Strategic Sponsor

Microsoft

Gold Sponsors

PYRAMID ANALYTICS

KRAJOWY REJESTR DŁUGÓW www.krd.pl

devart Database Interaction Solutions

Silver Sponsors

Profisee®

SQL EXPERT.pl

CREDIT SUISSE

Technical Partners

Microsoft

PTI POLSKIE TOWARZYSTWO INFORMATYCZNE ODDZIAŁ GÓRNOŚLĄSKI

Academic Partners

Wyższa Szkoła Zarządzania i Bankowości w Krakowie

Wyższe Szkoły Bankowe

Media Partners

SPECJALISTA IT | PROGRAMISTA | DESIGNER GEEKCLUB.pl wss.pl

ERP24.pl

4programmers.net

ITprofessional

IN FACT PR

itnews24

Vir virtual-it.pl

IT news POLSKA

VirtualStudy

MAGAZYN IT PL

Made in Wro Do IT here!

ERP-view.pl Portal rozwiązań IT w biznesie

Advanced Time Intelligence in DAX

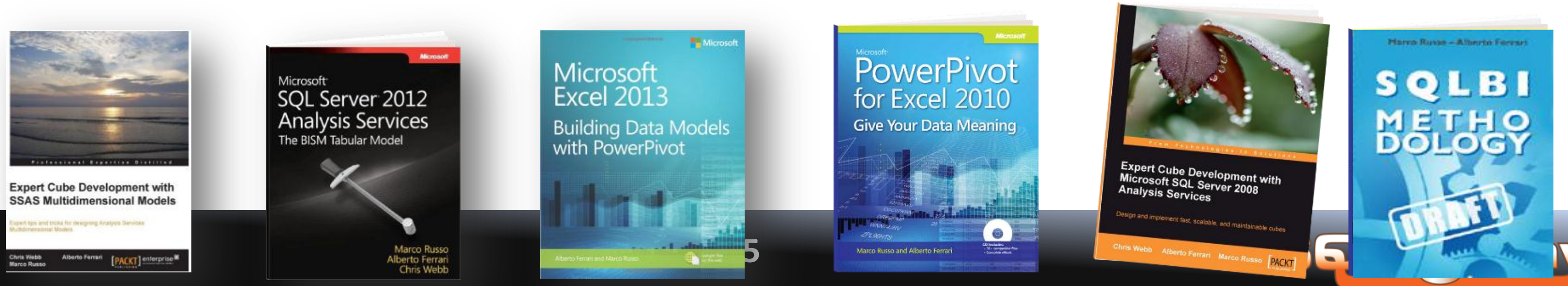
Alberto Ferrari

SQLBI

www.sqlbi.com

Alberto Ferrari

- BI Expert and Consultant
- Founder of www.sqlbi.com
  - Problem Solving
  - Complex Project Assistance
  - Data Warehouse Assessments and Development
  - Courses, Trainings and Workshops
- Book Writer
- Microsoft Gold Business Intelligence Partner
- SSAS Maestro – MVP – MCP


Expert Cube Development with SSAS Multidimensional Models
Chris Webb Marco Russo Alberto Ferrari
PACKT enterprise
Microsoft SQL Server 2012 Analysis Services
The BISM Tabular Model
Marco Russo Alberto Ferrari Chris Webb
Microsoft Excel 2013
Building Data Models with PowerPivot
Alberto Ferrari and Marco Russo
Microsoft PowerPivot for Excel 2010
Give Your Data Meaning
Marco Russo and Alberto Ferrari
Expert Cube Development with Microsoft SQL Server 2008 Analysis Services
Design and implement fast, scalable, and maintainable cubes
Chris Webb Alberto Ferrari Marco Russo
PACKT
Marco Russo – Alberto Ferrari
SQLBI METHODOLOGY
DRAFT

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

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

Create a Date Table

- Stored in a SQL table inside the DWH
- Optional
  - Create a holidays table
  - Useful for working days calculations
- Set all the necessary attributes
  - Do not create them in DAX
  - SQL is the best place for all attributes
- Building a view is always the best option

CALENDAR

2015

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[OrderDate] ),
    MAX ( Sales[OrderDate] )
)
```

CALENDAR

If you have multiple fact tables, you need to compute the correct values

```
=CALENDAR (
       MIN (
              MIN ( Sales[OrderDate] ),
              MIN ( Purchases[PurchaseDate] )
       ),
       MAX (
              MAX ( Sales[OrderDate] ),
              MAX ( Purchases[PurchaseDate] )
       )
)
```

CALENDARAUTO

2015

Automatically creates a calendar table, including the dates including the fiscal years (both start and end)

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

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

Mark as Date Table

- Need to mark the calendar as date table
- Set the column containing the date
- Needed to make time intelligence works
- Used by client tools as metadata information
  - Power View
  - Q&A
  - Excel
- Multiple tables can be marked as date table

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

Handling DateTime

- If time is a useful information
- Separate Date from Time
  - Date part → Calendar Table
  - Time part → Time Table
- Reduces distinct values
- Makes analysis much easier
- Time can often be normalized in time ranges of 5, 10 or 30 minutes

Multiple Dates

- Date is often a role dimension
  - Many roles for a date
  - Many date tables
- How many date tables?
  - Try to use only one table
  - Use many, only if needed by the model
  - Many date tables lead to confusion
    - And issues when slicing
- Use proper naming convention

Counting Working Days

- How many working days in a date range?
  - Easily solved with Calendar table
  - Define a new column «WorkingDays»
  - Aggregate with SUM
- Handles any date range
- Works on periods with «holes»
- No separation between fact tables and dimensions in Tabular

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: the easy way

TOTALYTD: the “DAX for dummies” version

```
SalesAmountYTD :=

TOTALYTD (
    SUM ( Sales[SalesAmount] ),
    '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"
)
```

Year To Date: the DAX way

- DATESYTD: Returns a set of dates, from the start of the year up to the parameter date

- CALCULATE: Creates a filter context and performs the SUM operation

```
SalesAmountYTD :=

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

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] )
)
```

Mixing Time Intelligence Functions

YTD on the previous year. In DAX, it is very simple, just mix the functions to obtain the result

```
Sales_YTDLY :=

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

DATEADD

Similar as SAMEPERIODLASTYEAR, used to calculate different periods: YEAR, MONTH, DAY …

Does not sum dates, it shifts periods over time

```
Sales_SPLY :=

CALCULATE (
    SUM( Sales[SalesAmount] ),
    DATEADD ( 'Date'[Date] , -1, YEAR )
)
```

PARALLELPERIOD

Returns a set of dates (a table) shifted in time

The whole period is returned, regardless dates in the first parameter

```
Sales_PPLY :=

CALCULATE (
    SUM ( Sales[SalesAmount] ),
    PARALLELPERIOD ( 'Date'[Date] , -1, YEAR )
)
```

Running Total

Running total, as most of the more complex time intelligence aggregations, needs the CALCULATE version, because there is no syntax sugar here

```
SalesAmountRT :=

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

Moving Annual Total

Moving window from the current date back one year

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

There are many week scenarios, depending on what you mean by «week»… CALCULATE is your best friend here

CALCULATIONS OVER WEEKS

Custom Calendars

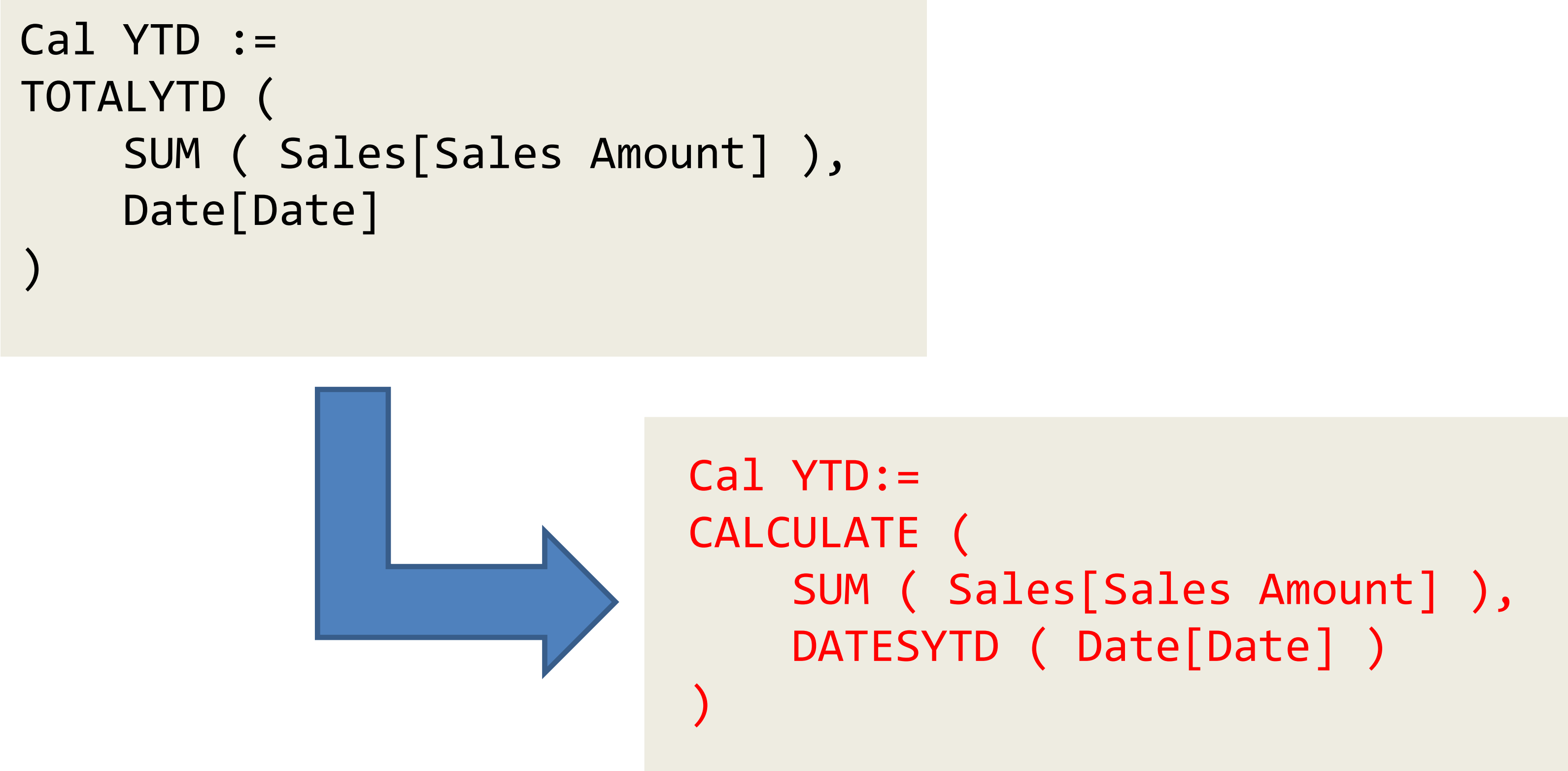
- Time Intelligence functions
  - One day belong to the same quarter every day
  - Not true if you use week calculations
- 4-4-5, 4-5-4, 5-4-4 ?????
  - One quarter is made of three months
  - Two of 4 weeks
  - One of 5 weeks
  - Difference only in position of the 5 weeks month
- No support in DAX for these calendars

Create Week Numbers

- Usually stored in the database
- If not, use Excel to compute them
- Or, as a better solution, use Power Query
  - M code is included in the attendee's pack!
  - Just copy and paste in Power Query

Time intelligence under the Hood

TOTALYTD is internally transformed into a CALCULATE statement



```
Cal YTD :=
TOTALYTD (
    SUM ( Sales[Sales Amount] ),
    Date[Date]
)
```

```
Cal YTD:=
CALCULATE (
    SUM ( Sales[Sales Amount] ),
    DATESYTD ( Date[Date] )
)
```

YTD without DATESYTD

You can compute YTD even if DATESYTD was not available, it is harder but it works the very same way

```
Cal YTD: :=
IF (
    HASONEVALUE ( Date[Year] ),
    CALCULATE (
        SUM ( Sales[Sales Amount] ),
        FILTER (
            ALL ( Date ),
            Date[Date] <= MAX ( Date[Date] )
        ),
        VALUES ( Date[Year] )
    )
)
```

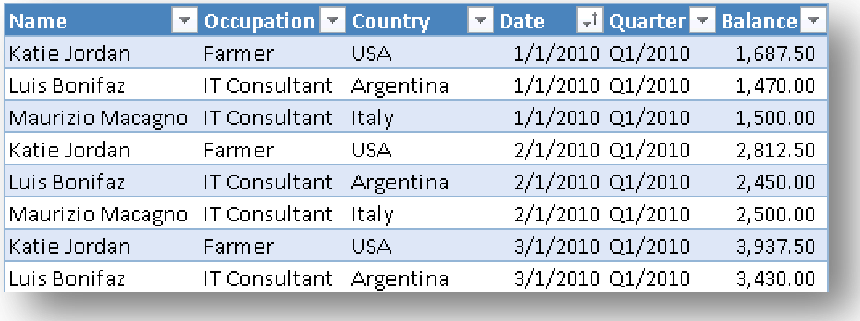
Week Calculations

- Fine, so how to work with weeks?
- Easier to look at a demo ☺

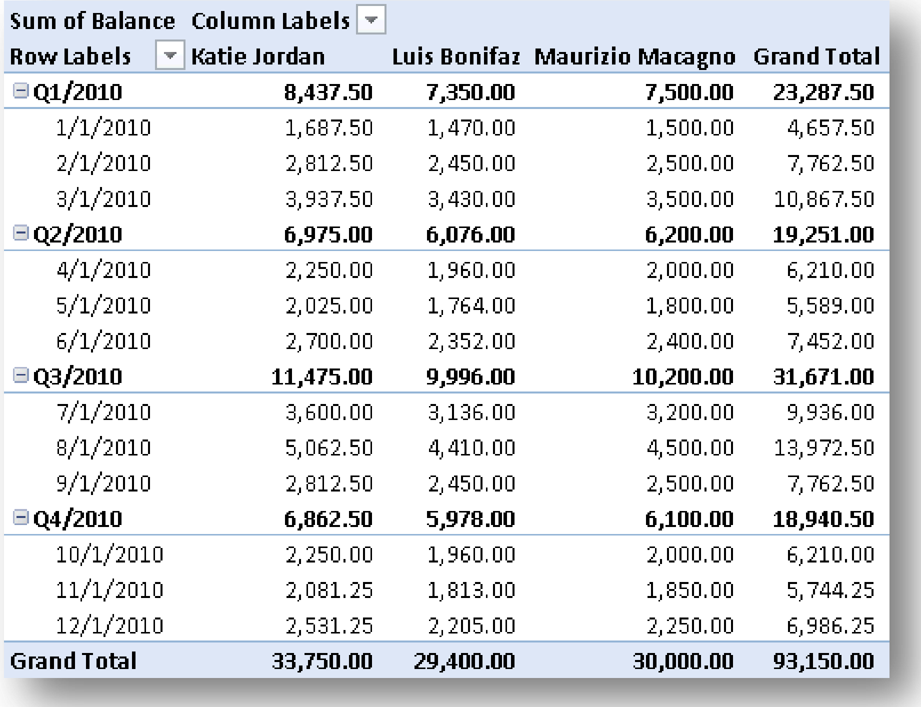
Semi Additive Measures

- Additive Measure
  - SUM over all dimensions
- Semi Additive Measure
  - SUM over some dimensions
  - Different function over other dimensions
  - Time is the standard exception for aggregations
  - Examples
    - Warehouse stocking
    - Current account balance

Current Account Balance



Name ▾	Occupation ▾	Country ▾	Date ▾	Quarter ▾	Balance ▾
Katie Jordan	Farmer	USA	1/1/2010	Q1/2010	1,687.50
Luis Bonifaz	IT Consultant	Argentina	1/1/2010	Q1/2010	1,470.00
Maurizio Macagno	IT Consultant	Italy	1/1/2010	Q1/2010	1,500.00
Katie Jordan	Farmer	USA	2/1/2010	Q1/2010	2,812.50
Luis Bonifaz	IT Consultant	Argentina	2/1/2010	Q1/2010	2,450.00
Maurizio Macagno	IT Consultant	Italy	2/1/2010	Q1/2010	2,500.00
Katie Jordan	Farmer	USA	3/1/2010	Q1/2010	3,937.50
Luis Bonifaz	IT Consultant	Argentina	3/1/2010	Q1/2010	3,430.00

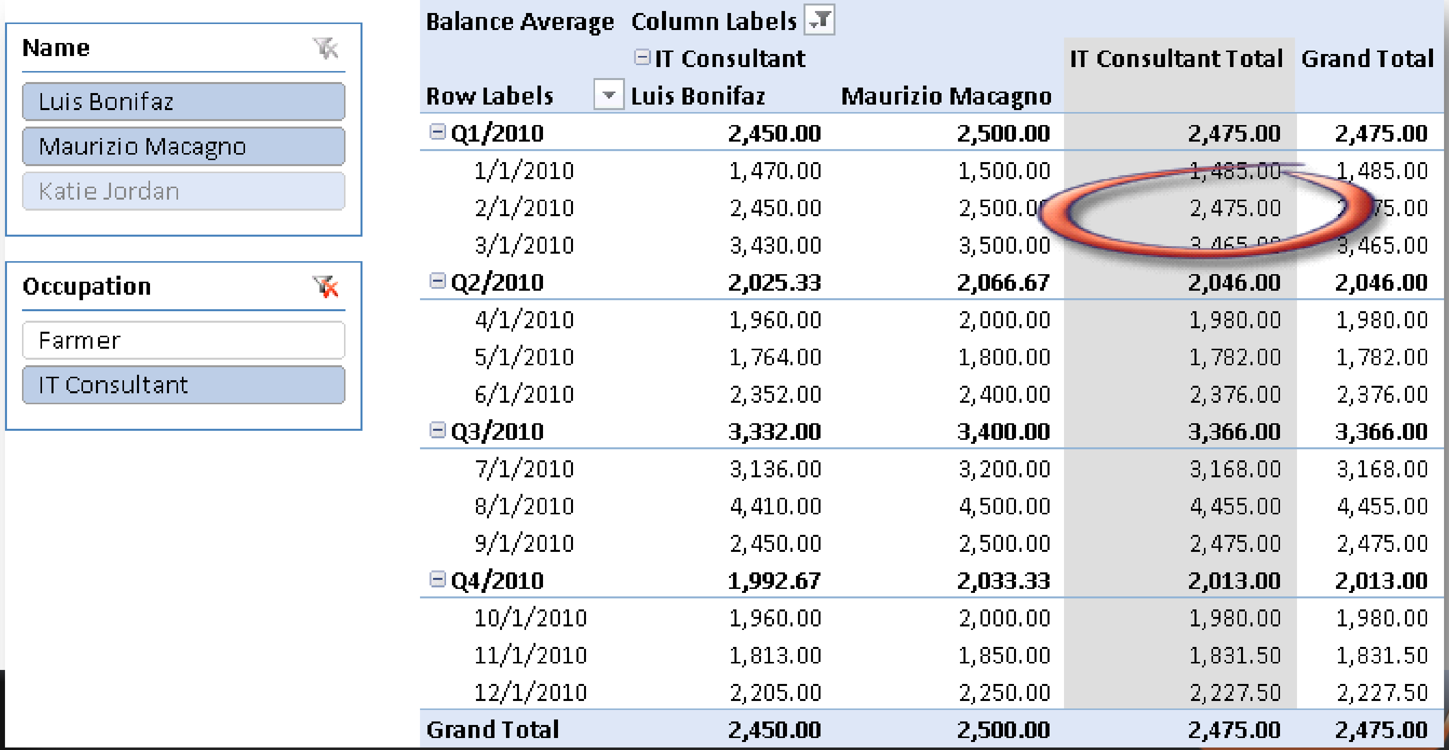


Sum of Balance	Column Labels ▾			
Row Labels ▾	Katie Jordan	Luis Bonifaz	Maurizio Macagno	Grand Total
⊟Q1/2010	**8,437.50**	**7,350.00**	**7,500.00**	**23,287.50**
1/1/2010	1,687.50	1,470.00	1,500.00	4,657.50
2/1/2010	2,812.50	2,450.00	2,500.00	7,762.50
3/1/2010	3,937.50	3,430.00	3,500.00	10,867.50
⊟Q2/2010	**6,975.00**	**6,076.00**	**6,200.00**	**19,251.00**
4/1/2010	2,250.00	1,960.00	2,000.00	6,210.00
5/1/2010	2,025.00	1,764.00	1,800.00	5,589.00
6/1/2010	2,700.00	2,352.00	2,400.00	7,452.00
⊟Q3/2010	**11,475.00**	**9,996.00**	**10,200.00**	**31,671.00**
7/1/2010	3,600.00	3,136.00	3,200.00	9,936.00
8/1/2010	5,062.50	4,410.00	4,500.00	13,972.50
9/1/2010	2,812.50	2,450.00	2,500.00	7,762.50
⊟Q4/2010	**6,862.50**	**5,978.00**	**6,100.00**	**18,940.50**
10/1/2010	2,250.00	1,960.00	2,000.00	6,210.00
11/1/2010	2,081.25	1,813.00	1,850.00	5,744.25
12/1/2010	2,531.25	2,205.00	2,250.00	6,986.25
Grand Total	**33,750.00**	**29,400.00**	**30,000.00**	**93,150.00**

- Month level correct
- Quarter level wrong
- Year level wrong

Semi additive measures

- Aggregation depends on the filter
  - LastChild over time
  - SUM for the other dimensions



Name

- Luis Bonifaz
- Maurizio Macagno
- Katie Jordan

Occupation

- Farmer
- IT Consultant

Balance Average	Column Labels			
	IT Consultant		IT Consultant Total	Grand Total
Row Labels	Luis Bonifaz	Maurizio Macagno		
Q1/2010	**2,450.00**	**2,500.00**	**2,475.00**	**2,475.00**
1/1/2010	1,470.00	1,500.00	1,485.00	1,485.00
2/1/2010	2,450.00	2,500.00	2,475.00	2,475.00
3/1/2010	3,430.00	3,500.00	3,465.00	3,465.00
Q2/2010	**2,025.33**	**2,066.67**	**2,046.00**	**2,046.00**
4/1/2010	1,960.00	2,000.00	1,980.00	1,980.00
5/1/2010	1,764.00	1,800.00	1,782.00	1,782.00
6/1/2010	2,352.00	2,400.00	2,376.00	2,376.00
Q3/2010	**3,332.00**	**3,400.00**	**3,366.00**	**3,366.00**
7/1/2010	3,136.00	3,200.00	3,168.00	3,168.00
8/1/2010	4,410.00	4,500.00	4,455.00	4,455.00
9/1/2010	2,450.00	2,500.00	2,475.00	2,475.00
Q4/2010	**1,992.67**	**2,033.33**	**2,013.00**	**2,013.00**
10/1/2010	1,960.00	2,000.00	1,980.00	1,980.00
11/1/2010	1,813.00	1,850.00	1,831.50	1,831.50
12/1/2010	2,205.00	2,250.00	2,227.50	2,227.50
Grand Total	**2,450.00**	**2,500.00**	**2,475.00**	**2,475.00**

SemiAdditive Measures

CALCULATE: to set the filter
LASTDATE: to find the last child

```
LastBalance :=

CALCULATE (
    SUM ( Balances[Balance] ),
    LASTDATE ( Date[Date] )
)
```

Last Non Blank

Searches in the fact table for the last non empty date.
Remember: LASTNONBLANK is an iterator

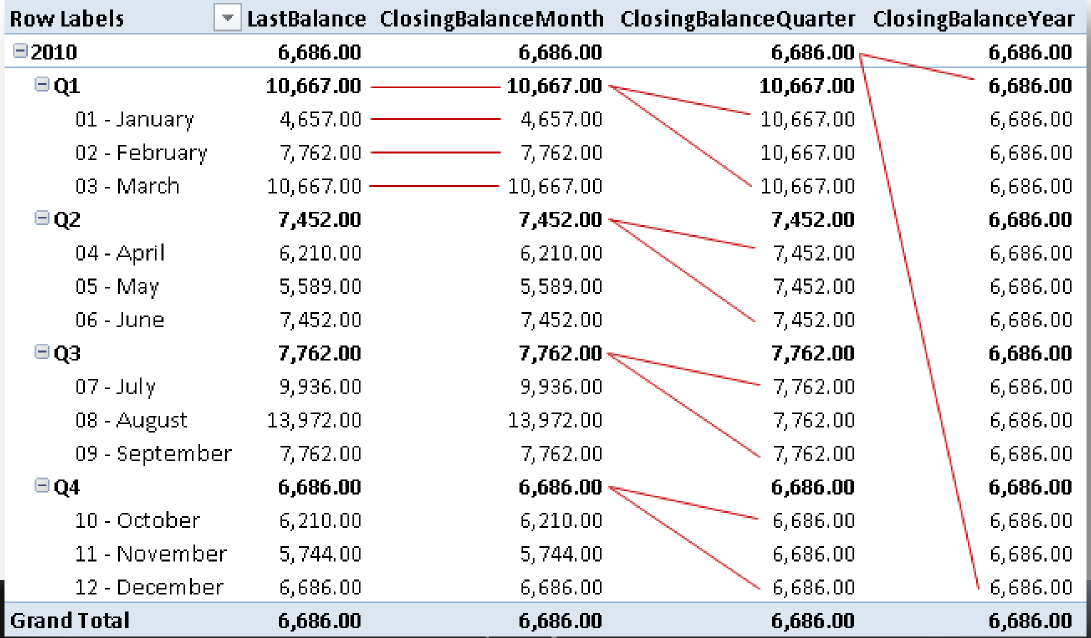
```
LastBalanceNonBlank :=

CALCULATE (
    SUM ( Balances[Balance] ),
    LASTNONBLANK (
        BalanceDate[Date],
        CALCULATE ( COUNTROWS ( Balances ) )
    )
)
```

Opening and Closing Balance

ClosingBalanceMonth =
CLOSINGBALANCEMONTH (SUM (Balances[Balance]), BalanceDate[Date])

ClosingBalanceQuarter =
CLOSINGBALANCEQUARTER (SUM (Balances[Balance]), BalanceDate[Date])

ClosingBalanceYear =
CLOSINGBALANCEYEAR (SUM (Balances[Balance]), BalanceDate[Date])



Row Labels	LastBalance	ClosingBalanceMonth	ClosingBalanceQuarter	ClosingBalanceYear
⊟2010	**6,686.00**	**6,686.00**	**6,686.00**	**6,686.00**
⊟Q1	**10,667.00**	**10,667.00**	**10,667.00**	**6,686.00**
01 - January	4,657.00	4,657.00	10,667.00	6,686.00
02 - February	7,762.00	7,762.00	10,667.00	6,686.00
03 - March	10,667.00	10,667.00	10,667.00	6,686.00
⊟Q2	**7,452.00**	**7,452.00**	**7,452.00**	**6,686.00**
04 - April	6,210.00	6,210.00	7,452.00	6,686.00
05 - May	5,589.00	5,589.00	7,452.00	6,686.00
06 - June	7,452.00	7,452.00	7,452.00	6,686.00
⊟Q3	**7,762.00**	**7,762.00**	**7,762.00**	**6,686.00**
07 - July	9,936.00	9,936.00	7,762.00	6,686.00
08 - August	13,972.00	13,972.00	7,762.00	6,686.00
09 - September	7,762.00	7,762.00	7,762.00	6,686.00
⊟Q4	**6,686.00**	**6,686.00**	**6,686.00**	**6,686.00**
10 - October	6,210.00	6,210.00	6,686.00	6,686.00
11 - November	5,744.00	5,744.00	6,686.00	6,686.00
12 - December	6,686.00	6,686.00	6,686.00	6,686.00
Grand Total	**6,686.00**	**6,686.00**	**6,686.00**	**6,686.00**

Opening and Closing Balance

DAX Function	Equivalent date filter in CALCULATE
OPENINGBALANCEMONTH	STARTOFMONTH
OPENINGBALANCEQUARTER	STARTOFQUARTER
OPENINGBALANCEYEAR	STARTOFYEAR
CLOSINGBALANCEMONTH	ENDOFMONTH
CLOSINGBALANCEQUARTER	ENDOFQUARTER
CLOSINGBALANCEYEAR	ENDOFYEAR

Time Intelligence: Conclusions

- Based on evaluation contexts
  - Replace filter on date
  - Many predefined functions
  - You can author your own functions
- Basic Time Itelligence
- Creating more complex aggregations
- Working with ISO weeks

Strategic Sponsor

Microsoft

Gold Sponsors

PYRAMID ANALYTICS

KRAJOWY REJESTR DŁUGÓW www.krd.pl

devart Database Interaction Solutions

Silver Sponsors

Profisee®

SQL EXPERT.pl

CREDIT SUISSE

Technical Partners

Microsoft

PTI POLSKIE TOWARZYSTWO INFORMATYCZNE ODDZIAŁ GÓRNOŚLĄSKI

Academic Partners

Wyższa Szkoła Zarządzania i Bankowości w Krakowie

Wyższe Szkoły Bankowe

Media Partners

SPECJALISTA IT PROGRAMISTA DESIGNER GEEKCLUB.pl wss.pl

ERP24.pl

4programmers.net

ITprofessional

IN FACT PR

itnews24

virtual-it.pl

IT news POLSKA

VirtualStudy

MAGAZYN IT PL

Made in Wro Do IT here!

ERP-view.pl Portal rozwiązań IT w biznesie