

[www.sqlbi.com](http://www.sqlbi.com)



**Microsoft Partner**

Gold Business Intelligence  
Gold Data Platform

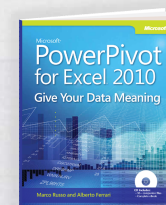
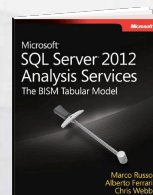
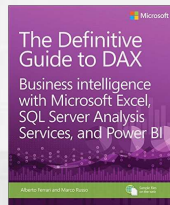
**SSAS  
MAESTRO**  
by Microsoft



# Who We Are



- BI Experts and Consultants
- Founders of [www.sqlbi.com](http://www.sqlbi.com)
  - Problem Solving
  - Complex Project Assistance
  - Data Warehouse Assessments and Development
  - Courses, Trainings and Workshops
- Book Writers
- Microsoft Gold Business Intelligence Partners
- SSAS Maestros – MVP – MCP



# The DAX Language

- Language of
  - Power Pivot
  - Power BI
  - SSAS Tabular
- DAX is simple, but it is not easy
- New programming concepts and patterns

Introduction to the DAX language

# Introduction to DAX



# What is DAX?

- Programming language
  - Power Pivot
  - SSAS Tabular
  - Power BI
- Resembles Excel
  - Because it was born with PowerPivot, in 2010
- Important differences
  - No concept of «row» and «column»
  - Different type system
- Many new functions
- Designed for data models and business calculations

# Functional Language

DAX is a functional language, the execution flows with function calls, here is an example of a DAX formula

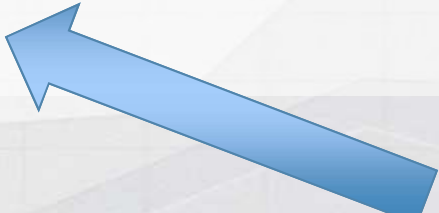
```
=SUMX (  
    FILTER (  
        VALUES ( 'Date'[Year] ),  
        'Date'[Year] < 2005  
    ),  
    IF (  
        'Date'[Year] >= 2000,  
        [Sales Amount] * 100,  
        [Sales Amount] * 90  
    )  
)
```

# If it is not formatted, it is not DAX.

Code formatting is of paramount importance in DAX.

```
=SUMX(FILTER(VALUES('Date'[Year]), 'Date'[Year]<2005), IF('Date'[Year]>=2000, [Sales Amount]*100, [Sales Amount]*90))
```

```
=SUMX (  
    FILTER (  
        VALUES ( 'Date'[Year] ),  
        'Date'[Year] < 2005  
    ),  
    IF (  
        'Date'[Year] >= 2000,  
        [Sales Amount] * 100,  
        [Sales Amount] * 90  
    )  
)
```



[www.daxformatter.com](http://www.daxformatter.com)

# Calculated Columns



- Columns computed using DAX
- Always computed for the current row
- Product[Price] means
  - The value of the Price column (explicit)
  - In the Product table (explicit, optional)
  - For the current row (implicit)
  - Different for each row



# Measures



- Written using DAX
- Do not work row by row
- Instead, use tables and aggregators
- Do not have the «current row» concept
- Examples
  - GrossMargin
    - is a calculated column
    - but can be a measure too
  - GrossMargin %
    - needs to be a measure

# Naming Convention

- Measures do not belong to a table
  - Avoid table name
  - [Margin%] instead of Sales[Margin%]
  - Easier to move to another table
  - Easier to identify as a measure
- Thus
  - Calculated columns → Table[Column]
  - Measures → [Measure]

# Measures vs Calculated Columns

- Use a column when
  - Need to slice or filter on the value
- Use a measure
  - Calculate percentages
  - Calculate ratios
  - Need complex aggregations
- Space and CPU usage
  - Columns consume memory
  - Measures consume CPU

# Aggregation Functions

- Useful to aggregate values
  - SUM
  - AVERAGE
  - MIN
  - MAX
- Work only on numeric columns
- Aggregate only one column
  - SUM ( Orders[Price] )
  - SUM ( Orders[Price] \* Orders[Quantity] )

# The «X» Aggregation Functions

- Iterators: useful to aggregate formulas
  - SUMX
  - AVERAGEX
  - MINX
  - MAXX
- Iterate over the table and evaluate the expression for each row
- Always receive two parameters
  - Table to iterate
  - Formula to evaluate for each row

# Example of SUMX

For each row in the Sales table, evaluates the formula, then sum up all the results  
Inside the formula, there is a «current row»

```
SUMX (  
    Sales,  
    Sales[Price] * Sales[Quantity]  
)
```

# Using Variables

Very useful to avoid repeating subexpressions in your code

```
VAR
    TotalQuantity = SUM ( Sales[Quantity] )

RETURN

    IF (
        TotalQuantity > 1000,
        TotalQuantity * 0.95,
        TotalQuantity * 1.25
    )
```

# Date Functions

- Many useful functions
  - DATE, DATEVALUE, DAY, EDATE,
  - EOMONTH, HOUR, MINUTE,
  - MONTH, NOW, SECOND, TIME,
  - TIMEVALUE, TODAY, WEEKDAY,
  - WEEKNUM, YEAR, YEARFRAC
- Time Intelligence functions will be covered later



Some functions return tables instead of values

# Table Functions



# Table Functions

- Basic functions that work on full tables
  - FILTER
  - ALL
  - VALUES
  - DISTINCT
  - RELATEDTABLE
- Their result is often used in other functions
- They can be combined together to form complex expressions
- We will discover many other table functions later in the course

# CALCULATED TABLES



- In the newest versions of Tabular (and in Power BI Desktop), you can create calculated tables
- User interface is different in different products

# Filtering a Table

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

```
SUMX (  
  FILTER (  
    Orders,  
    Orders[Price] > 1  
  ),  
  Orders[Quantity] * Orders[Price]  
)
```

The screenshot shows the PowerPivot interface with a PivotTable and the Field List. The PivotTable has 'Channel' and 'City' as filters, 'CalcAmount' as the value field, and 'Size' as the column labels. The 'Grand Total' for the 'Large' size is 192. The Field List on the right shows the 'Orders' table with fields: City, Channel, Color, Size, Quantity, Price, Amount, and CalcAmount.

Channel	City	CalcAmount	Size	Grand Total
Internet			Large	192
Store			Large	112
	New York		Large	4
	Paris		Large	1
	Torino		Large	4
			Small	160
			Grand Total	464

# The FILTER Function



- FILTER
  - Adds a new condition
  - Restricts the number of rows of a table
  - Returns a table
  - Can be iterated by an «X» function
- Needs a table as input
- The input can be another FILTER

# Removing Filters

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

```
SUMX (  
    ALL ( Orders ),  
    Orders[Quantity] * Orders[Price]  
)
```

	A	B	C	D	E	F	G	H	I	J
1										
2										
3		Channel								
4		Internet								
5		Store								
6										
7										
8		City								
9										
10										
11		New York								
12		Paris								
13		Torino								
14										

Column Labels	Large	Small	Total AllAmount	Total Sum of Amount
Row Labels	AllAmount	Sum of Amount	AllAmount	Sum of Amount
Green	749	192	749	128
Red	749	112	749	160
Grand Total	749	304	749	288

PowerPivot Field List
Choose fields to add to report
Search
Orders
<input checked="" type="checkbox"/> City
<input checked="" type="checkbox"/> Channel
<input checked="" type="checkbox"/> Color
<input checked="" type="checkbox"/> Size
<input type="checkbox"/> Quantity
<input type="checkbox"/> Price
<input checked="" type="checkbox"/> Amount
<input checked="" type="checkbox"/> AllAmount

# The ALL Function



- ALL
  - Returns all the rows of a table
  - Ignores the filter context
  - Returns a table
  - That can be iterated by an «X» function
- Needs a table as input
- Can be used with a single column
  - **ALL ( Customers[CustomerName] )**
  - The result contains a table with one column

## Mixing Filters

- Table functions can be mixed
- Each one requires a table
- Each one returns a table
- **FILTER** ( **ALL** ( Table ), Condition )
  - Puts a filter over the entire table
  - Ignores the current filter context



# Mixing Filters

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

```
SUMX (
  FILTER (
    ALL( Orders ),
    Orders[Channel]="Internet"
  ),
  Orders[Quantity] * Orders[Price]
)
```

The screenshot shows a PivotTable with the following data:

Row Labels	AllInternet	Sum of Amount	AllInternet	Sum of Amount	Total AllInter	Total Sum of A
Green	592	192	592	128	592	320
Red	592	112	592	160	592	272
<b>Grand Total</b>	<b>592</b>	<b>304</b>	<b>592</b>	<b>288</b>	<b>592</b>	<b>592</b>

The Field List on the right shows the Orders table with the following fields selected:

- City
- Channel
- Color
- Size
- Quantity
- Price
- Amount
- AllInternet

# Mixing Filters

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

```
SUMX (
  FILTER (
    ALL( Orders ),
    Orders[Channel]="Internet"
  ),
  Orders[Quantity] * Orders[Price]
)
```

Channel: Internet, Store

City: New York, Paris, Torino

Column Labels	Large	Small	Total AllInter	Total Sum of A
Row Labels	AllInternet	Sum of Amount	AllInternet	Sum of Amount
Green	592	192	592	128
Red	592	112	592	160
Grand Total	592	304	592	288

PowerPivot Field List: Orders (City, Channel, Color, Size, Quantity, Price, Amount, AllInternet)

# Mixing Filters

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

```
SUMX (
  FILTER (
    ALL( Orders ),
    Orders[Channel]="Internet"
  ),
  Orders[Quantity] * Orders[Price]
)
```

The screenshot shows a PivotTable with the following data:

Channel	City	Large	Small	Total AllInternet	Total Sum of Amount
Internet	New York	16	32	48	104
Internet	Paris	1	2	3	13
Internet	Torino	4	8	12	11
Store	New York	4	8	12	11
Store	Paris	1	2	3	13
Store	Torino	4	8	12	11
<b>Grand Total</b>		<b>29</b>	<b>58</b>	<b>87</b>	<b>136</b>

The 'AllInternet' column is highlighted with a blue box, and its value 592 is circled. A blue arrow points from the DAX formula in the top right to this value.

The Field List on the right shows the 'Orders' table with fields City, Channel, Color, Size, Quantity, Price, Amount, and AllInternet selected.

# Mixing Filters

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

```
SUMX (
  FILTER (
    ALL( Orders ),
    Orders[Channel]="Internet"
  ),
  Orders[Quantity] * Orders[Price]
)
```

The screenshot shows a PivotTable with the following data:

Row Labels	AllInternet	Sum of Amount	AllInternet	Sum of Amount	Total AllInter	Total Sum of A
Green	592	192	592	128	592	320
Red	592	112	592	160	592	272
<b>Grand Total</b>	<b>592</b>	<b>304</b>	<b>592</b>	<b>288</b>	<b>592</b>	<b>592</b>

The Field List on the right shows the Orders table with the following fields selected:

- City
- Channel
- Color
- Size
- Quantity
- Price
- Amount
- AllInternet

# DISTINCT



Returns the distinct values of a column, only the ones visible in the current context

```
NumOfProducts :=  
  
COUNTROWS (  
    DISTINCT ( Product[ProductCode] )  
)
```

# RELATEDTABLE

Returns a table with all the rows related with the current one.

```
NumOfProducts =  
COUNTROWS (  
    RELATEDTABLE ( Product )  
)
```

# Example of RELATEDTABLE

Compute the number of red products for a category. Build a calculated column in the Categories table:

```
NumOfRedProducts =  
  
COUNTROWS (  
    FILTER (  
        RELATEDTABLE ( Product ),  
        Product[Color] = "Red"  
    )  
)
```

Let us take a look at how DAX works

# Evaluation Contexts



# What is an Evaluation Context?

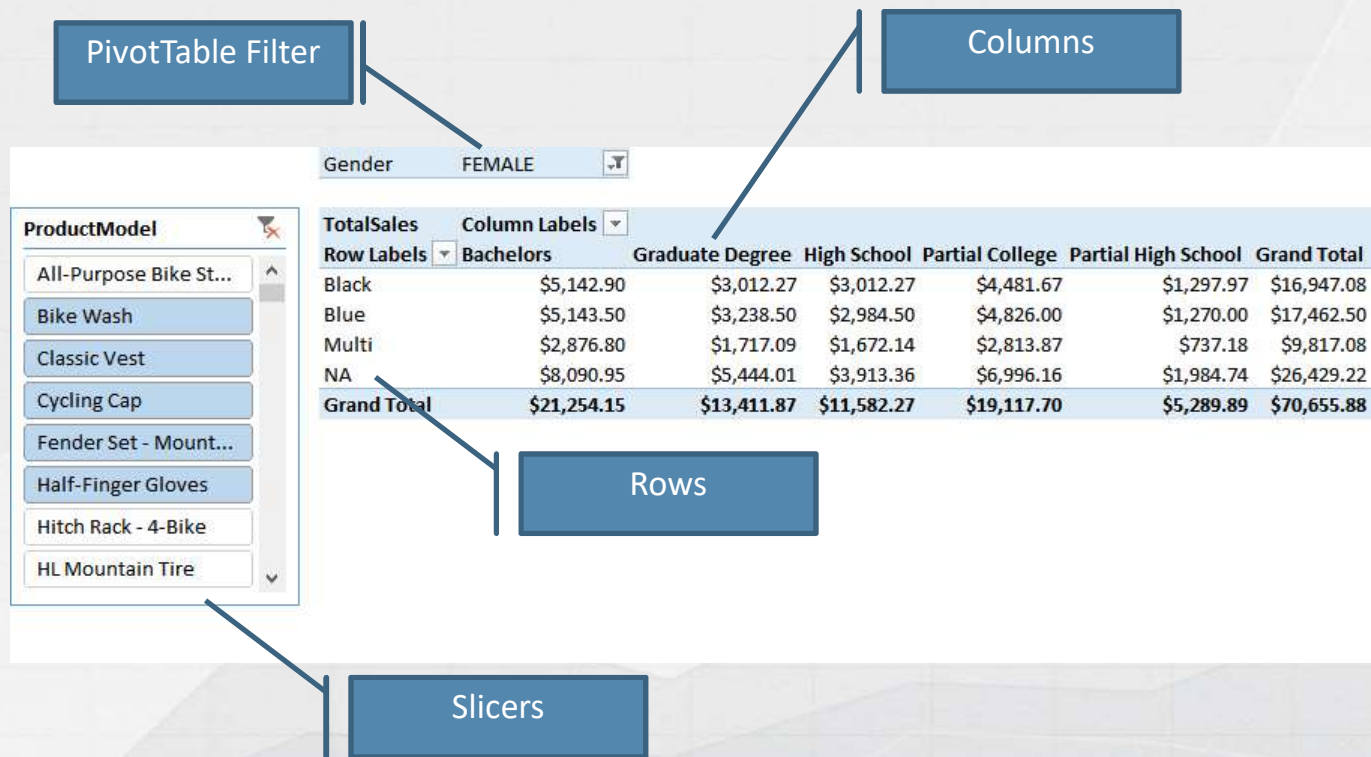
```
TotalSales := SUM ( Sales[SalesAmount] )
```

TotalSales	Row Labels	TotalSales
\$29,358,677.22	Black	\$8,838,411.96
	Blue	\$2,279,096.28
	Multi	\$106,470.74
	NA	\$435,116.69
	Red	\$7,724,330.52
	Silver	\$5,113,389.08
	White	\$5,106.32
	Yellow	\$4,856,755.63
	Grand Total	\$29,358,677.22

Numbers are sliced by color, i.e. the formula is NOT computing sum of sales, it is computing it for only a subset of the data model

The value of a formula depends on its context

# Sources of the Context



# Example of a Filter Context

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

	A	B	C	D	E	F	G
8							
9							
10							
11							
12							
13							

**Channel**

Internet Store

**City**

New York

Paris

Torino

**Sum of Quantity** Column Labels

Row Labels Large Small Grand Total

Green 64 128 192

Red 16 32 48

**Grand Total** 80 160 240



# Filter Context

- Defined by
  - Row Selection
  - Column Selection
  - Report Filters
  - Slicers Selection
- Rows outside of the filter context
  - Are not considered for the computation
- Defined automatically by PivotTable, can be created with specific functions too

# Row Context

- Defined by
  - Calculated column definition
    - Defined automatically for each row
  - Row Iteration functions
    - SUMX, AVERAGEX ...
    - All «X» functions and iterators
    - Defined by the user formulas
- Needed to evaluate column values, it is the concept of “current row”

# SUMX ( Orders, [Quantity]\*[Price] )

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9

	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

$$16 \times 7 = 112$$

$$32 \times 5 = 160$$

$$64 \times 3 = 192$$

$$128 \times 1 = 128$$

SUM = 592

Excel screenshot showing the PowerPivot Field List and a PivotTable.

**PowerPivot Field List:**

- Choose fields to add to report:
- Search:
- Orders
  - ☒ City
  - ☒ Channel
  - ☒ Color
  - ☒ Size
  - ☐ Quantity
  - ☐ Price
  - ☐ Amount
  - ☒ CalcAmount

**PivotTable:**

	Column Labels		
Row Labels	Large	Small	Grand Total
Green	192	128	320
Red	112	160	272
<b>Grand Total</b>	<b>304</b>	<b>288</b>	<b>592</b>

# There Are Always Two Contexts

- Filter context
  - Filters tables
  - Might be empty
    - All the tables are visible
    - But this never happens in the real world
- Row context
  - Iterates rows
  - For the rows active in the filter context
  - Might be empty
    - There is no iteration running
- Both are «evaluation contexts»

# Filtering a Table

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

```
SUMX (  
  FILTER (  
    Orders,  
    Orders[Price] > 1  
  ),  
  Orders[Quantity] * Orders[Price]  
)
```

PowerPivot Field List

Choose fields to add to report:

Search

Orders

- ☒ City
- ☒ Channel
- ☒ Color
- ☒ Size
- ☐ Quantity
- ☐ Price
- ☐ Amount
- ☒ CalcAmount

Channel: Internet, Store

City: New York, Paris, Torino

CalcAmount

Column Labels: Large, Small, Grand Total

Row Labels: Green, Red, Grand Total

Green	192		192
Red	112	160	272
Grand Total	304	160	464

# Removing Filters

City	Channel	Color	Size	Quantity	Price
Paris	Store	Red	Large	1	15
Paris	Store	Red	Small	2	13
Torino	Store	Green	Large	4	11
New York	Store	Green	Small	8	9
	Internet	Red	Large	16	7
	Internet	Red	Small	32	5
	Internet	Green	Large	64	3
	Internet	Green	Small	128	1

```
SUMX (
  ALL ( Orders ),
  Orders[Quantity] * Orders[Price]
)
```

	A	B	C	D	E	F	G	H	I	J
1										
2										
3		Channel								
4		Internet								
5		Store								
6										
7										
8		City								
9										
10										
11		New York								
12		Paris								
13		Torino								
14										

Column Labels	Large	Small	Total AllAmount	Total Sum of Amount
Row Labels	AllAmount	Sum of Amount	AllAmount	Sum of Amount
Green	749	192	749	128
Red	749	112	749	160
<b>Grand Total</b>	<b>749</b>	<b>304</b>	<b>749</b>	<b>288</b>

PowerPivot Field List
Choose fields to add to report
Search
Orders
<input checked="" type="checkbox"/> City
<input checked="" type="checkbox"/> Channel
<input checked="" type="checkbox"/> Color
<input checked="" type="checkbox"/> Size
<input type="checkbox"/> Quantity
<input type="checkbox"/> Price
<input checked="" type="checkbox"/> Amount
<input checked="" type="checkbox"/> AllAmount

The queen of all DAX functions 😊

# CALCULATE



# CALCULATE Syntax

Filters are evaluated in the outer filter context, then combined together in AND and finally used to build a new filter context into which DAX evaluates the expression

```
CALCULATE (  
    Expression,  
    Filter1,  
    ...  
    Filtern  
)
```

**Repeated many times, as needed**



# CALCULATE Examples



Compute the sum of sales where the price is more than \$100.00

```
NumOfBigSales :=
```

```
CALCULATE (  
    SUM ( Sales[SalesAmount] ),  
    Sales[SalesAmount] > 100  
)
```

Filter and SUM are  
on the same table.  
You can obtain the same  
result using FILTER

# Filters are tables

Each filter is a table. Boolean expressions are nothing but shortcuts for table expressions

```
CALCULATE (  
    SUM ( Sales[SalesAmount] ),  
    Sales[SalesAmount] > 100  
)
```

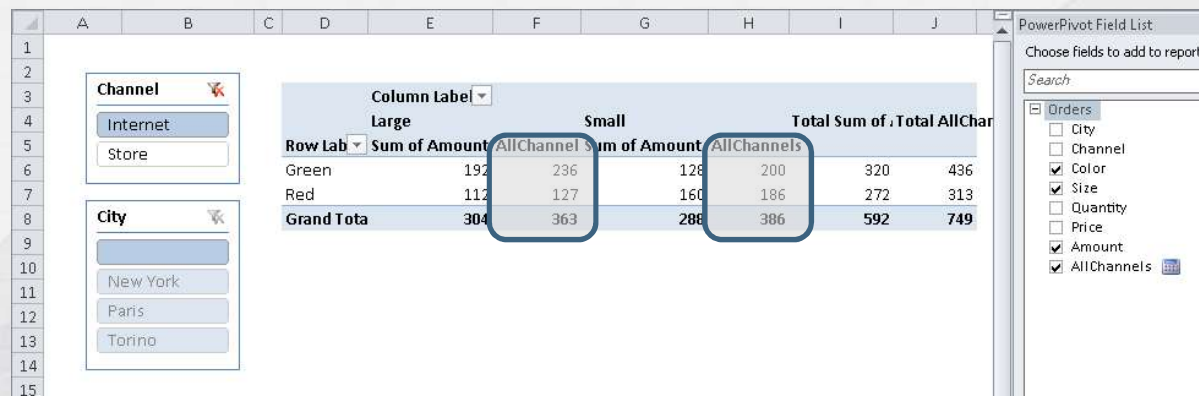
Is equivalent to

```
CALCULATE (  
    SUM ( Sales[SalesAmount] ),  
    FILTER (  
        ALL ( Sales[SalesAmount] ),  
        Sales[SalesAmount] > 100  
    )  
)
```

# Clear filter on one column only

```
CALCULATE (  
    SUMX (  
        Orders,  
        Orders[Amount]  
    ),  
    ALL ( Orders[Channel] )  
)
```

ALL used with a  
single column  
table



Channel

Internet

Store

City

New York

Paris

Torino

Column Label	Large	Small	Total Sum of .Total AllChar
Row Label	Sum of Amount	Sum of Amount	Sum of Amount
Green	192	236	128
Red	112	127	160
Grand Total	304	363	288

PowerPivot Field List

Choose fields to add to report:

Search

Orders

- ☐ City
- ☐ Channel
- ☒ Color
- ☒ Size
- ☐ Quantity
- ☐ Price
- ☒ Amount
- ☒ All Channels

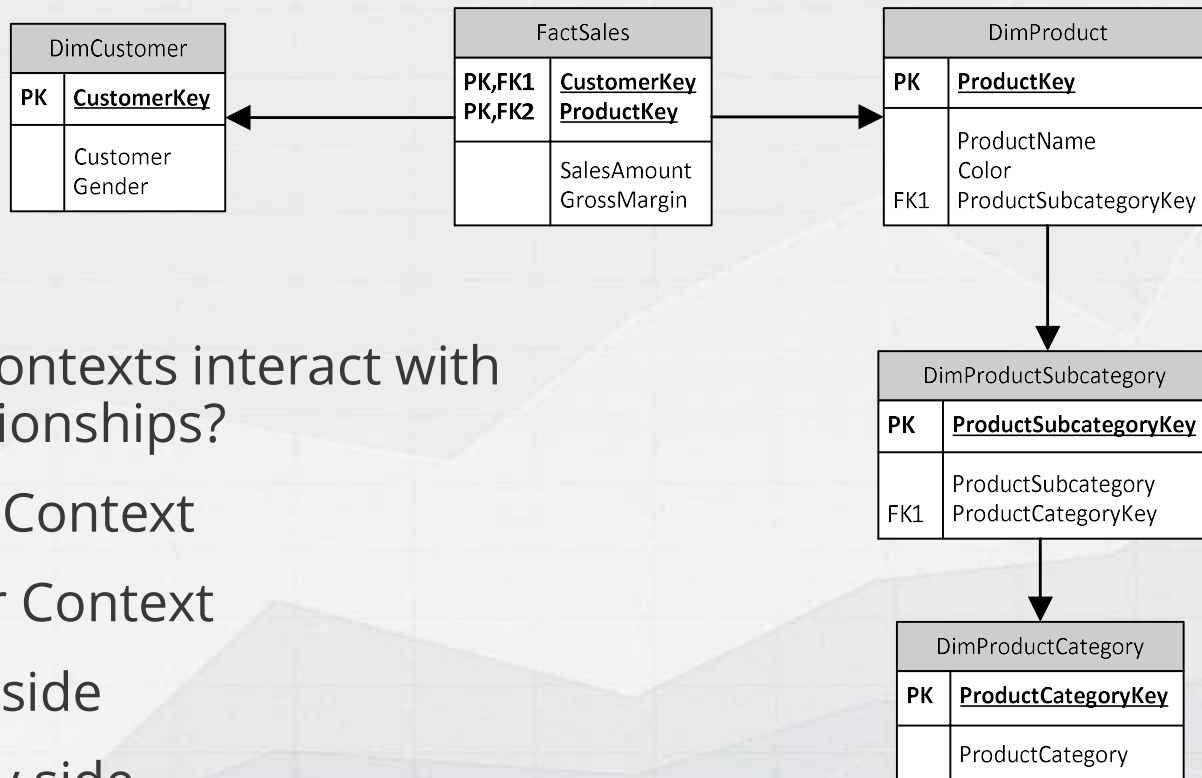
# CALCULATE Tasks

- Partially replaces the filter context
- Conditions
  - Can replace a whole table
  - Can replace a single column
- CALCULATE works on the filter context

# Evaluation Contexts And Relationships



# Filters and Relationships



- Do contexts interact with relationships?
- Row Context
- Filter Context
- One side
- Many side

# Row Context – Many Tables



- Row Context
  - Does not propagate over relationships

fx =Orders[Amount] \* (1 - Channels[Discount])

Channel	Color	Size	Quantity	Price	Amount	DiscountedAmount
Store	Red	Large	1	15	15	#ERROR
Store	Red	Small	2	13	26	#ERROR
Store	Green					
Store	Green					
Internet	Red					
Internet	Red					
Internet	Green					
Internet	Green	Small	120	1	120	#ERROR

PowerPivot for Excel

The value for column 'Discount' in table 'Channels' cannot be determined in the current context.

OK

# RELATED

- RELATED ( table[column] )
  - Opens a new row context on the target table
  - Following relationships

[DiscountedAmo ▾]		fx =Orders[Amount] * (1 - RELATED(Channels[Discount]))						
City ▾	Cha... ▾	Color ▾	Size ▾	Quantity ▾	Price ▾	Amount ▾	DiscountedAmount ▾	
Paris	Store	Red	Large	1	15	15	14.25	
Paris	Store	Red	Small	2	13	26	24.7	
Torino	Store	Green	Large	4	11	44	41.8	
New York	Store	Green	Small	8	9	72	68.4	
	Internet	Red	Large	16	7	112	100.8	
	Internet	Red	Small	32	5	160	144	
	Internet	Green	Large	64	3	192	172.8	
	Internet	Green	Small	128	1	128	115.2	



# RELATEDTABLE

- RELATEDTABLE ( table )
  - Filters the parameter table
  - Returns only rows related with the current one
- It is the companion of RELATED

[OrdersCount]		<i>fx</i>	=COUNTROWS( RELATEDTABLE( Orders ) )	
Cha...	Discount	OrdersCount	Add Column	
Internet	0.1	4		
Store	0.05	4		
Orders	Channels			

# Filter Context – Many Tables

- Filter Context and Relationships
  - Relationships affect filter context
- Orders → Products
  - Filter context on Orders
  - Follows the relationship
  - Shows only products of the selected orders
- Different from Row Context

# Filter Context – Many Tables

The screenshot illustrates filter context in a multi-table environment. A PivotTable is configured with 'Continent' and 'Channel' as filters, and 'Sum of DiscountedAmount' as the value field. The PivotTable shows a grand total of 68.4 for the 'Small' size category. A red arrow points from the 'North America' filter selection in the PivotTable to the 'New York' row in the table below. Another red arrow points from the 'Store' filter selection to the 'New York' row. A blue arrow points from the 'Grand Total' cell in the PivotTable to the 'DiscountedAmount' cell in the 'New York' row of the table.

Continent	Channel	Country	City	Color	Size	Quantity	Price	Amount	DiscountedAmount
Europe	Store	France	Paris	Red	Large	1	15	15	14.25
Europe	Store	France	Paris	Red	Small	2	13	26	24.7
Europe	Store	Italy	Torino	Green	Large	4	11	44	41.8
North America	Store	USA	New York	Green	Small	8	9	72	68.4
Europe	Internet			Red	Large	16	7	112	100.8
Europe	Internet			Red	Small	32	5	160	144
Europe	Internet			Green	Large	64	3	192	172.8
Europe	Internet			Green	Small	128	1	128	115.2

PowerPivot Field List

Choose fields to add to report:

Search

Orders

- ☐ City
- ☐ Channel
- ☒ Color
- ☒ Size
- ☐ Quantity
- ☐ Price
- ☐ Amount
- ☒ DiscountedAmount
- ☐ CalcAmount

Channels

- ☒ Channel
- ☐ Discount

Cities

- ☐ City
- ☐ Country
- ☒ Continent

# Filter Context – Many Tables

The screenshot displays an Excel worksheet with a PivotTable and a PivotChart. The PivotTable is filtered by Continent (Europe) and Channel (Store). The PivotChart is a bar chart showing the sum of discounted amount for each color and size. The PivotTable data is as follows:

Row Labels	Large	Small	Grand Total
Green	41.8		41.8
Red	14.25	24.7	38.95
<b>Grand Total</b>	<b>56.05</b>	<b>24.7</b>	<b>80.75</b>

The PivotChart data is as follows:

Color	Size	Sum of Discounted Amount
Green	Large	41.8
Green	Small	0
Red	Large	14.25
Red	Small	24.7

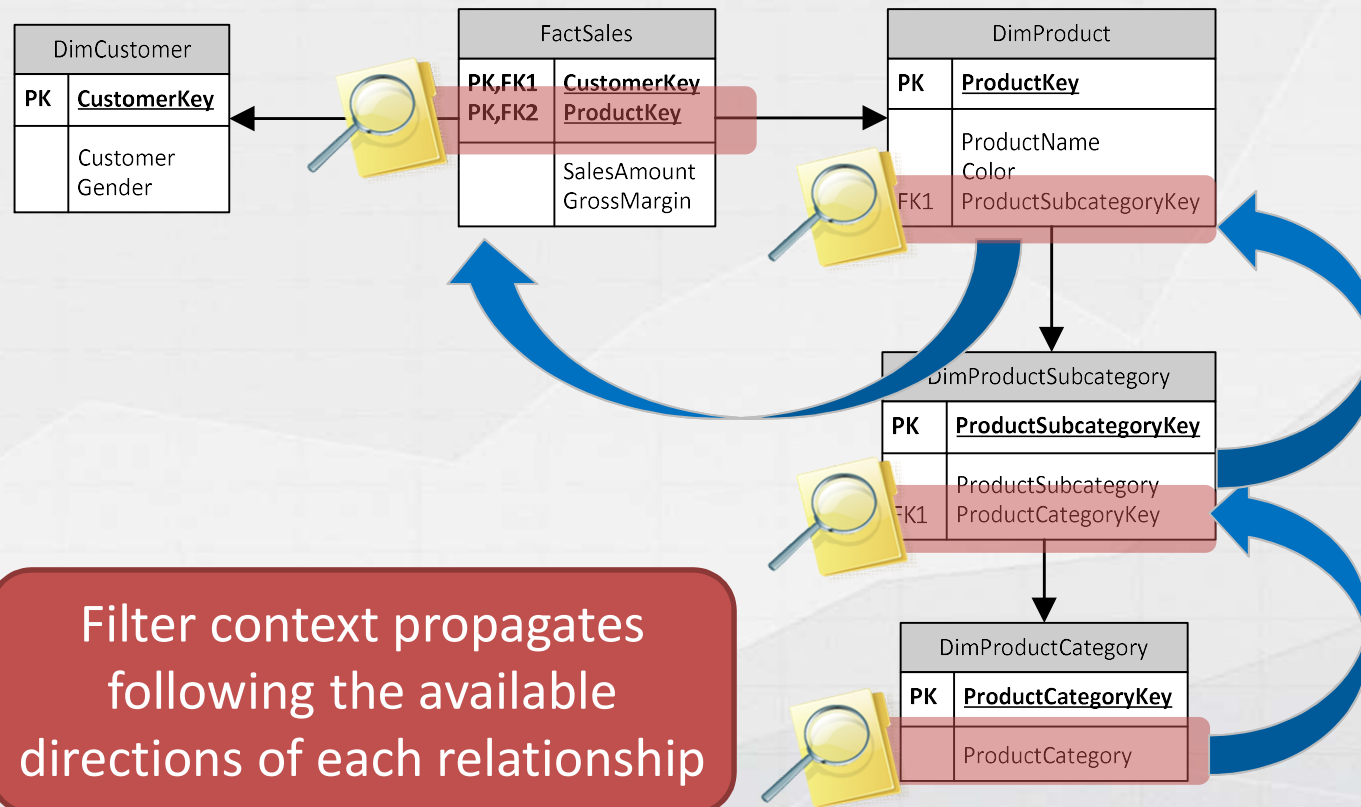
The PivotTable is filtered by Continent (Europe) and Channel (Store). The PivotChart is a bar chart showing the sum of discounted amount for each color and size. The PivotTable data is as follows:

Row Labels	Large	Small	Grand Total
Green	41.8		41.8
Red	14.25	24.7	38.95
<b>Grand Total</b>	<b>56.05</b>	<b>24.7</b>	<b>80.75</b>

The PivotChart data is as follows:

Color	Size	Sum of Discounted Amount
Green	Large	41.8
Green	Small	0
Red	Large	14.25
Red	Small	24.7

# Filters and Relationships



# Context Transition



- Calculate performs another task
- If executed inside a row context
  - It takes the row context
  - Transforms it into an equivalent filter context
  - Applies it to the data model
  - Before computing its expression
- Very important and useful feature
  - Better to learn it writing some code...

# Automatic CALCULATE

Whenever a measure is computed, an automatic CALCULATE is added around the measure

The diagram illustrates the automatic addition of a CALCULATE function in DAX. It shows two versions of a measure. The top version is a SUMX function with two arguments: 'Orders' and '[Sales Amount]'. A horizontal blue line is drawn under the second argument, '[Sales Amount]'. A blue curved arrow originates from this line and points down to the second argument of a second SUMX function below. The second SUMX function has the same first argument 'Orders', but its second argument is wrapped in a red 'CALCULATE' function: 'CALCULATE ( [Sales Amount] )'.

```
SUMX (  
    Orders,  
    [Sales Amount]  
)
```

```
SUMX (  
    Orders,  
    CALCULATE ( [Sales Amount] )  
)
```

Time intelligence functions

# Time Intelligence in DAX





# Date Table

- Time intelligence needs a date table
  - Built in DAX
  - Or in a SQL Table
- Date table properties
  - All dates should be present
  - From 1° of January, to 31° of December
  - No holes
  - Otherwise time intelligence will not work

# CALENDAR



Returns a table with a single column named "Date" containing a contiguous set of dates in the given range, inclusive.

```
CALENDAR (  
    DATE ( 2005, 1, 1 ),  
    DATE ( 2015, 12, 31 )  
)
```

```
CALENDAR (  
    MIN ( Sales[Order Date] ),  
    MAX ( Sales[Order Date] )  
)
```

# CALENDARAUTO



Automatically creates a calendar table based on the database content. Optionally you can specify the last month (useful for fiscal years)

```
--  
-- The parameter is the last month  
-- of the fiscal year  
--  
= CALENDARAUTO (  
    6  
)
```

Beware: CALENDARAUTO uses  
all the dates in your model,  
excluding only calculated  
columns and tables

# Set Sorting Options

- Month names do not sort alphabetically
  - April is not the first month of the year
- Use Sort By Column
- Set all sorting options in the proper way
- Beware of sorting granularity
  - 1:1 between names and sort keys

# What is Time Intelligence?

- Many different topics in one name
  - Year To Date
  - Quarter To Date
  - Running Total
  - Same period previous year
  - Working days computation
  - Fiscal Year
- In short: anything related with time
  - Handled in an intelligent way

# Aggregations Over Time

- Many useful aggregations
  - YTD: Year To Date
  - QTD: Quarter To Date
  - MTD: Month To Date
- They all need a Calendar Table
- And some understanding of CALCULATE

# Year To Date (Time Intelligence)

DATESYTD makes filtering much easier

SalesAmountYTD :=

```
CALCULATE (  
    SUM ( Sales[SalesAmount] ),  
    DATESYTD ( 'Date'[Date] )  
)
```

# Handling Fiscal Year

The last, optional, parameter is the end of the fiscal year  
Default: 12-31 (or 31/12 - locale dependent)

```
SalesAmountYTD :=  
TOTALYTD (  
    SUM ( Sales[SalesAmount] ),  
    'Date'[Date],  
    "06-30"  
)
```

```
SalesAmountYTD :=  
CALCULATE (  
    SUM ( Sales[SalesAmount] ),  
    DATESYTD ( 'Date'[Date], "06-30" )  
)
```



# Same Period Last Year

Same period in previous year. CALCULATE is needed

Specialized version of DATEADD

```
Sales_SPLY :=
```

```
CALCULATE (
    SUM ( Sales[SalesAmount] ),
    SAMEPERIODLASTYEAR ( 'Date'[Date] )
)
```

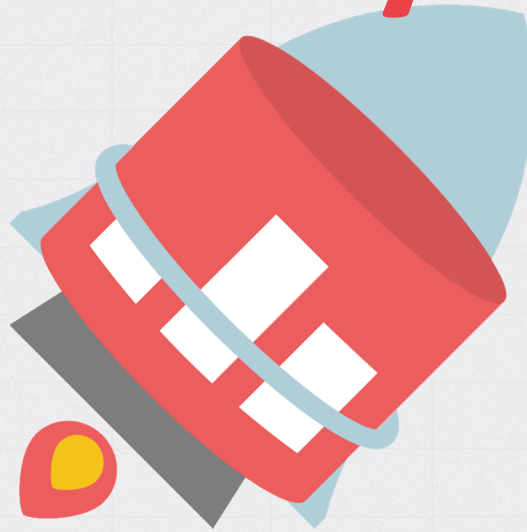
# Running Total

Running total requires an explicit filter

SalesAmountRT :=

```
CALCULATE (
    SUM ( Sales[SalesAmount] ),
    FILTER (
        ALL ( 'Date' ),
        'Date'[Date] <= MAX ( 'Date'[Date] )
    )
)
```

# Thank you!



Check our articles, whitepapers and courses on  
[www.sqlbi.com](http://www.sqlbi.com)