# Basic Select statements

select

from [DWCube]

-- This is the number of Units In

select

from [DWCube]

where [Measures].[Units In]

-- Explicitly giving the measure

Note the lack of column headings and row headings. This clause, which is also known as the slicer axis, on which you specify a set, tuple, or member that restricts the members returned on the column and row axis by a query. The WHERE clause changes the default member

select

from [DWCube]

where [Measures].[Units Balance]

-- A different measure

select Measures.[Units Balance]

from [DWCube]

-- This doesn't work

Error (Data mining): Either the user does not have permission to access the referenced mining model, 'DWCube', or the object does not exist.

select Measures.[Units Balance] on 0

from [DWCube]

-- This does work. This is the shorthand for axis(0)

select Measures.[Units Balance] on columns

from [DWCube]

-- As does this

select Measures.[Units Balance] on columns

from [DWCube]

select Measures.[Units Balance] on 0

from [DWCube]

-- This doesn't work

Parser: The syntax for 'select' is incorrect.

select Measures.[Units Balance] on columns

from [DWCube];

select Measures.[Units Balance] on 0

from [DWCube]

-- This doesn't work either

Multiple statements are not allowed without a GO in the middle.

select Measures.[Units Balance], [Measures].[Units In] on 0

from [DWCube]

-- This doesn't work

select {Measures.[Units Balance], [Measures].[Units In]} on 0

from [DWCube]

-- This does work, as you have created a tuple

A tuple uniquely identifies a slice of data from a cube. The tuple is formed by a combination of dimension members, as long as there are no two or more members that belong to the same hierarchy.

select {Measures.[Units Balance], [Measures].[Units In]} on 1

from [DWCube]

-- This doesn't work

Axis numbers specified in a query must be sequentially specified, and cannot contain gaps.

select {} on 0,

{Measures.[Units Balance], [Measures].[Units In]} on 1

from [DWCube]

-- This does work, but it doesn't show much.

with MEMBER [Units Daily Balance] as

[Measures].[Units In]-[Measures].[Units Out]

select [Units Daily Balance] on 0,

from [DWCube]

create MEMBER [DWCube].[Measures].[Units Daily Balance] as

'[Measures].[Units In]-[Measures].[Units Out]'

go

select [Measures].[Units Daily Balance] on 0,

from [DWCube]

drop member [DWCube].[Measures].[Units Daily Balance]

go

# Dimensions

select [Dim Product] on 0

from [DWCube]

where Measures.[Units Balance]

-- This contains the dimension, but doesn't work.

-- We'll find out why soon.

The 'Dim Product' dimension contains more than one hierarchy, therefore the hierarchy must be explicitly specified.

select [Dim Product].[(All)] on 0

from [DWCube]

where Measures.[Units Balance]

-- This returns the member "All" from this dimension.

select [Dim Product].[Color] on 0

from [DWCube]

where Measures.[Units Balance]

-- This does work, only again shows the All member.

-- This shows the Color hierarchy within the Dim Product dimension

select [Dim Product].Color.[(All)] on 0

from [DWCube]

where Measures.[Units Balance]

-- This returns the member "All" from the Color

-- hierarchy from the Dim Product dimension.

select [Dim Product].Color.Color on 0

from [DWCube]

where Measures.[Units Balance]

-- Expand the Color hierarchy to show the Color members

-- This is the Dim Product dimension

-- The Color hierarchy, the color members

select [Dim Product].Color.members on 0

from [DWCube]

where Measures.[Units Balance]

-- This shows all of the members, including All.

select [Dim Product].[Color].Color.&[Black] on 0

from [DWCube]

where Measures.[Units Balance]

-- This retrieves Black only.

select [Dim Product].[Color].&[Black] on 0

from [DWCube]

where Measures.[Units Balance]

-- This also retrieves Black only

-- as there is no ambiguity

select [Dim Date].[Calendar Year].members on 0

from [DWCube]

where Measures.[Units Balance]

-- What does this do?

select [Dim Date].[Calendar Year].[Calendar Year] on 0

from [DWCube]

where Measures.[Units Balance]

-- What does this do?

select [Dim Date].[Calendar Year].[Calendar Year].[2010] on 0

from [DWCube]

where Measures.[Units Balance]

-- What does this do?

select [Dim Date].[Calendar Year].[Calendar Year].[2010].[December] on 0

from [DWCube]

where Measures.[Units Balance]

--- This doesn't work. December is not part of the Calendar Year hierarchy.

select [Dim Date].[Hierarchy].[Month Number Of Year].&[12]&[2010] on 0

from [DWCube]

where Measures.[Units Balance]

--- This gives the December 2010 figure.

select [Dim Date].[Hierarchy].[Date Key].&[20101231] on 0

from [DWCube]

where Measures.[Units Balance]

--- This gives the 31 December 2010 figure.

select [Dim Date].[Calendar Year].[Calendar Year].[2010]

, [Dim Date].[Calendar Year].[Calendar Year].[2012]

on 0

from [DWCube]

where Measures.[Units Balance]

-- This doesn't work

The statement dialect could not be resolved due to ambiguity.

select {[Dim Date].[Calendar Year].[Calendar Year].[2010]

, [Dim Date].[Calendar Year].[Calendar Year].[2012]}

on 0

from [DWCube]

where Measures.[Units Balance]

-- Tuples also work for Dimensions

select [Dim Date].[Calendar Year].[Calendar Year].[2010]:

[Dim Date].[Calendar Year].[Calendar Year].[2012]

on 0

from [DWCube]

where Measures.[Units Balance]

-- This gives a range. The brackets are optional.

select null:

[Dim Date].[Calendar Year].[Calendar Year].[2012]

on 0

from [DWCube]

where Measures.[Units Balance]

-- This goes from the beginning

select [Dim Date].[Calendar Year].[Calendar Year].[2012]:null

on 0

from [DWCube]

where Measures.[Units Balance]

-- This goes to the end

select [Dim Product].[Color].&[Blue]:[Dim Product].[Color].&[Multi]

on 0

from [DWCube]

where Measures.[Units Balance]

-- This range function also works with other dimensions

select [Dim Product].[Color].&[Blue]:[Dim Product].[Color].&[Marron]

on 0

from [DWCube]

where Measures.[Units Balance]

-- But be careful if the member does not exist!

select {[Dim Date].[Calendar Year].[Calendar Year].members

, [Dim Date].[Calendar Year].[(All)]}

on 0

from [DWCube]

where Measures.[Units Balance]

-- TThis combines the years with All.

-- Note that All is now on the right-hand side.

select {[Dim Date].[Calendar Year].[Calendar Year].members}

on 0

from [DWCube]

where Measures.[Units Balance]

-- You can also put curly brackets around one member reference.

-- But there are many years which are Null.

select non empty [Dim Date].[Calendar Year].[Calendar Year].members

on 0

from [DWCube]

where Measures.[Units Balance]

-- Adding non empty removes columns or rows which are completely NULL.

select non empty [Dim Date].[Calendar Year].[Calendar Year].members

on 0

from [DWCube]

where Measures.[Units In]

go

with MEMBER [Units In Measure] as

distinctcount([Dim Date].[Calendar Year].[Calendar Year],[Measures].[Units In])

select non empty [Units In Measure]

on 0

from [DWCube]

-- Changing to Units In (to make the maths easier), do SUM, MAX, MIN, MEDIAN

select [Dim Date].[Calendar Year].[Calendar Year].members

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- You can add a second dimension on the Rows axis.

select [Dim Date].[Calendar Year].[Calendar Year].members

on 0,

[Dim Product].[Color].[Color].members on 1

,[Dim Date].[Calendar Quarter].[Calendar Quarter].members on 2

from [DWCube]

where Measures.[Units Balance]

-- You can't add a third dimension on the Pages axis.

Results cannot be displayed for cellsets with more than two axes.

select {[Dim Date].[Calendar Year].[Calendar Year].members

, [Dim Date].[Calendar Quarter].[Calendar Quarter].members}

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Tuples need to use the same hierarchies.

Query (1, 8) Members, tuples or sets must use the same hierarchies in the function.

select crossjoin([Dim Date].[Calendar Year].[Calendar Year].members

,[Dim Date].[Calendar Quarter].[Calendar Quarter].members)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Instead, you can use a cross-join

select [Dim Date].[Calendar Year].[Calendar Year].members

\* [Dim Date].[Calendar Quarter].[Calendar Quarter].members

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Because it multiples the number of columns, you can also use the \* sign.

select non empty [Dim Date].[Calendar Year].[Calendar Year].members

\* [Dim Date].[Calendar Quarter].[Calendar Quarter].members

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- and yes, you can use NON EMPTY as well. Note that 2010 Q1-3 have also disappeared.

select [Dim Date].[Hierarchy].[Month Number Of Year].&[12]&[2012]

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- What is the Units Balance in December 2012?

select [Dim Date].[Hierarchy].[Month Number Of Year].&[12]&[2012].members

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- We can't get a member of a member.

The MEMBERS function expects a level expression for the 1 argument. A member expression was used.

# Hierarchy functions

select [Dim Date].[Hierarchy].[Month Number Of Year].&[12]&[2012].children

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Instead, members have children.

select [Dim Date].[Hierarchy].[Month Number Of Year].&[12]&[2012].parent

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Similarly, members have parents.

select [Dim Date].[Hierarchy].[Month Number Of Year]

.&[12]&[2012].parent.children

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Parents can have children as well

select [Dim Date].[Hierarchy].[Date Key].&[20121205].parent.parent

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Parents can have parents as well

select [Dim Date].[Hierarchy].[Date Key].&[20121205].grandparent

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- But this is not a grandparent

select ancestor([Dim Date].[Hierarchy].[Date Key].&[20121205]

, 2)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Another way of getting a grand parent

select ancestor([Dim Date].[Hierarchy].[Date Key].&[20121205]

, [Dim Date].[Calendar Year])

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- But this doesn't work

select ancestor([Dim Date].[Hierarchy].[Date Key].&[20121205]

, [Dim Date].[Calendar Year].[Calendar Year])

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- But this doesn't work; this at least gives an error message

Members, tuples or sets must use the same hierarchies in the ANCESTOR function.

select ancestor([Dim Date].[Hierarchy].[Date Key].&[20121205]

,[Dim Date].[Hierarchy].[Calendar Year] )

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Another way of getting a grand parent

select Ascendants([Dim Date].[Hierarchy].[Date Key].&[20121205]

)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- All of the parents and grandparents etc., including itself

select [Dim Date].[Hierarchy].[Calendar Year].&[2012].children

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Children work

select [Dim Date].[Hierarchy].[Calendar Year].&[2012].firstchild

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The first child

select [Dim Date].[Hierarchy].[Calendar Year].&[2012].lastchild

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The last child

select [Dim Date].[Hierarchy].[Calendar Year].&[2012].children.children

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- But not necessarily children.children

select descendants([Dim Date].[Hierarchy].[Calendar Year].&[2012]

, 2)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- This however do not work

select descendants([Dim Date].[Hierarchy].[Calendar Year].&[2012]

, [Dim Date].[Hierarchy].[Date Key])

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- But this does.

select descendants([Dim Date].[Hierarchy].[Calendar Year].&[2012]

)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- This shows all of the children and grandchildren, including itself.

select [Dim Date].[Hierarchy].[Date Key].&[20121205].siblings

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Other days in the month

select cousin([Dim Date].[Hierarchy].[Date Key].&[20121205],

[Dim Date].[Hierarchy].[Month Number Of Year].&[3]&[2012])

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The fifth date in, in another month.

select [Dim Date].[Hierarchy].[Date Key].&[20121205].prevmember

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The previous day

select [Dim Date].[Hierarchy].[Date Key].&[20121205].lag(1)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The previous day; another way

select [Dim Date].[Hierarchy].[Date Key].&[20121205].lag(4)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- 4 days earlier

select [Dim Date].[Hierarchy].[Date Key].&[20121205].nextmember

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The next day

select [Dim Date].[Hierarchy].[Date Key].&[20121205].lead(1)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The next day

select [Dim Date].[Hierarchy].[Date Key].&[20121205].lastsibling

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The end of the month

select [Dim Date].[Hierarchy].[Date Key].&[20121205].firstsibling

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- The start of the month

select tail([Dim Date].[Hierarchy].[Date Key].&[20121205].siblings,1)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

-- Also the end of the month

select head([Dim Date].[Hierarchy].[Date Key].&[20121205].siblings,1)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

--- Also the start of the month

select head([Dim Date].[Hierarchy].[Date Key].&[20121205].siblings,3)

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

where Measures.[Units Balance]

--- The first three days of the month

# Date functions

Problem with YTD

http://www.msbiguide.com/ytd-by-default-a-year-level-was-expected-no-such-level-was-found-in-the-cube-error-in-mdx/

with member TotalToDate as

aggregate(ytd([Dim Date].[Hierarchy].[Date Key].&[20121205])

, [Measures].[Units In])

select TotalToDate

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

-- Year to date

with member TotalToDate as

aggregate(lastperiods(7,[Dim Date].[Hierarchy].[Date Key].&[20121205])

, [Measures].[Units In])

select TotalToDate

on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

-- 5 December 2015 and previous 7 days

with member YTDThisYear as

aggregate(ytd([Dim Date].[Hierarchy].[Date Key].&[20120605])

, [Measures].[Units In])

member YTDLastYear as

aggregate(ytd(parallelperiod([Dim Date].[Hierarchy].[Calendar Year]

,1,[Dim Date].[Hierarchy].[Date Key].&[20120605]))

, [Measures].[Units In])

select {YTDThisYear,YTDLastYear} on 0,

[Dim Product].[Color].[Color].members on 1

from [DWCube]

-- YTD and YTD Last Year

# Functions

select [Measures].[Units In] on 0,

[Dim Product].[English Product Name].[English Product Name].members on 1

from [DWCube]

-- There are quite a lot of Products. Can we reduce them to the top 10?

select [Measures].[Units In] on 0,

topcount([Dim Product].[English Product Name].[English Product Name].members,10) on 1

from [DWCube]

-- This is the top 10 by alphabetical order

select [Measures].[Units In] on 0,

topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 1

from [DWCube]

-- This is the top 10 by [Units In]

select [Measures].[Units In] on 0,

toppercent([Dim Product].[English Product Name]

.[English Product Name].members,10,[Measures].[Units In]) on 1

from [DWCube]

-- This is the top 10% by total [Units In]

select [Measures].[Units In] on 0,

bottomcount([Dim Product].[English Product Name]

.[English Product Name].members,300,[Measures].[Units In]) on 1

from [DWCube]

-- This is the bottom 300 by [Units In]

select [Measures].[Units In] on 0,

bottompercent([Dim Product].[English Product Name]

.[English Product Name].members,10,[Measures].[Units In]) on 1

from [DWCube]

-- This is the bottom 10% by total [Units In]

select [Measures].[Units In] on 0,

topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 1

from [DWCube]

-- This is the top 10 by [Units In]. Can we restrict it to 2012?

select [Measures].[Units In] on 0,

topcount([Dim Product].[English Product Name].[English Product Name].members,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2012])) on 1

from [DWCube]

---The top 10 in 2012 only.

select [Measures].[Units In] on 0,

topcount([Dim Product].[English Product Name].[English Product Name].members,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2013])) on 1

from [DWCube]

---The top 10 in 2013 only. "AWC Logo Cap" is in both.

select [Measures].[Units In] on 0,

union(topcount([Dim Product].[English Product Name].[English Product Name].members,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2013]))

, topcount([Dim Product].[English Product Name].[English Product Name].members,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2012])))

on 1

from [DWCube]

---Here it is for both 2012 and 2013. Note that there are not 20 of them.

select {[Dim Date].[Hierarchy].[Calendar Year].[2012],

[Dim Date].[Hierarchy].[Calendar Year].[2013]} on 0,

union(topcount([Dim Product].[English Product Name].[English Product Name].members,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2013]))

, topcount([Dim Product].[English Product Name].[English Product Name].members,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2012])))

on 1

from [DWCube]

where [Measures].[Units In]

---Here it is separated into 2012 and 2013.

with member Ranking as

RANK([Dim Product].[English Product Name].currentmember,

[Dim Product].[English Product Name].currentmember.siblings,

[Measures].[Units In])

set MyTopItems2012 as

topcount([Dim Product].[English Product Name].[English Product Name].members

,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2012])

)

set MyTopItems2013 as

topcount([Dim Product].[English Product Name].[English Product Name].members

,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2013])

)

select {[Dim Date].[Hierarchy].[Calendar Year].[2012],

[Dim Date].[Hierarchy].[Calendar Year].[2013]} on 0,

union(MyTopItems2012,MyTopItems2013) --

on 1

from [DWCube]

where [Measures].[Ranking]

---Here are the ranks separated into 2012 and 2013.

-- or except or intersect, as in the next two examples

with member Ranking as

RANK([Dim Product].[English Product Name].currentmember,

[Dim Product].[English Product Name].currentmember.siblings,

[Measures].[Units In])

set MyTopItems2012 as

topcount([Dim Product].[English Product Name].[English Product Name].members

,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2012])

)

set MyTopItems2013 as

topcount([Dim Product].[English Product Name].[English Product Name].members

,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2013])

)

select {[Dim Date].[Hierarchy].[Calendar Year].[2012],

[Dim Date].[Hierarchy].[Calendar Year].[2013]} on 0,

intersect(MyTopItems2013, MyTopItems2012)

on 1

from [DWCube]

where [Measures].[Ranking]

---Those that appear in both 2012 and 2013.

with member Ranking as

RANK([Dim Product].[English Product Name].currentmember,

[Dim Product].[English Product Name].currentmember.siblings,

[Measures].[Units In])

set MyTopItems2012 as

topcount([Dim Product].[English Product Name].[English Product Name].members

,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2012])

)

set MyTopItems2013 as

topcount([Dim Product].[English Product Name].[English Product Name].members

,10,

([Measures].[Units In],[Dim Date].[Hierarchy].[Calendar Year].[2013])

)

select {[Dim Date].[Hierarchy].[Calendar Year].[2012],

[Dim Date].[Hierarchy].[Calendar Year].[2013]} on 0,

except(MyTopItems2013,MyTopItems2012)

on 1

from [DWCube]

where [Measures].[Ranking]

---Those that appear in 2013 (the first item), but not 2012.

select non empty [Dim Date].[Calendar Year].[Calendar Year].members on 0,

non empty [Dim Product].[Color].[Color].members on 1

from [DWCube]

where [Measures].[Units In]

-- This shows the entirety of the cube

-- Can we combine the two?

select non empty [Dim Date].[Calendar Year].[Calendar Year].members on 0,

[Dim Product].[Color].[Color].members on 1

from (

select topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 0

from [DWCube])

where [Measures].[Units In]

-- This is the reduced subselect

select non empty [Dim Date].[Calendar Year].members on 0,

[Dim Product].[Color].members on 1

from (

select topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 0

from [DWCube])

where [Measures].[Units In]

-- Because this is all the members of the hierarchy, it includes totals

-- Note that the totals all add up to the reduced cube totals

select [Dim Date].[Calendar Year].[(All)] on 0,

[Dim Product].[Color].[(All)] on 1

from [DWCube]

where [Measures].[Units In]

-- But suppose we wanted the totals for the entire cube instead.

select non empty [Dim Date].[Calendar Year].members on 0,

[Dim Product].[Color].members on 1

from non visual (

select topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 0

from [DWCube])

where [Measures].[Units In]

-- Now the totals are for the entire cube

select non empty [Dim Date].[Calendar Year].members

-[Dim Date].[Calendar Year].[2013] on 0,

[Dim Product].[Color].members on 1

from non visual (

select topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 0

from [DWCube])

where [Measures].[Units In]

-- I don't want to see 2013

select non empty except([Dim Date].[Calendar Year].members

,[Dim Date].[Calendar Year].[2013]) on 0,

[Dim Product].[Color].members on 1

from non visual (

select topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 0

from [DWCube])

where [Measures].[Units In]

-- I still don't want to see 2013

With Set [AllBut2013] as

except([Dim Date].[Calendar Year].members

,[Dim Date].[Calendar Year].[2013])

select non empty [AllBut2013] on 0,

[Dim Product].[Color].members on 1

from non visual (

select topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 0

from [DWCube])

where [Measures].[Units In]

-- Creating Set

create Set [AllBut2013] as

except([Dim Date].[Calendar Year].members

,[Dim Date].[Calendar Year].[2013])

go

select non empty [AllBut2013] on 0,

[Dim Product].[Color].members on 1

from non visual (

select topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 0

from [DWCube])

where [Measures].[Units In]

-- Creating Set; doesn't work

create Set [DWCube].[AllBut2013] as

except([Dim Date].[Calendar Year].members

,[Dim Date].[Calendar Year].[2013])

go

select non empty [AllBut2013] on 0,

[Dim Product].[Color].members on 1

from non visual (

select topcount([Dim Product].[English Product Name].[English Product Name].members,10,[Measures].[Units In]) on 0

from [DWCube])

where [Measures].[Units In]

go

drop set [DWCube].[AllBut2013]

go

-- Creating session Set - does work

(Create set in SSAS.)

with set [AllBut2013] as

except([Dim Date].[Calendar Year].members

,[Dim Date].[Calendar Year].[2013])

set [SomeColors] as

{[Dim Product].[Color].&[Grey],[Dim Product].[Color].&[Red]

,[Dim Product].[Color].[(All)],[Dim Product].[Color].&[Silver]}

select non empty [AllBut2013] on 0,

[SomeColors] on 1

from [DWCube]

where [Measures].[Units In]

-- Another set, with more than one level.

with set [AllBut2013] as

except([Dim Date].[Calendar Year].members

,[Dim Date].[Calendar Year].[2013])

set [SomeColors] as

{[Dim Product].[Color].&[Grey],[Dim Product].[Color].&[Red]

,[Dim Product].[Color].[(All)],[Dim Product].[Color].&[Silver]}

select non empty [AllBut2013] on 0,

HIERARCHIZE([SomeColors]) on 1

from [DWCube]

where [Measures].[Units In]

-- Another set, with more than one level. All is at the top.

with set [AllBut2013] as

except([Dim Date].[Calendar Year].members

,[Dim Date].[Calendar Year].[2013])

set [SomeColors] as

{[Dim Product].[Color].&[Grey],[Dim Product].[Color].&[Red]

,[Dim Product].[Color].[(All)],[Dim Product].[Color].&[Silver]}

select non empty [AllBut2013] on 0,

HIERARCHIZE([SomeColors],post) on 1

from [DWCube]

where [Measures].[Units In]

-- Another set, with more than one level. All is at the bottom.

select [Measures].[Units In] on 0,

order([Dim Product].[English Product Name].[English Product Name].members,[Measures].[Units In]) on 1

from [DWCube]

-- sort ascending

select [Measures].[Units In] on 0,

order([Dim Product].[English Product Name]

.[English Product Name].members,[Measures].[Units In],asc) on 1

from [DWCube]

-- also sort ascending

select [Measures].[Units In] on 0,

order([Dim Product].[English Product Name]

.[English Product Name].members,[Measures].[Units In],desc) on 1

from [DWCube]

-- sort descending

select [Measures].[Units In] on 0,

order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],desc) on 1

from [DWCube]

-- does this really sort descending?

select [Measures].[Units In] on 0,

order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],desc)

\*[Dim Date].[Calendar Year].[Calendar Year].members

on 1

from [DWCube]

-- it is respecting the hierarchy

select [Measures].[Units In] on 0,

order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],bdesc)

\*[Dim Date].[Calendar Year].[Calendar Year].members

on 1

from [DWCube]

-- This is how to break the hierarchy

select [Measures].[Units In] on 0,

non empty order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],basc)

\*[Dim Date].[Calendar Year].[Calendar Year].members

on 1

from [DWCube]

-- This is how to break the hierarchy ascending, non empty

select [Measures].[Units In] on 0,

filter(order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],basc)

\*[Dim Date].[Calendar Year].[Calendar Year].members, [Measures].[Units In]>0)

on 1

from [DWCube]

-- We want more than zero (null) units in.

select [Measures].[Units In] on 0,

filter(order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],basc)

\*[Dim Date].[Calendar Year].[Calendar Year].members, [Measures].[Units In]>=1000)

on 1

from [DWCube]

-- A minimum of 1,000.

select [Measures].[Units In] on 0,

filter(order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],basc)

\*[Dim Date].[Calendar Year].[Calendar Year].members, [Measures].[Units In]>=1000 and [Measures].[Units In]<=3000)

on 1

from [DWCube]

-- Between 1,000 and 3,000

select [Measures].[Units In] on 0,

filter(order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],basc)

\*[Dim Date].[Calendar Year].[Calendar Year].members, not [Measures].[Units In]>=1000 and [Measures].[Units In]<=3000)

on 1

from [DWCube]

-- Not Between 1,000 and 3,000

select [Measures].[Units In] on 0,

filter(order([Dim Date].[Hierarchy].[Month Number Of Year],[Measures].[Units In],basc)

\*[Dim Date].[Calendar Year].[Calendar Year].members, ([Measures].[Units In]>0 and [Measures].[Units In]<1000) or [Measures].[Units In]>3000)

on 1

from [DWCube]

-- Not Between 1,000 and 3,000, and not zero

with set MyDates as

order([Dim Date].[Hierarchy].[Month Number Of Year]

\*[Dim Date].[Calendar Year].[Calendar Year]

, [Measures].[Units In],basc)

member MyMeasure as

[Measures].[Units In], format\_string = 'Yes/No'

select MyMeasure on 0,

filter(MyDates,([Measures].[Units In]>=0 and [Measures].[Units In]<1000)

or [Measures].[Units In]>3000) on 1

from [DWCube]

//General Number, Currency, Fixed, Standard, Percent, Scientific

//Yes/No, True/False, On/Off

# KPIs

--Value

--Goal = Last Year's Value

--Status = Comparison of Last Year's Value with This Year's

--Trend

-- rise - rise = 1

-- fall - fall = -1

-- rise - fall / fall - rise = 0

-- IsEmpty(ThisYear), IsEmpty(LastYear), IsEmpty(The Year Before) = Null

with member UnitsInValue as

[Measures].[Units In]

member UnitsInGoal as

([Measures].[Units In],

parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1,

[Dim Date].[Calendar Year].currentmember))

member UnitsInStatusCalc as

(UnitsInValue-UnitsInGoal) / UnitsInGoal

member UnitsInStatus as

--alternatively, you can use, for part of this:

--iif(UnitsInStatusCalc>.2, 1, iif(UnitsInStatusCalc<-.2,-1,0))

case when isempty(UnitsInGoal) then null

when isempty(UnitsInValue) then null

when UnitsInStatusCalc>.2 then 1

when UnitsInStatusCalc<-.2 then -1

else 0

end

member UnitsInTwoYearsAgo as

([Measures].[Units In],

parallelperiod([Dim Date].[Calendar Year].[Calendar Year],2,

[Dim Date].[Calendar Year].currentmember))

member UnitsInTrend as

case when isempty(UnitsInGoal) then null

when isempty(UnitsInValue) then null

when isempty(UnitsInTwoYearsAgo) then null

when UnitsInValue>UnitsInGoal and UnitsInGoal>UnitsInTwoYearsAgo

then 1

when UnitsInValue<UnitsInGoal and UnitsInGoal<UnitsInTwoYearsAgo

then -1

else 0

end

select {UnitsInValue,UnitsInGoal,UnitsInStatus,UnitsInTrend} on 0,

non empty ([Dim Product].[Color].[Color],

[Dim Date].[Calendar Year].[Calendar Year])

on 1

from [DWCube]

--Here is the LONG version of Status

case when isempty(([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember))) then null

when isempty([Measures].[Units In]) then null

when ([Measures].[Units In]-([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember))) / ([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember))>.2 then 1

when ([Measures].[Units In]-([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember))) / ([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember))<-.2 then -1

else 0

end

-- Here is the LONG version of Trend

case when isempty(([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember))) then null

when isempty([Measures].[Units In]) then null

when isempty(([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],2, [Dim Date].[Calendar Year].currentmember))) then null

when [Measures].[Units In]>([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember)) and ([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember))>([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],2, [Dim Date].[Calendar Year].currentmember))

then 1

when [Measures].[Units In]<([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember)) and ([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],1, [Dim Date].[Calendar Year].currentmember))<([Measures].[Units In], parallelperiod([Dim Date].[Calendar Year].[Calendar Year],2, [Dim Date].[Calendar Year].currentmember))

then -1

else 0

end

select {kpivalue("Units In KPI"),

kpigoal("Units In KPI"),

kpistatus("Units In KPI"),

kpitrend("Units In KPI")} on 0,

non empty ([Dim Product].[Color].[Color],

[Dim Date].[Calendar Year].[Calendar Year])

on 1

from [DWCube]