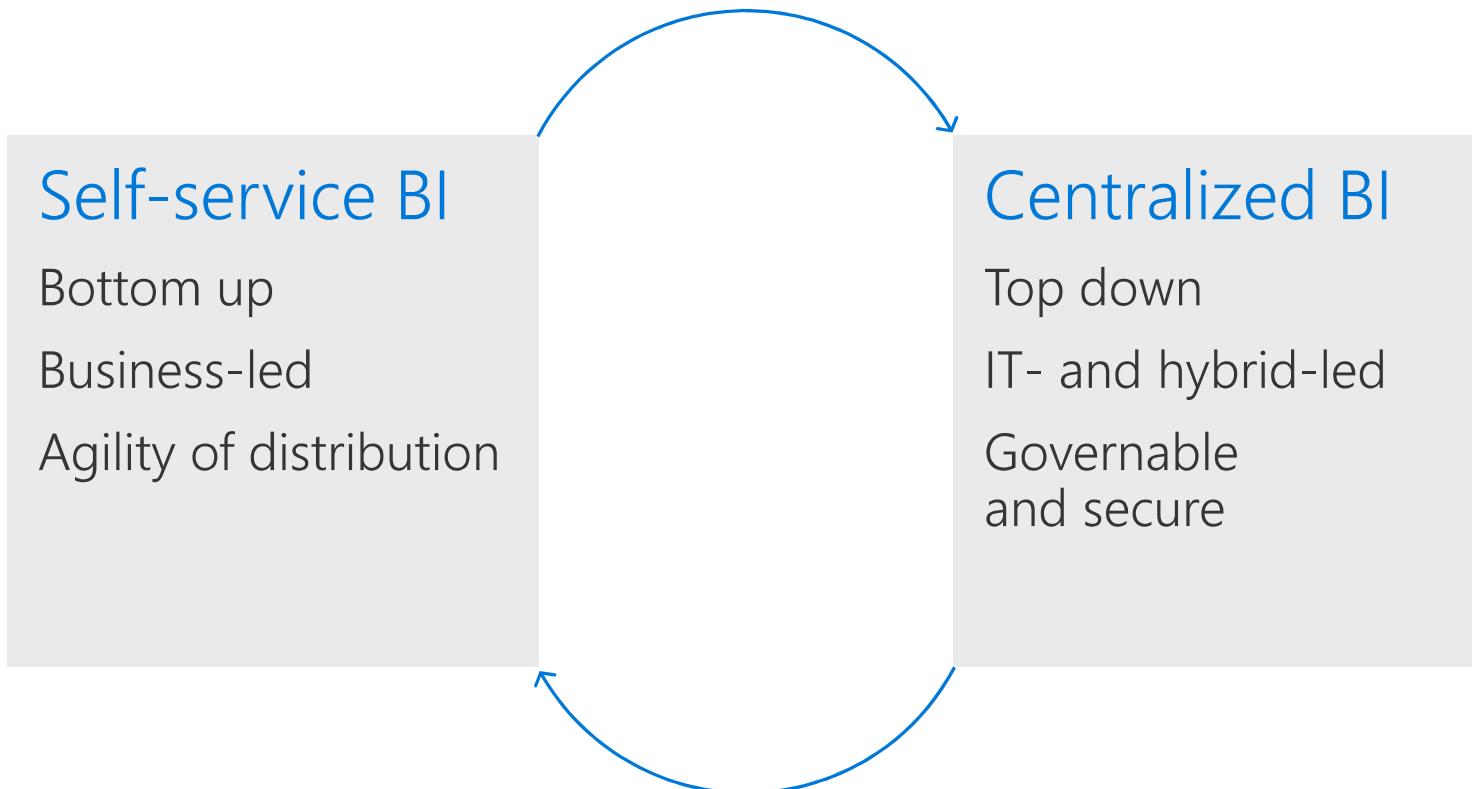


Our Vision

DRIVING A DATA CULTURE

- Pervasive data literacy
- Facts replace opinions
- Empower every business user and organization on the planet to achieve more with their data

Two adoption models within enterprises

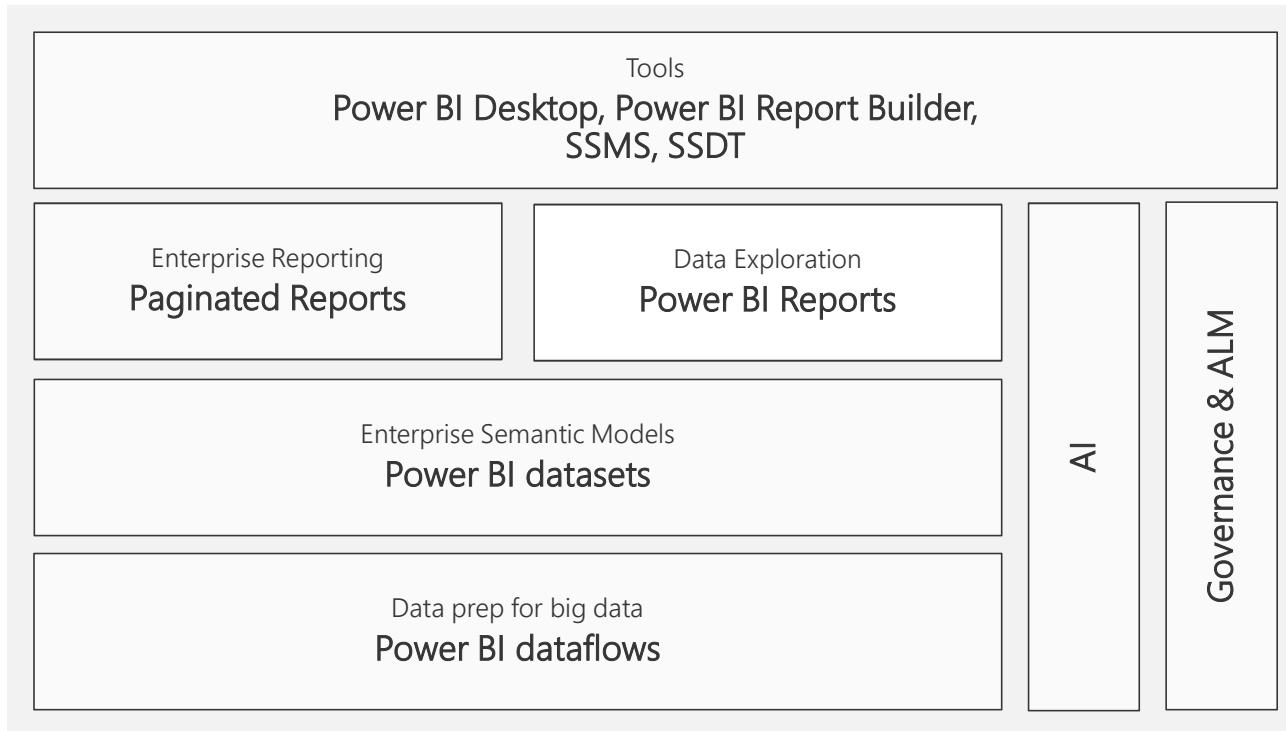


Power BI 2015

Data Exploration

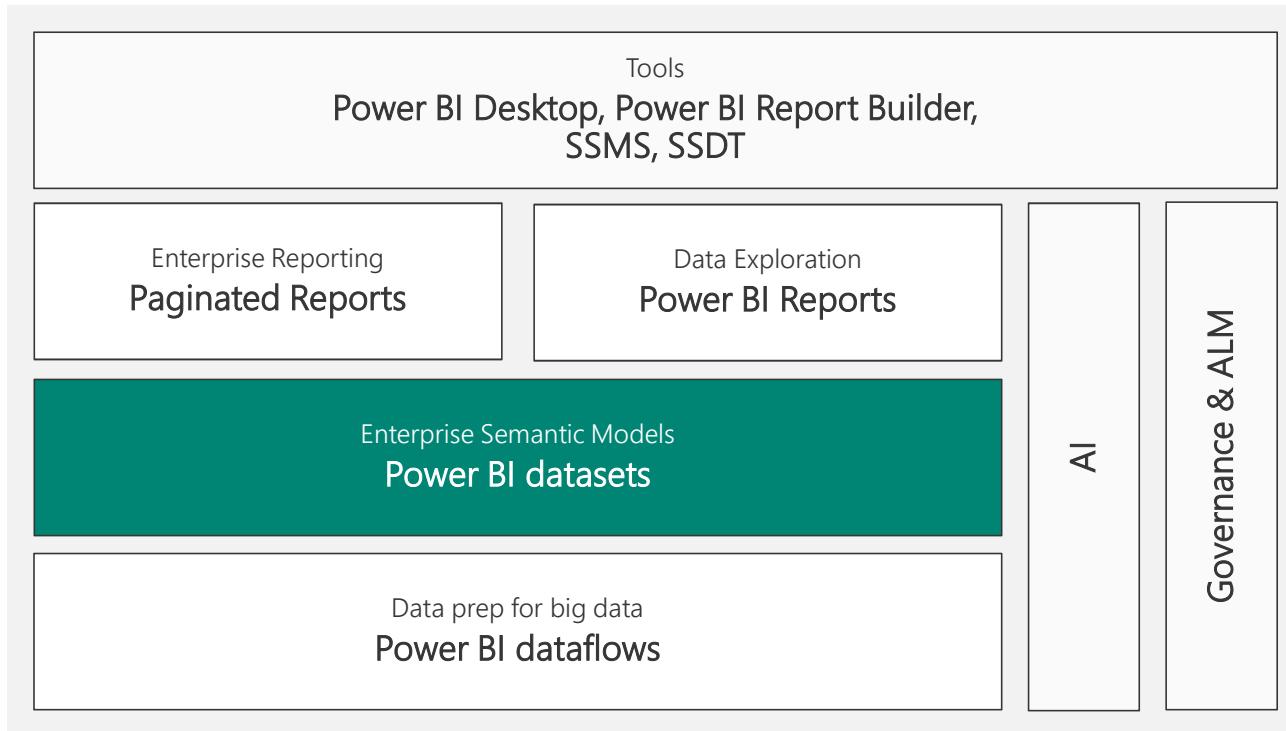
Power BI Reports

Power BI 2019



The Unified Enterprise BI Platform

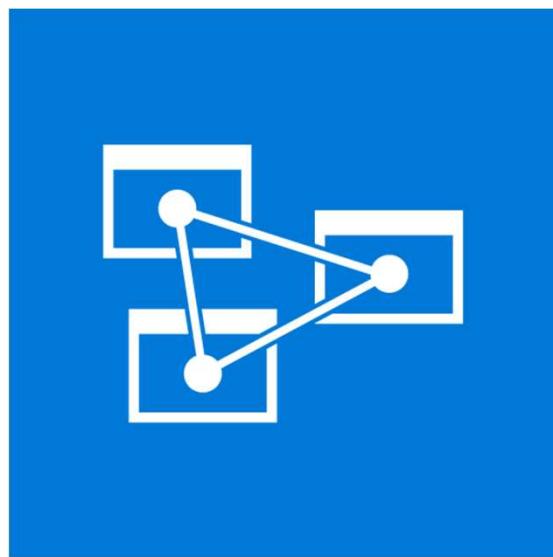
Power BI 2019



The Unified Enterprise BI Platform

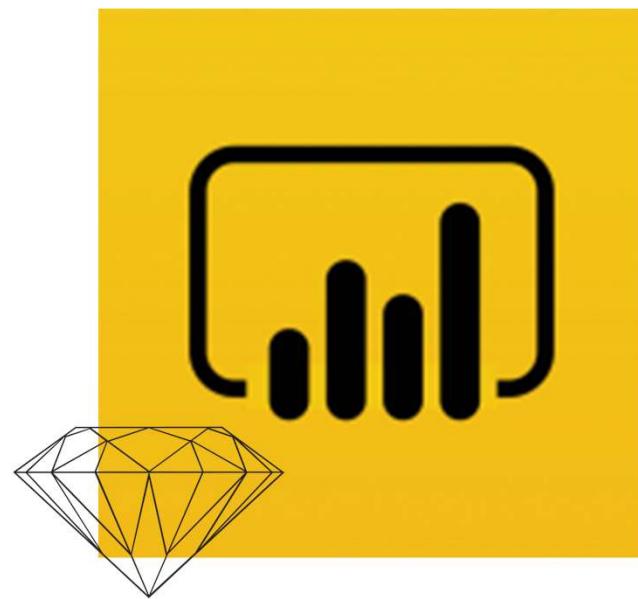
Enterprise semantic models

Enterprise BI



Azure
Analysis Services

Self-service BI
users

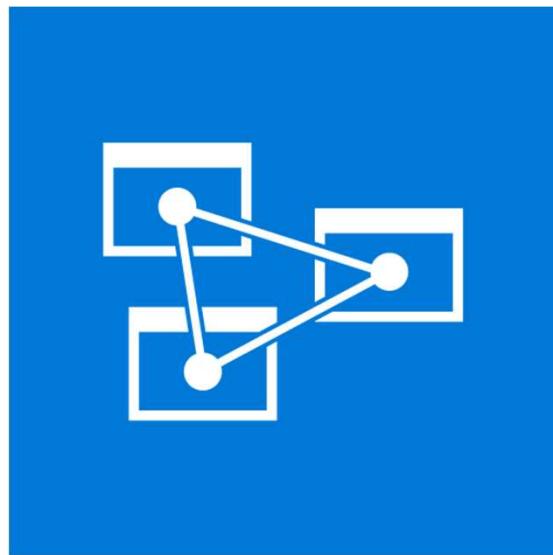


Power BI

Enterprise BI models require analysis services

- Model size with Power BI today is limited to 1 GB for shared and up to 10 GB on premium
- For larger and more complex models you need (Azure) Analysis services, this gives you:
 - Larger datasets up to 400 GB compressed
 - Ability to “pin to memory”
 - Connectivity from everywhere, even with other tools like Excel
 - Ability to use source control
 - Fine grained partitioning
 - Access to the DMV’s for diagnostics
- **Why do I need to use a “separate” service?**

Enterprise BI



Azure
Analysis Services

All BI users

Self-service BI
users



Power BI
Premium

Power BI

Azure AS vs. Power BI

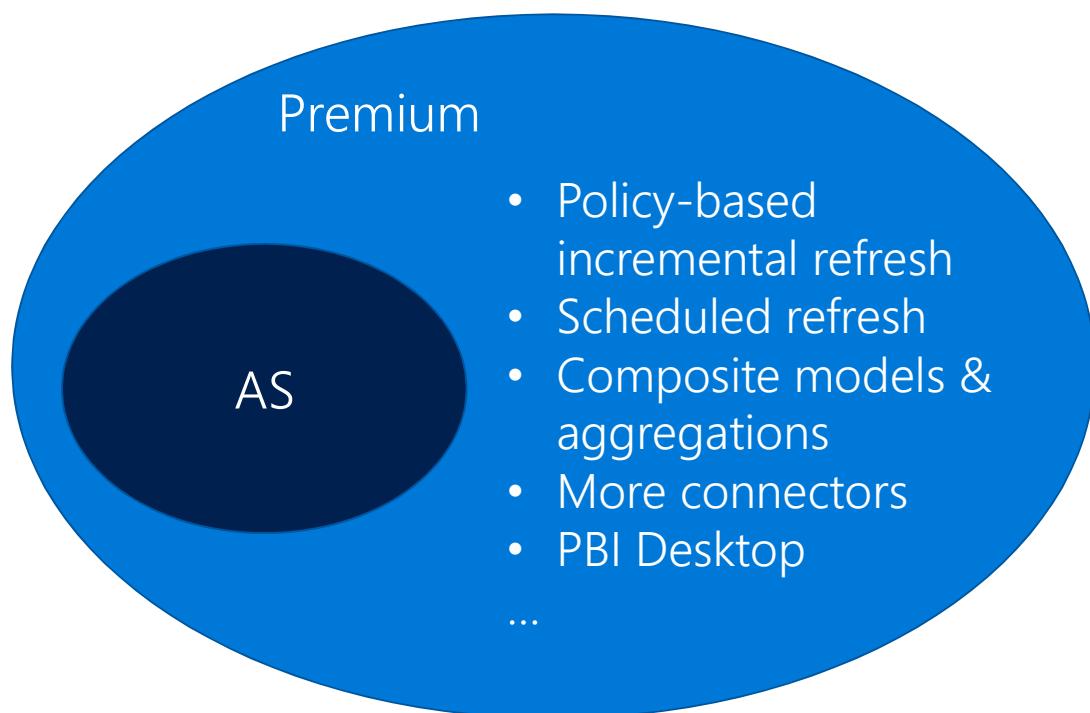


Power BI



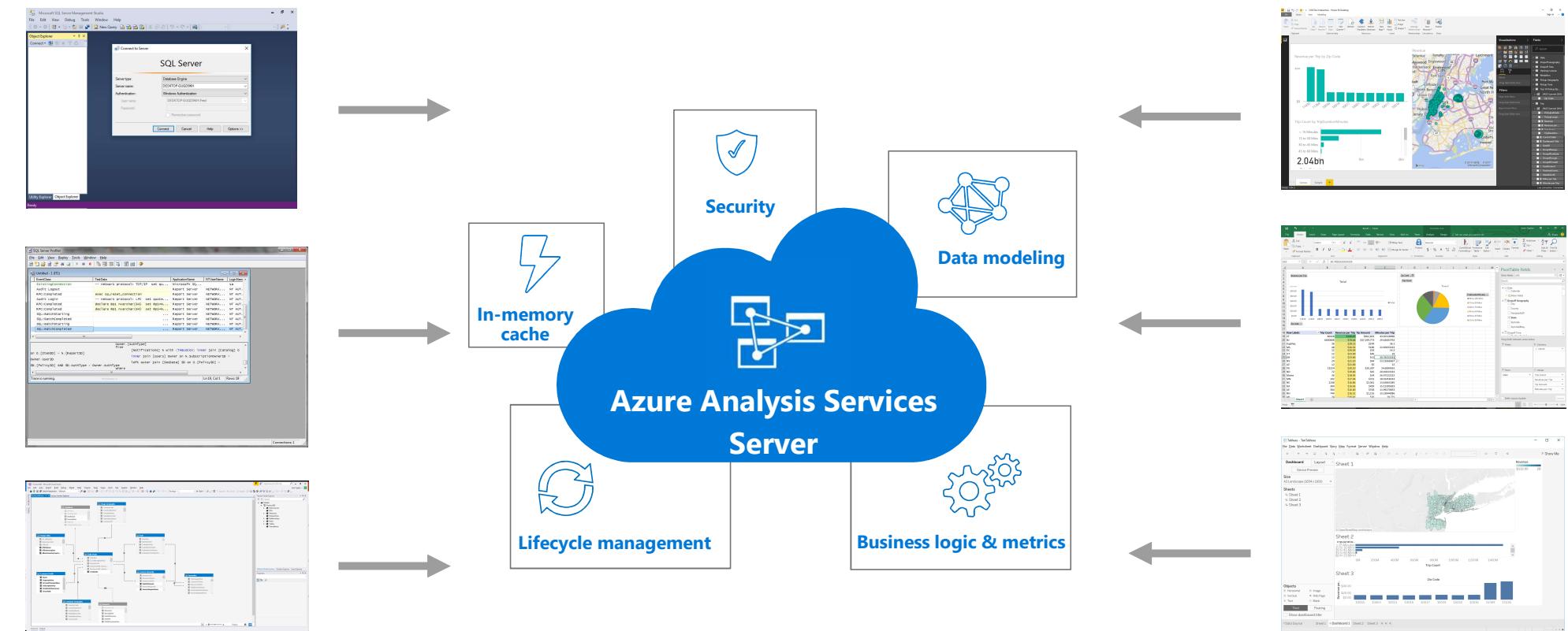
SQL Server
Analysis Services
(Tabular)

Power BI: a unified platform for self-service and enterprise BI



“XMLA Endpoints”

Connectivity: Analysis Services



Connectivity: Analysis Services



The screenshot displays three windows side-by-side, illustrating the process of an incremental refresh:

- Microsoft SQL Server Management Studio (SSMS):** The Object Explorer shows a connection to `pbi://api.powerbi.com/microsoft.com/Chwade Incremental Refresh`. A red box highlights this connection. The Details pane shows a table named `EventClass` with three rows:

EventClass	ActivityID	CPUTime	ClientProcessID	ConnectionID
VertiPaq SE Query Begin	C6BFE53...			2
DirectQuery Begin	C6BFE53...	16	25240	2
VertiPaq SE Query Begin	C6BFE53...			2

A red box highlights the first row. Below the table is a query window containing DAX code for calculating sales:

```
SET DC_KIND="AUTO";  
SELECT  
[Date (631)].[CalendarYear (648)] AS [Date (631)$CalendarYear (648)],  
SUM([FactInternetSalesAgg (2829)].[SalesAmount Sum (2835)]) AS [$Measure0]  
FROM [FactInternetSalesAgg (2829)]  
<  
Trace is running.
```
- Power BI:** The browser window shows the Power BI interface with the URL `https://msit.powerbi.com/groups/6d2ddd9e-d23a-4523-98ff-7d85d054`. It displays a list of datasets under the **Datasets** tab, including `Aggs Demo DQ - C`, `Internet Sales`, `IR_Taxi`, `IR_Taxi_PROD`, `TPC-H10`, and `TPC-H11`.
- DaxStudio:** The bottom window shows the results of a DAX query named `Aggs Hit Miss.dax`. A red box highlights the URL in the address bar: `powerbi://api.powerbi.com/v1.0/microsoft.com/Chwade Incremental Refresh`. The results pane shows the evaluated DAX code:

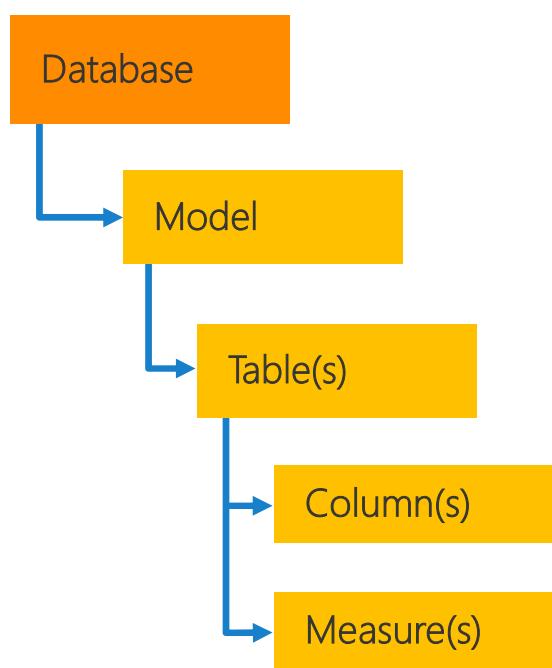
```
1 EVALUATE  
2 SUMMARIZECOLUMNS(  
3     'Date'[CalendarYear],  
4     "Sales", SUM('Sales')[SalesAmount])
```

Below the code is a detailed server timing report table:

Total	SE CPU	Line	Subclass	Duration	CPU	Rows	KB	Query
7 ms	0 ms	x0.0		2 Scan	0	0	13	1 SELECT 'D
7 ms	0 ms	FE		0 ms	0.0%			

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",  
      ...  
    }  
  }  
}
```

Demo: Connecting to Power BI with XMLA

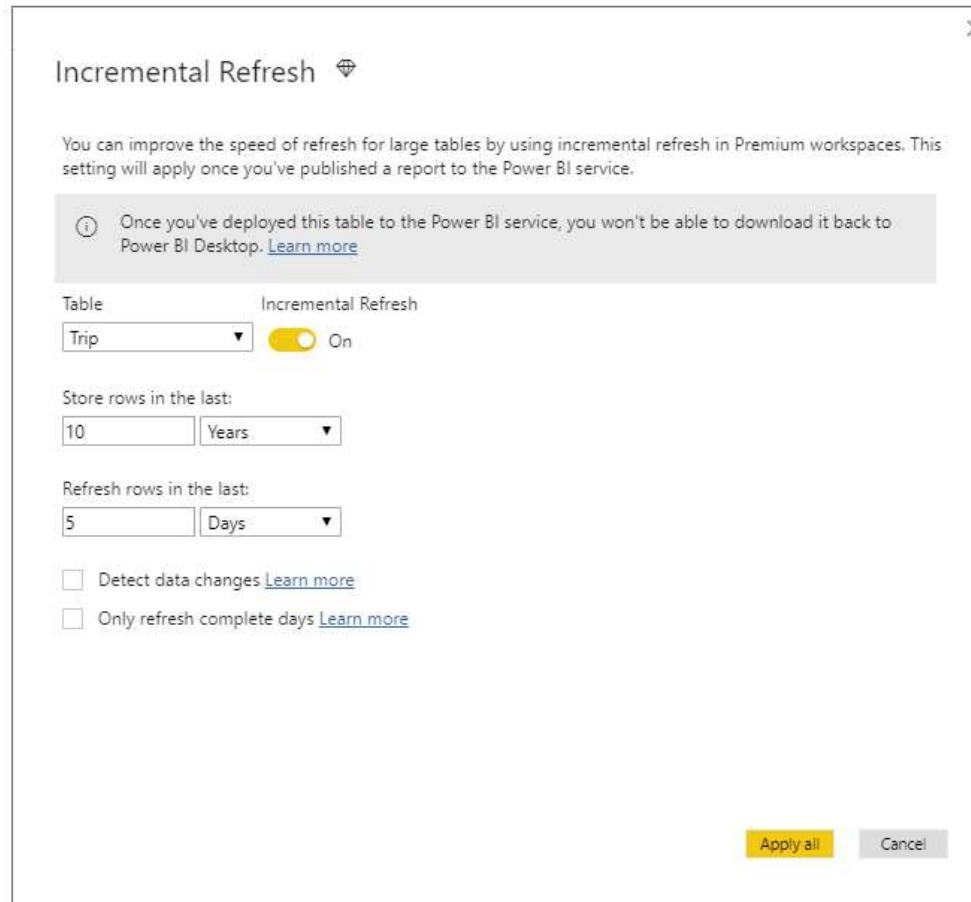
Incremental refresh

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

Incremental refresh

- Incremental refresh demo: <https://aka.ms/IncrementalRefreshDemo>

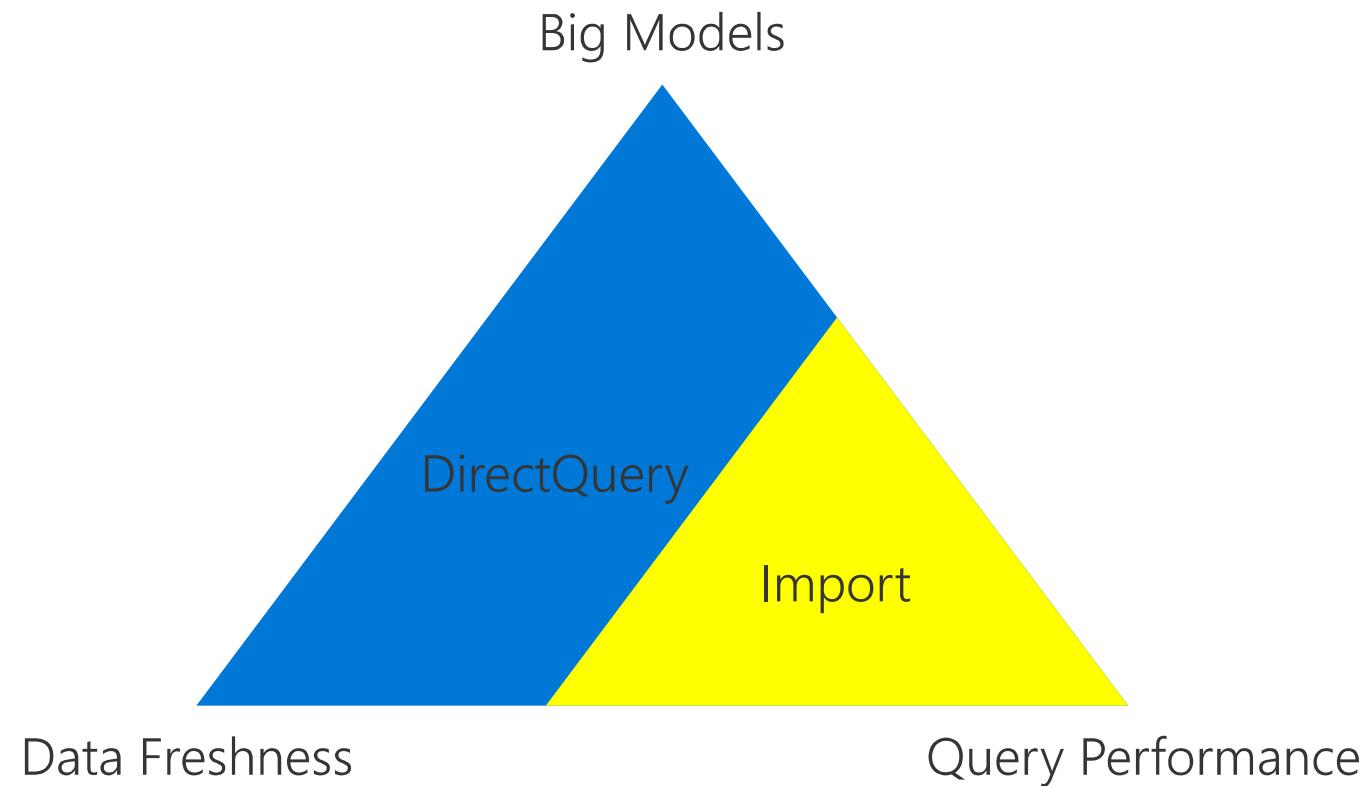


Demo: XMLA endpoints & incremental refresh

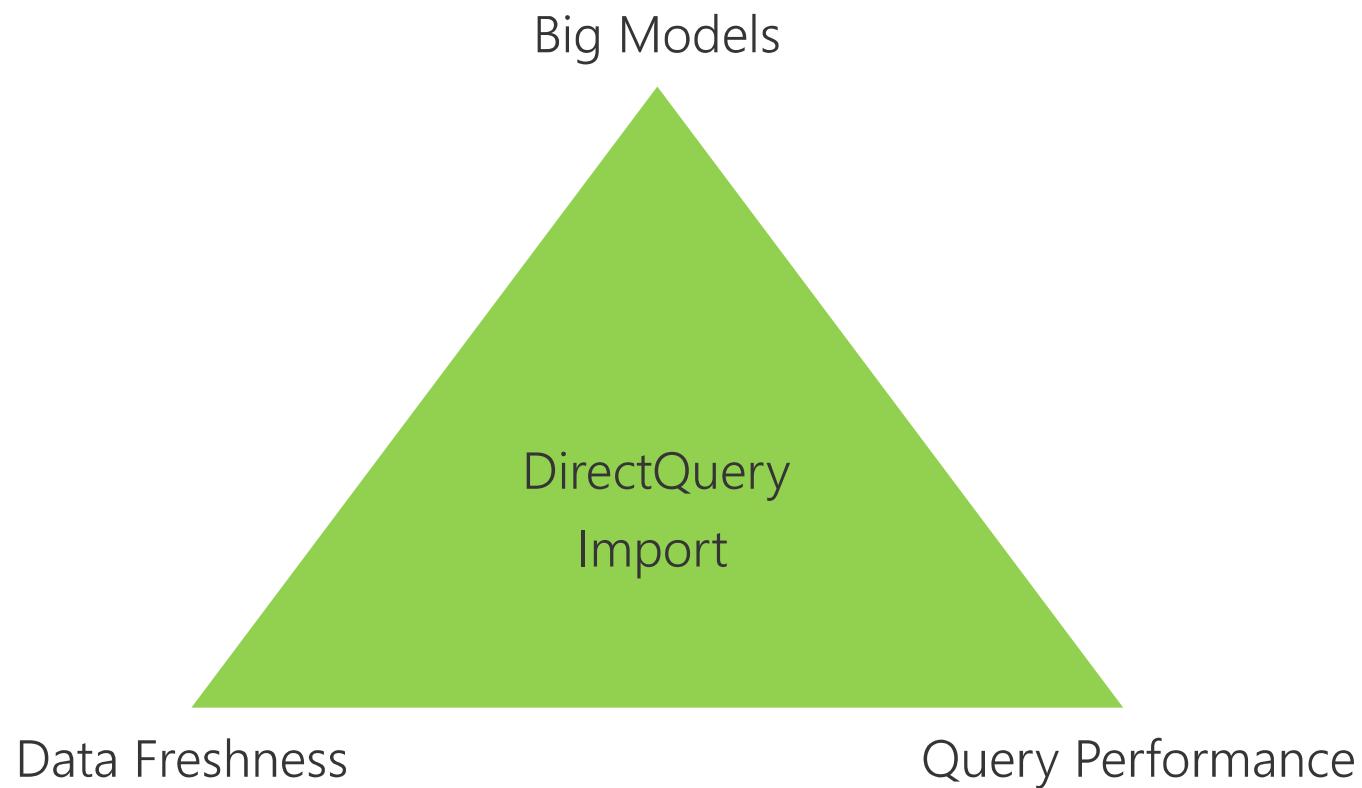
Aggregations

Aggregations: demo

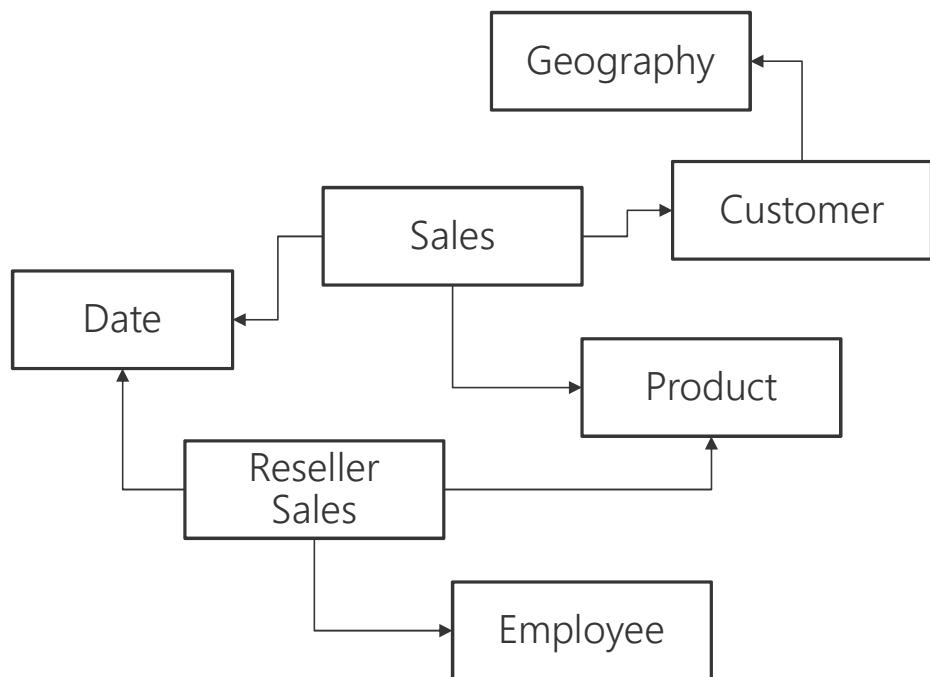
Import vs. DirectQuery



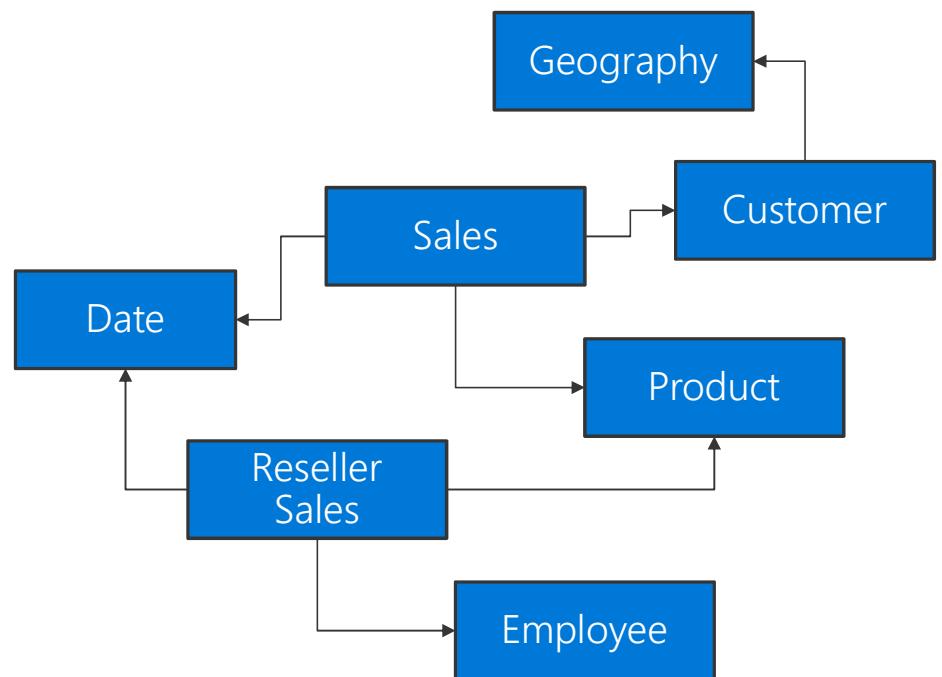
Import vs. DirectQuery



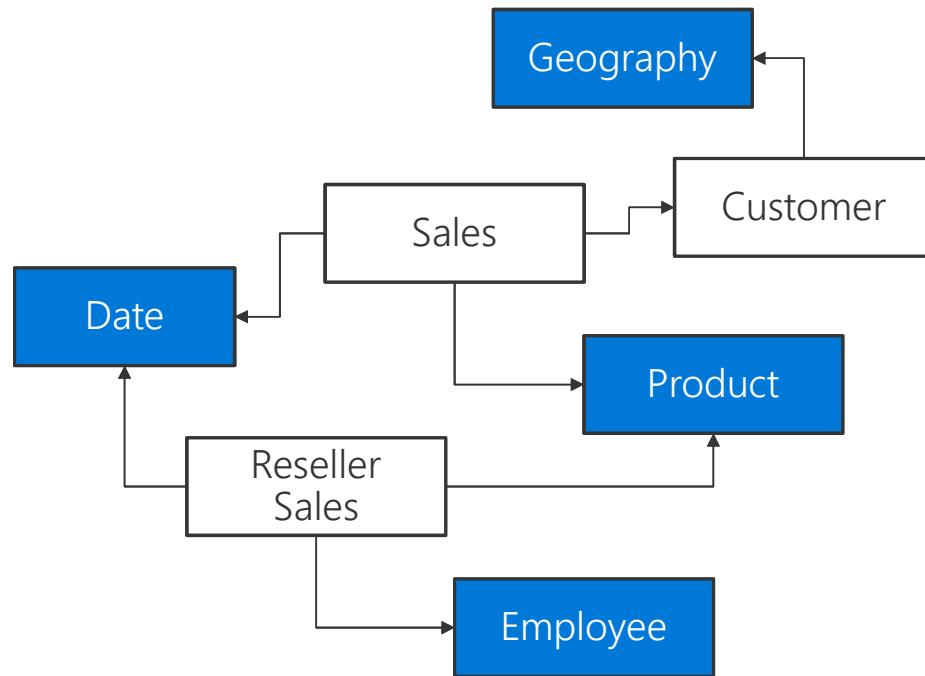
DirectQuery



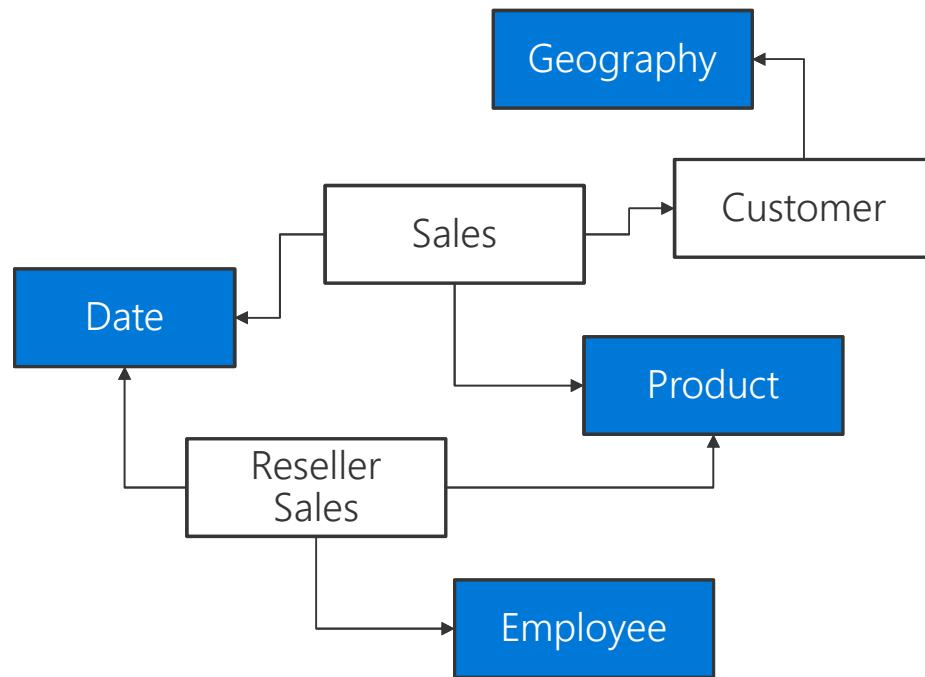
Import



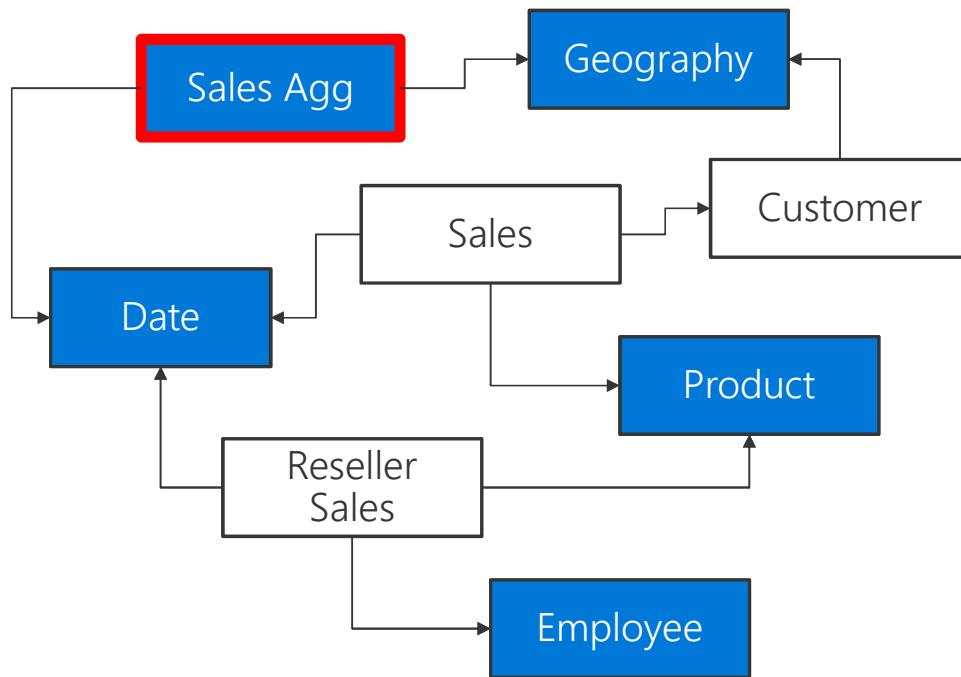
DirectQuery & Import



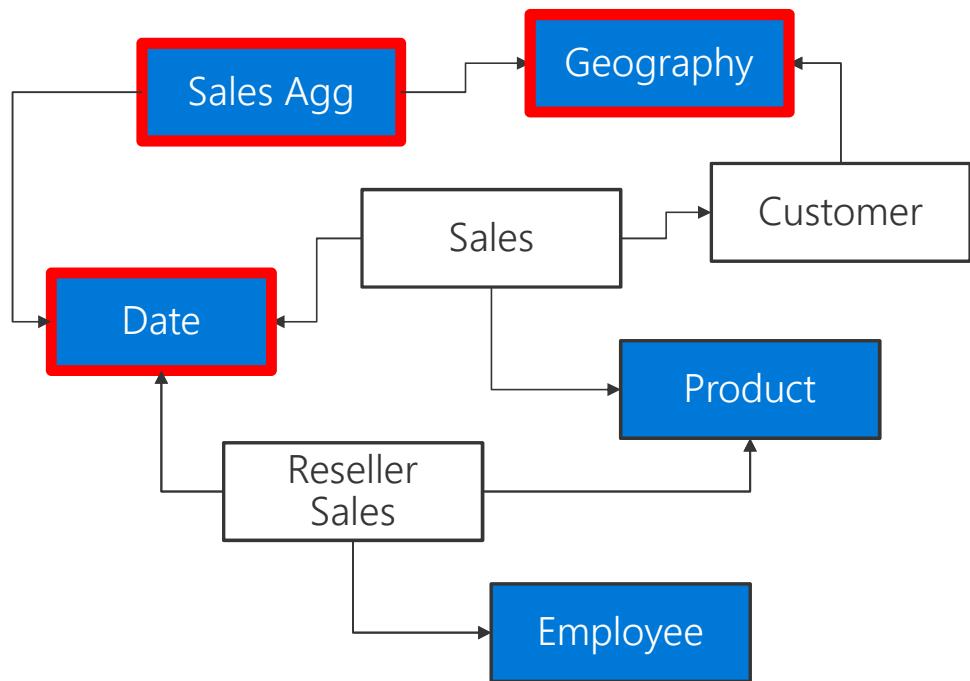
DirectQuery & Import



Aggregations



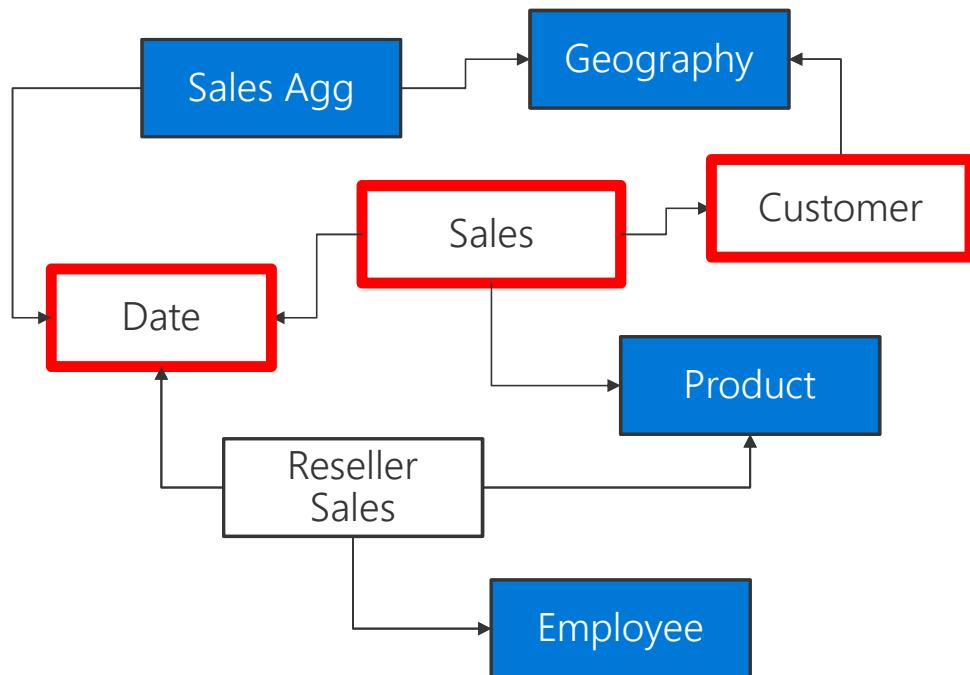
Aggregations



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

Hits in-memory cache

Aggregations



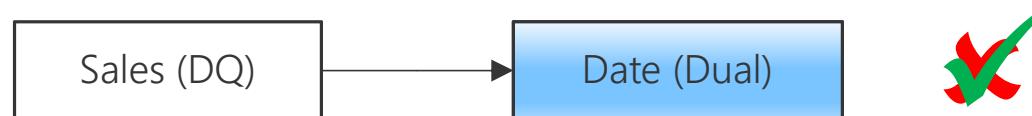
```
SummarizeColumns(  
    Date[Year],  
    Customer[Name],  
    "Sales", Sum(Sales[Amount])  
)
```

DirectQuery

Aggregations: How to set it up

Dual storage mode

- Avoids having to do the “join” in Power BI



Relationship rules for aggregation hits

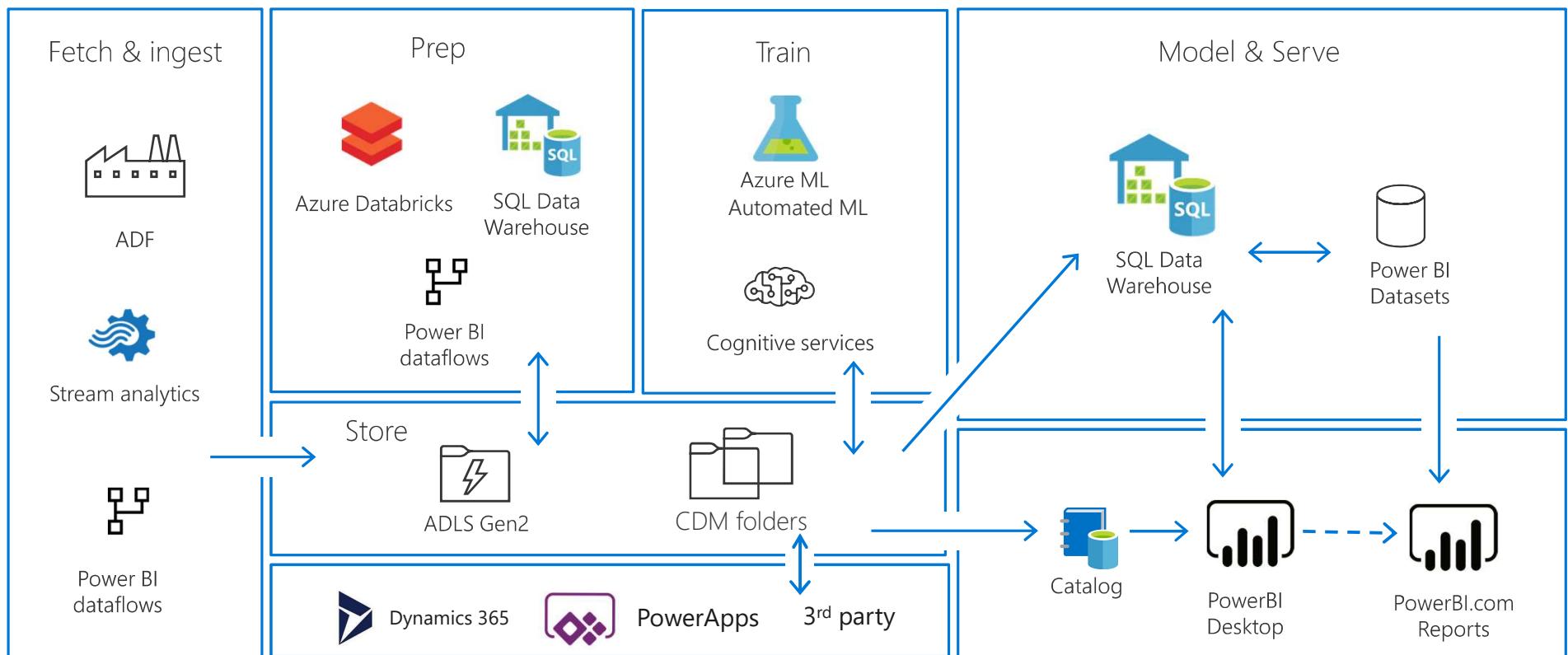
"Many side"	"One side"
Dual	Dual
Import	Dual or Import
DQ	Dual or DQ

(both sides are from a single source)

Advanced analytics



Topology for Power BI and Azure



Calculation groups

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 (as of CTP 2.3)!
 - Azure AS
 - Power BI Premium (XMLA endpoint enablement initially)
- Require new 1470 compat level

Demo: calculation groups

New DAX functions

Function name	Description
SELECTMEASURE()	Returns a reference to the measure currently in context.
SELECTMEASURENAME()	Returns a string containing the name of the measure currently in context.
ISSELECTEDMEASURE(M1, M2, ...)	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 ) )
)
```

Calculation groups

Table	Time Intelligence
Column	Time Calculation
Precedence	20

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>

Demo: Precedence

Reuse in regular measures as well..

```
=YTD Daily AVERAGE =  
CALCULATE (  
    [Sum of SalesAmount],  
    'Time Intelligence'[Time Calculation] = "YTD",  
    'Averages'[Average Calculation] = "Daily Average"  
)
```

Calculation groups

Internet Sales Time Intelligence

CalendarYear	InternetTotalSales
2012	
January	\$495,364
February	\$506,994
March	\$373,483
April	\$400,336
May	\$358,878
June	\$555,160
July	\$444,558
August	\$523,917
September	\$486,177
October	\$535,159
November	\$537,956
December	\$624,502
2013	
January	\$857,690
February	\$771,349
March	\$1,049,907
April	\$1,046,023
May	\$1,284,593
June	\$1,643,178
July	\$1,371,676
August	\$1,551,066
September	\$1,447,496

VISUALIZATIONS >

ROWS

- CalendarYear
- EnglishMonthName

COLUMNS

Add data fields here

VALUES

- InternetTotalSales

FILTERS

Visual level filters

- CalendarYear is 2012 or 2013
- EnglishMonthName is (All)

FIELDS

Search

- Averages
- DimCustomer
- DimEmployee
- DimGeography
- DimProduct
- DimProductCategory
- DimProductSubcategory
- FactInternetSales
- FactResellerSales
- Time Intelligence
- Time Calculation

Multiple calculation groups w/ precedence

Table	Averages
Column	Average Calculation
Precedence	10

Calculation Item	Expression
"Base"	SELECTEDMEASURE()
"Daily Average"	DIVIDE(SELECTEDMEASURE(), COUNTROWS(DimDate))

Precedence

YTD Daily Average = **CALCULATE** ([Sum of SalesAmount],
'Time Intelligence'[Time Calculation] = "YTD",
'Averages'[Average Calculation] = "Daily Average")

"YTD"	CALCULATE(SELECTEDMEASURE() , DATESYTD(DimDate[Date]))	Precedence 20
"DAILY AVERAGE"	DIVIDE(SELECTEDMEASURE() , COUNTROWS(DimDate))	Precedence 10
"YTD" "DAILY AVERAGE"	CALCULATE(DIVIDE(SELECTEDMEASURE() , COUNTROWS(DimDate)) , DATESYTD(DimDate[Date]))	NOT DIVIDE(CALCULATE(SELECTEDMEASURE() , DATESYTD(DimDate[Date])) , COUNTROWS(DimDate))

Not working (yet)

- OLS/RLS
- Implicit measures

DiscourageImplicitMeasures=true

Dynamic format strings: currency conversion

Demo: Currency conversion

Calculation groups: currency conversion

The screenshot shows the Power BI Data view on the left and the visualization builder interface on the right.

Data View:

CalendarYear	Sales
2013	
January	\$857,690
February	\$771,349
March	\$1,049,907
April	\$1,046,023
May	\$1,284,593
June	\$1,643,178
July	\$1,371,676
August	\$1,551,066
September	\$1,447,496
October	\$1,673,293
November	\$1,780,920
December	\$1,874,360

Visualizations: A grid of visualization icons.

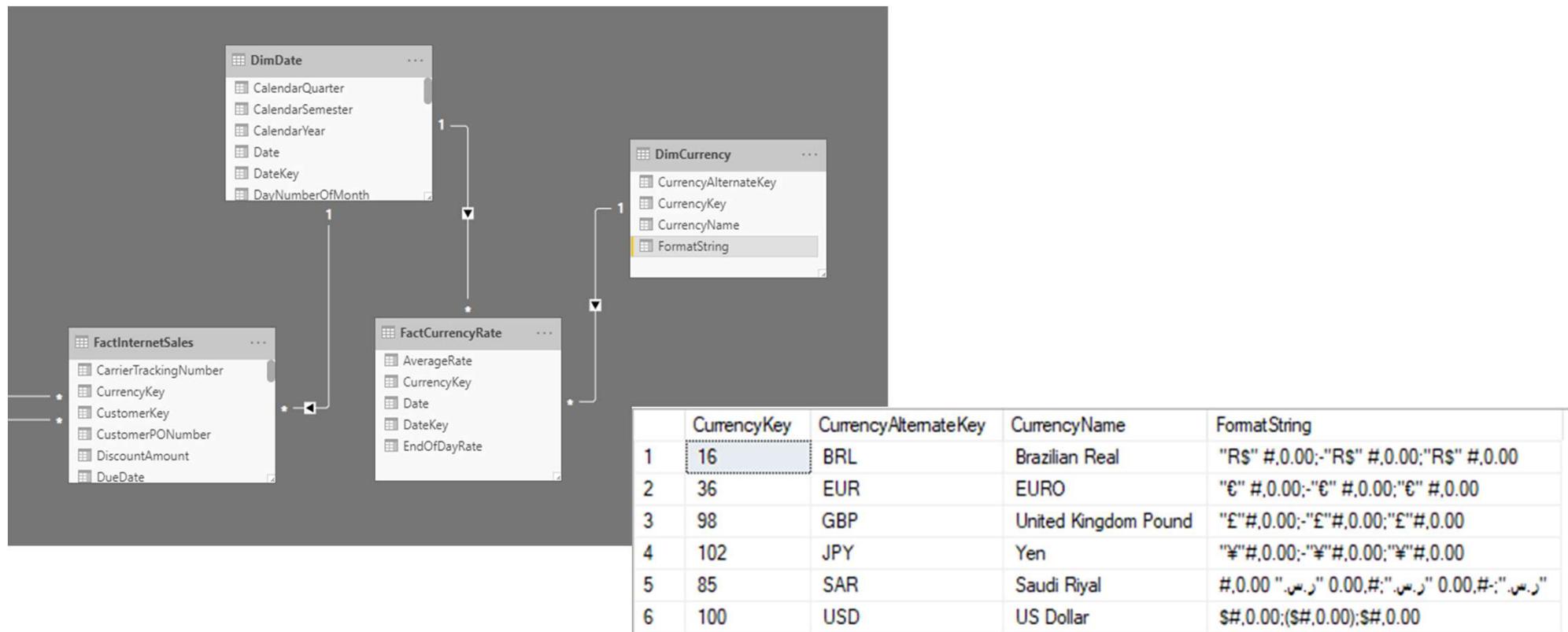
Fields:

- Search bar: Search
- Category tree:
 - Averages
 - Currency Conversion
 - DimCurrency
 - CurrencyAlternateKey
 - CurrencyKey
 - CurrencyName** (selected)
 - FormatString
 - DimCustomer
 - DimDate
 - DimProduct
 - FactCurrencyRate
 - FactInternetSales
 - Time Intelligence

Selected Item Details:

- Name: 'DimCurrency'[CurrencyName]
- Columns:
 - Calendar
 - CalendarYear
 - EnglishMonthName
- Values: Sales
- Filters: None

Dynamic format strings: currency conversion



Dynamic format strings: currency conversion

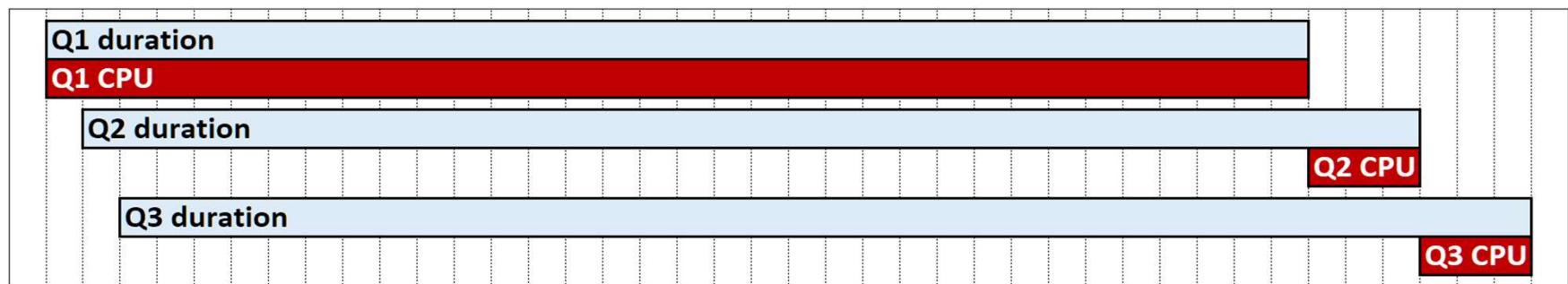
Table	Currency Conversion	
Column	Conversion Calculation	
Precedence	5	
Calculation Item	"No Conversion"	"Converted Currency"
Expression	SELECTEDMEASURE()	<pre>IF (--Check one currency in context and not US Dollar, which is the pivot currency: SELECTEDVALUE (DimCurrency[CurrencyName], "US Dollar") = "US Dollar", SELECTEDMEASURE (), SUMX (VALUES(DimDate[Date]), CALCULATE(DIVIDE(SELECTEDMEASURE(), MAX(FactCurrencyRate[EndOfDayRate])))))</pre>
Format String Expression		<pre>SELECTEDVALUE(DimCurrency[FormatString], SELECTEDMEASUREFORMATSTRING())</pre>

Query interleaving

SSAS queries

- Queries in the FE are single threaded
- This means when you have a very expensive query all others are blocked until it finishes.
- This is a problem for concurrency \ using many visuals.

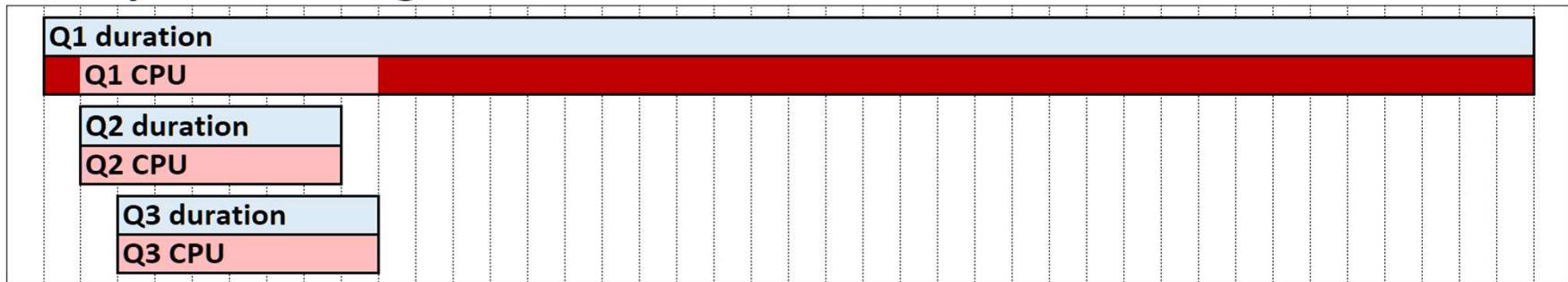
FIFO



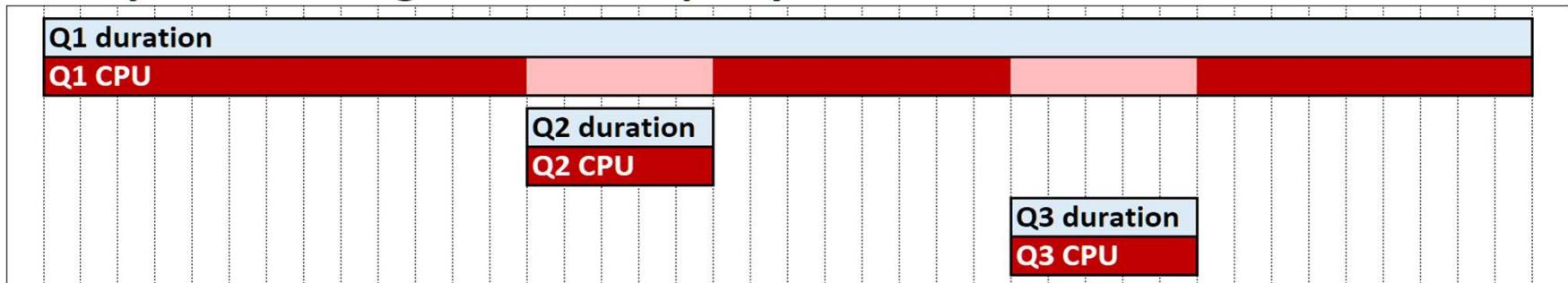
SSAS Query interleaving

- Going forward queries will be interleaved

Query interleaving



Query interleaving with short query bias



What's new and coming in Azure AS & SSAS 2019

- Modeling & Analytics
 - Calculation groups
 - Many-to-many relationships
 - SuperDAXMD (Multidimensional)
- New DAX functions:
 - `IsInScope()`, `SelectedValue()` ...
- Resource governance
 - Control Power BI background refreshes
 - *ClientCacheRefreshPolicy*
 - Memory management:
 - *QueryMemoryLimit*
 - *RowsetSerializationLimit*
 - Query interleaving (coming soon in Azure AS):
 - *SchedulingBehavior*