Tip or Technique

OLAP Functions

Product(s): Cognos 8 BI

Area of Interest: Report Design

Copyright

This document is maintained by the Best Practices, Product and Technology team. You can send comments, suggestions, and additions to Best Practices, Product and Technologies.

Contents

1 Introduction

1.1 Purpose

This document provides additional detail and examples of the OLAP functions provided within Report Studio for Cognos 8.

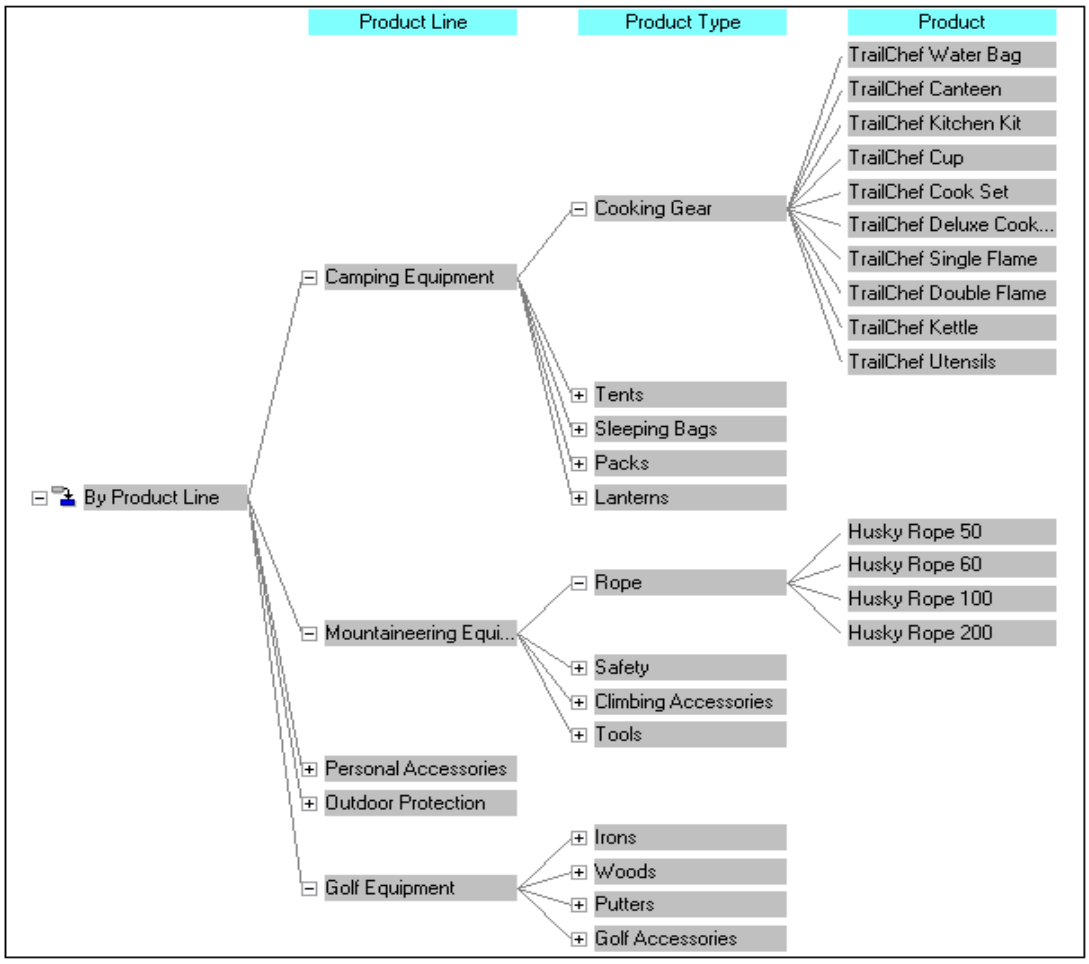
1.2 Applicability

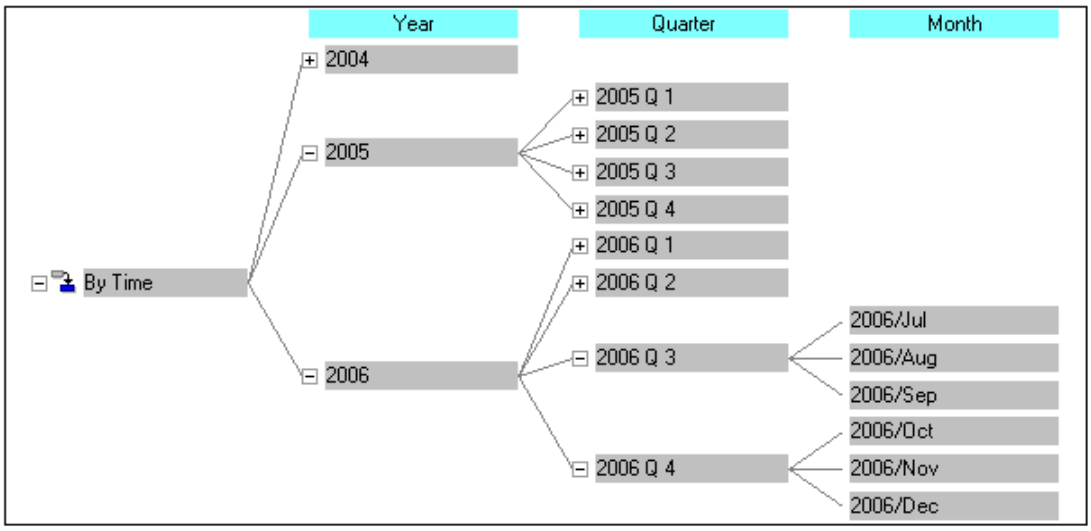
The functions and sample data listed below apply to Cognos 8, Report Studio. The PowerCube and deployment package for "Great Outdoors Company" are provided with the product samples. Please see the Administration and Security Guide for Cognos 8 regarding the steps to restore these samples.

The provided deployment of sample reports can be found at the end of this document. This deployment file is valid for Cognos 8 MR1.

2 Background Information

Below is an example hierarchy from the Product Dimension of the Great Outdoors Company PowerCube showing the members below the Camping Equipment Product Line. This dimension will be the basis of many of the subsequent examples.





3 Function Definitions

3.1 _firstFromSet

Syntax:

_firstFromSet (set_exp, numeric_exp_max, numeric_exp_overflow)

Description:

Returns the first members found in the set up to numeric_exp_max + numeric_exp_overflow. If numeric_exp_max + numeric_exp_overflow is exceeded, then only the max number of members are returned.

For a set that has only a few members more than the specified numeric_exp_max the numeric_exp_overflow allows the small set of extra members to be included. If the set has a large number of members then only the numeric_exp_max members will be returned as the remainder of the set is greater than the specified overflow value.

Example:

_firstFromSet([great_outdoors_company].[Products].[Products].[Product line],2,8)
result: Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories

_firstFromSet([great_outdoors_company].[Products].[Products].[Product line],2,2)
result: Camping Equipment
Golf Equipment

3.2 _remainderSet

Syntax:

_remainderSet (member_exp, set_exp, numeric_exp)

Description:

The member expression will be included in the returned set when the size of the set_exp set is greater than numeric_exp. i.e. a new member will be generated if the number of members in set_exp larger than the specified numeric_exp

Example:

	<#Quantity sold#>	<#Quantity sold#>
<#_remainderSet Calc#>	<#1234#>	<#1234#>
<#_remainderSet Calc#>	<#1234#>	<#1234#>

_remainderSet(member(aggregate(currentMeasure WITHIN SET [great_outdoors_company].[Products].[Products].[Product line]), 'Product Aggregate', 'Product Aggregate', [great_outdoors_company].[Products].[Products]),[great_outdoors_company].[Products].[Products].[Product line],1)
result:

	Quantity sold
Product Aggregate	2,215,354

_remainderSet(member(aggregate(currentMeasure WITHIN SET [great_outdoors_company].[Products].[Products].[Product line]), 'Product Aggregate', 'Product Aggregate', [great_outdoors_company].[Products].[Products]),[great_outdoors_company].[Products].[Products].[Product line],100)
result:

	Quantity sold

3.3 ancestor

Syntax:

ancestor (member, level | integer)

Description:

Returns the ancestor of the specified member at either the specified (named) level or the specified number of levels above the member.

Note: The result is not guaranteed to be consistent when there is more than one such ancestor.

Example:

ancestor([TrailChef Water Bag], 1)
result: Cooking Gear

ancestor([TrailChef Water Bag], 2)
result: Camping Equipment

ancestor([TrailChef Water Bag],[great_outdoors_company].[Products].[Products].[Product type])
result: Cooking Gear

3.4 ancestors

Syntax:

ancestors (member, level | integer)

Description:

Returns all the ancestors of a member at a specified level, or distance above the member. (Most data sources support only one ancestor at a specified level, but some support more than one. Hence the result is a member set.)

Example:

ancestors([TrailChef Water Bag], 1)
result: Cooking Gear

ancestors([TrailChef Water Bag], 2)
result: Camping Equipment

ancestors([TrailChef Water Bag],[great_outdoors_company].[Products].[Products].[Product type])
result: Cooking Gear

3.5 bottomCount

Syntax:

bottomCount (set_exp , index_exp , numeric_exp)

Description:

This function sorts a set according to the value of "numeric_expression" evaluated at each of the members of "set_exp", and returns the bottom "index_exp" members.

Example:

Based on a crosstab report using the intersection of [2006] and the default measure [Revenue] to determine numeric_exp.

bottomCount(set([Camping Equipment],[Golf Equipment],[Mountaineering Equipment]),2,[2006])

Revenue	*<#Function#>*	<#Camping Equipment#>	<#Golf Equipment#>	<#Mountaineering Equipment#>
<#2004#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2005#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2006#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>

result:

Revenue	*Golf Equipment*	*Mountaineering Equipment*	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	*$5,597,980.86*	*$0.00*	$20,471,328.88	$5,597,980.86	$0.00
2005	*$9,598,268.88*	*$9,642,674.54*	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	*$10,709,215.84*	*$11,248,676.06*	$37,869,055.58	$10,709,215.84	$11,248,676.06

Based on a list report using a direct reference to the [Revenue] measure for numeric_exp.

bottomCount([great_outdoors_company].[Products].[Products].[Product line],2,[Revenue])
result: Outdoor Protection $3,171,114.92
Mountaineering Equipment $20,891,350.60

3.6 bottomPercent

Syntax:

bottomPercent (set_exp , numeric_exp1 , numeric_exp2)

Description:

This function is similar to bottomSum, but the threshold is "numeric_exp1" percent of the total.

numeric_exp1 ranges from 0 to 100.

Example:

Based on a crosstab report

bottomPercent(set([Camping Equipment],[Golf Equipment],[Mountaineering Equipment]),40,[2006])

For the set of Camping Equipment, Golf Equipment and Mountaineering Equipment return the members whose percentage total are greater than or equal to 40% for the tuple 2006

Revenue	*<#Function#>*	<#Camping Equipment#>	<#Golf Equipment#>	<#Mountaineering Equipment#>
<#2004#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2005#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2006#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>

result:

Revenue	*Golf Equipment*	*Mountaineering Equipment*	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	*$5,597,980.86*	*$0.00*	$20,471,328.88	$5,597,980.86	$0.00
2005	*$9,598,268.88*	*$9,642,674.54*	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	*$10,709,215.84*	*$11,248,676.06*	$37,869,055.58	$10,709,215.84	$11,248,676.06

bottomPercent(set([Camping Equipment],[Golf Equipment],[Mountaineering Equipment]),20,tuple([2006],[great_outdoors_company].[Measures].[Gross profit]))

result:

Revenue	*Mountaineering Equipment*	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	*$0.00*	$20,471,328.88	$5,597,980.86	$0.00
2005	*$9,642,674.54*	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	*$11,248,676.06*	$37,869,055.58	$10,709,215.84	$11,248,676.06

3.7 bottomSum

Syntax:

bottomSum (set_exp , numeric_exp1 , numeric_exp2)

Description:

This function sorts on "numeric_exp2", evaluated at the corresponding member of "set_exp", and picks up the bottommost elements whose cumulative total is at least "numeric_exp1".

Example:

Revenue	*<#Function#>*	<#Camping Equipment#>	<#Golf Equipment#>	<#Mountaineering Equipment#>
<#2004#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2005#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2006#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>

Based on a crosstab report

bottomSum(members([great_outdoors_company].[Products].[Products].[Product line]),6000000, tuple([2006],[great_outdoors_company].[Measures].[Gross profit]))
result:

Revenue	*Outdoor Protection*	*Mountaineering Equipment*	*Golf Equipment*	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	*$1,536,456.24*	*$0.00*	*$5,597,980.86*	$20,471,328.88	$5,597,980.86	$0.00
2005	*$988,230.64*	*$9,642,674.54*	*$9,598,268.88*	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	*$646,428.04*	*$11,248,676.06*	*$10,709,215.84*	$37,869,055.58	$10,709,215.84	$11,248,676.06

3.8 caption

Syntax:

caption (level | member | set_exp)

Description:

Returns the caption values of the specified element.

The caption is the string display name for an element and does not necessarily match the unique identifier used to generate the business key or member unique name for the element. The caption is not necessarily unique. The caption for a month may return the month name without further year details to make the value unique.

Example:

caption([TrailChef Water Bag])
result: TrailChef Water Bag

caption([great_outdoors_company].[Products].[Products].[Product line])
result: Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment

3.9 children

Syntax:

children (member)

Description:

Returns the set of children of a specified member.

Example:

children([Camping Equipment])
result: Cooking Gear
Tents
Sleeping Bags
Packs
Lanterns

3.10 closingPeriod

Syntax:

closingPeriod (level [, member])

Description:

Returns the last sibling among the descendants of a member at a specified level. Typically used with a time dimension.

Example:

closingPeriod([great_outdoors_company].[Years].[Years].[Month])
result: 2006/Dec

closingPeriod([great_outdoors_company].[Years].[Years].[Year])
result: 2006

closingPeriod([great_outdoors_company].[Years].[Years].[Month],[2006 Q 4])
result: 2006/Dec

3.11 completeTuple

Syntax:

completeTuple (member { , member })

Description:

Similar to "tuple", identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. However, completeTuple implicitly includes the default member from all dimensions not otherwise specified in the arguments, rather than the current member. The value of this cell can be obtained with the "value" function.

Example:

Below, the first column uses the expression:

completetuple([Mountaineering Equipment],[Fax])

The second column uses a more precise expression:

completetuple([Mountaineering Equipment],[Fax],[Quantity sold],currentMember([great_outdoors_company].[Years].[Years]))

result:

Quantity sold	*Mountaineering Sales by Fax (default Measure, default year)*	*Mountaineering Sales by Fax (specific Measure, current year)*
2004	*$1,220,329.38*	*0*
2005	*$1,220,329.38*	*8,746*
2006	*$1,220,329.38*	*7,860*

The completetuple does not pick up the currentMember by default as the tuple function does. The values in the first column are identical across each year because the default member of the Years dimension, the root member, is used rather than the current member. Likewise, the first column displays Revenue rather than Quantity Sold because the Revenue measure is the default from the Measures dimension. Completetuple will use the default measure rather than the currentMeasure in the query if the measure is not defined in the completetuple function.

The second column in the above output specifies that the completetuple function is to use the currentMember of the Years dimension and the Quanityt sold measure. This replicates the example

below that is used for the Tuple function where the currentMember and currentMeasure are selected by default.

3.12 cousin

Syntax:

cousin (member1 , member2)

Description:

Returns the child member of member2 with the same relative position as the member1 is under its parent.

Example:

cousin([Irons],[Camping Equipment])
result: Cooking Gear

cousin([Putters],[Camping Equipment])
result: Sleeping Bags

3.13 currentMember

Syntax:

currentMember (hierarchy)

Description:

Returns the current member of the hierarchy during an iteration. If the specified hierarchy is not present in the context in which the expression is being evaluated, its default member is assumed.

Example:

currentMember([great_outdoors_company].[Products].[Products])

Quantity sold		Columns
<#Product line#>	<#currentMember#>	<#1234#>
	<#currentMember#>	<#1234#>
<#Product type#>	<#currentMember#>	<#1234#>
	<#currentMember#>	<#1234#>

results:

Camping Equipment	Camping Equipment	866,234
Golf Equipment	Golf Equipment	99,400
Mountaineering Equipment	Mountaineering Equipment	301,958
Outdoor Protection	Outdoor Protection	557,854
Personal Accessories	Personal Accessories	389,908
Cooking Gear	Cooking Gear	198,676
Sleeping Bags	Sleeping Bags	96,246
Packs	Packs	95,552
Tents	Tents	130,664
Lanterns	Lanterns	345,096
Irons	Irons	14,244
Putters	Putters	23,244
Woods	Woods	13,924
Golf Accessories	Golf Accessories	47,988
Climbing Accessories	Climbing Accessories	142,150
Tools	Tools	96,798
Rope	Rope	40,678
Safety	Safety	22,332
First Aid	First Aid	72,348
Insect Repellents	Insect Repellents	270,074
Sunscreen	Sunscreen	215,432
Binoculars	Binoculars	43,330
Navigation	Navigation	56,666
Eyewear	Eyewear	53,510
Knives	Knives	134,134
Watches	Watches	102,268

3.14 defaultMember

Syntax:

defaultMember (hierarchy)

Description:

Returns the default member of a hierarchy.

Example:

defaultMember([great_outdoors_company].[Products].[Products])
result: Products

defaultMember([great_outdoors_company].[Years].[Years])
result: Years

defaultMember(hierarchy([great_outdoors_company].[Measures].[Quantity sold]))
result: Revenue

defaultMember Measures	<#defaultMember Measures#>	<#Revenue#>	<#Quantity sold#>
<#defaultMember Products#>	<#1234#>	<#1234#>	<#1234#>
<#defaultMember Products#>	<#1234#>	<#1234#>	<#1234#>

result:

defaultMember Measures	defaultMember Measures	Revenue	Quantity sold
Products	$171,576,387.88	$171,576,387.88	2,215,354

3.15 descendants

Syntax:

descendants (set_expr , level | index [, { self | before | beforewithmember | after }])

Description:

Returns the set of descendants a set of members at a specified level or distance from the root, with the option of including or excluding descendants in other levels. Duplicates will be removed from the set.

Example:

Note: [great_outdoors_company].[Products].[Products].[Products] is the root member of the Products hierarchy.

descendants([Member], [Level]) or descendants([Member], [Level], self)

descendants([great_outdoors_company].[Products].[Products].[Products],[great_outdoors_company].[Products].[Products].[Product type])

*results:*Cooking Gear
Sleeping Bags
Packs
Tents
…
First Aid
Insect Repellents
Sunscreen
Binoculars
Navigation
Eyewear
Knives
Watches

descendants([Member], <distance>)

descendants([great_outdoors_company].[Products].[Products].[Products], 1)
*results:*Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories

descendants([Member], <distance>, before)

descendants([great_outdoors_company].[Products].[Products].[Products], 3, before)
*results:*Camping Equipment
Cooking Gear
Sleeping Bags
Packs

Tents
Lanterns
Golf Equipment
Irons
Putters
Woods
Golf Accessories
Mountaineering Equipment
Climbing Accessories
Tools
Rope
Safety
Outdoor Protection
First Aid
Insect Repellents
Sunscreen
Personal Accessories
Binoculars
Navigation
Eyewear
Knives
Watches

descendants([great_outdoors_company].[Products].[Products].[Products], 2, self before)
results: Camping Equipment
Cooking Gear
Sleeping Bags
Packs
Tents
Lanterns
Golf Equipment
Irons
Putters
Woods
Golf Accessories
Mountaineering Equipment
Climbing Accessories
Tools
Rope
Safety
Outdoor Protection
First Aid
Insect Repellents
Sunscreen
Personal Accessories
Binoculars
Navigation
Eyewear
Knives
Watches

3.16 emptySet

Syntax:

emptySet (hierarchy)

Description:

Returns an empty member set for the specified hierarchy.

This is most often used as a placeholder during development or with dynamic report design (either with the SDK or via report design). By creating a data item that contains the emptyset function it is possible to build complex expressions that can later be revised by redefining the emptyset data item.

Example:

except([great_outdoors_company].[Products].[Products].[Product line],emptyset([great_outdoors_company].[Products].[Products]))
*results:*Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories

3.17 except

Syntax:

except (set_exp1 , set_exp2 [,ALL])

Description:

Returns the members of "set_exp1" that are not also in "set_exp2". Duplicates are retained only if the optional keyword ALL is supplied as the third argument.

Example:

except(set([Camping Equipment],[Mountaineering Equipment]), set([Camping Equipment],[Golf Equipment]))
result: Mountaineering Equipment

3.18 filter

Syntax:

filter (set_exp , boolean_exp)

Description:

Returns the set resulting from filtering a specified set based on the boolean condition. Each member is included in the result if and only if the corresponding value of "boolean_exp" is true.

Example:

Source Data:

Product line	Gross margin
Camping Equipment	29.28%
Golf Equipment	46.39%
Mountaineering Equipment	37.43%
Outdoor Protection	55.96%
Personal Accessories	35.27%

Expression <#High Margin Product Lines#>:
filter([Product line], [Gross margin] > .30)

Revenue	<#Year#>	<#Year#>
<#High Margin Product Lines#>	<#1234#>	<#1234#>
<#High Margin Product Lines#>	<#1234#>	<#1234#>

Results:

Revenue	2004	2005	2006
Golf Equipment	$5,597,980.86	$9,598,268.88	$10,709,215.84
Mountaineering Equipment	$0.00	$9,642,674.54	$11,248,676.06
Outdoor Protection	$1,536,456.24	$988,230.64	$646,428.04
Personal Accessories	$7,144,797.52	$10,955,708.04	$13,793,960.30

Source Data:

Year	Product line	Gross margin
2006	Camping Equipment	28.18%
2006	Golf Equipment	45.58%
2006	Mountaineering Equipment	36.59%
2006	Outdoor Protection	55.50%
2006	Personal Accessories	34.13%

Expression:
filter([Product line], tuple([Gross margin], [2006]) > .30)
Results:

Revenue	2004	2005	2006
Golf Equipment	$5,597,980.86	$9,598,268.88	$10,709,215.84
Mountaineering Equipment	$0.00	$9,642,674.54	$11,248,676.06
Outdoor Protection	$1,536,456.24	$988,230.64	$646,428.04
Personal Accessories	$7,144,797.52	$10,955,708.04	$13,793,960.30

3.19 firstChild

Syntax:

firstChild (member)

Description

Returns the first child of a member.

Example:

firstChild([By Product Lines])
result: Camping Equipment

firstChild([Camping Equipment])
result: Cooking Gear

3.20 firstSibling

Syntax:

firstSibling (member)

Description:

Returns the first child of the parent of a member.

Example:

firstSibling ([Outdoor Protection])
result: Camping Equipment

firstSibling ([Camping Equipment])
result: Camping Equipment

3.21 generate

Syntax:

generate (set_exp1 , set_exp2 [, ALL])

Description:

This function evaluates "set_exp2" for each member of "set_exp1" and joins the resulting sets by union. If ALL is specified, duplicates in the result are retained.

Example:

The following example calculates the top 2 products by revenue for each product line.

generate([Product line], topCount(descendants(currentMember([great_outdoors_company].[Products].[Products]), [great_outdoors_company].[Products].[Products].[Product name]), 2, [Revenue]))
result:

		Revenue
Camping Equipment	Star Dome	$14,948,640.60
	Star Gazer 3	$13,044,951.40
Golf Equipment	Hailstorm Titanium Woods Set	$6,264,117.96
	Hailstorm Steel Woods Set	$4,519,643.36
Mountaineering Equipment	Husky Rope 100	$4,613,953.30
	Husky Rope 60	$2,559,977.34
Outdoor Protection	BugShield Extreme	$426,718.18
	BugShield Lotion Lite	$335,255.84
Personal Accessories	Mountain Man Extreme	$4,334,879.56
	Glacier GPS Extreme	$3,089,784.26

3.22 head

Syntax:

head (set_exp [, index_exp])

Description:

Returns the first "index_exp" elements of "set_exp". The default for "index_exp" is 1.

Example:

head(members([great_outdoors_company].[Products].[Products].[Product line]))
result: Camping Equipment

head(members([great_outdoors_company].[Products].[Products].[Product line]),2)

result: Camping Equipment
Mountaineering Equipment

3.23 hierarchize

Syntax:

hierarchize (set_exp)

Description:

This function orders the members of a set in a hierarchy. Members in a level are sorted in their natural order, which is the default ordering of the members along a dimension when no other sort conditions are specified.

Example:

Revenue	<#Function#>	<#Camping Equipment#>	<#Golf Equipment#>	<#Mountaineering Equipment#>
<#2004#>	<#1234#>	<#1234#>	<#1234#>	<#1234#>
<#2005#>	<#1234#>	<#1234#>	<#1234#>	<#1234#>
<#2006#>	<#1234#>	<#1234#>	<#1234#>	<#1234#>

hierarchize(set([Golf Equipment],[Mountaineering Equipment],[Camping Equipment]))
result:

Revenue	Camping Equipment	Golf Equipment	Mountaineering Equipment	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	$20,471,328.88	$5,597,980.86	$0.00	$20,471,328.88	$5,597,980.86	$0.00
2005	$31,373,606.46	$9,598,268.88	$9,642,674.54	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	$37,869,055.58	$10,709,215.84	$11,248,676.06	$37,869,055.58	$10,709,215.84	$11,248,676.06

3.24 hierarchy

Syntax:

hierarchy (level | member | set_exp)

Description:

Returns the hierarchy that contains the specified level, member or member set.

Example:

hierarchy([Cooking Gear])
result: (every member in the hierarchy that contains Cooking Gear)
Products
Camping Equipment
Cooking Gear
TrailChef Water Bag
TrailChef Canteen
TrailChef Deluxe Cook Set
TrailChef Double Flame
TrailChef Kettle
TrailChef Kitchen Kit
TrailChef Cup
TrailChef Cook Set
TrailChef Single Flame
TrailChef Utensils
Sleeping Bags
Hibernator Extreme

...
Mountain Man Analog
Mountain Man Digital
Mountain Man Combination
Mountain Man Extreme
Mountain Man Deluxe

hierarchy([great_outdoors_company].[Products].[Products].[Product line])
result: (every member in the hierarchy that contains the Product line)
Products
Camping Equipment
Cooking Gear
TrailChef Water Bag
TrailChef Canteen
TrailChef Deluxe Cook Set
TrailChef Double Flame
TrailChef Kettle
TrailChef Kitchen Kit
TrailChef Cup
TrailChef Cook Set
TrailChef Single Flame
TrailChef Utensils
Sleeping Bags
Hibernator Extreme
...
Mountain Man Analog
Mountain Man Digital
Mountain Man Combination
Mountain Man Extreme
Mountain Man Deluxe

3.25 item

Syntax:

item (set_exp , index)

Description:

Returns a member from a specified location within a set. The index into the set is zero based

Example:

item(children([Camping Equipment]),2)
result: Sleeping Bags

3.26 intersect

Syntax:

intersect (set_exp1 , set_exp2 [, ALL])

Description:

Returns the intersection of two input sets. The result retains duplicates only when the optional keyword ALL is supplied as the third argument.

Example:

intersect(set([Camping Equipment],[Mountaineering Equipment]), set([Camping Equipment],[Outdoor Protection],),ALL)
result: Camping Equipment

3.27 lag

Syntax:

lag (member , index_exp)

Description:

Returns the sibling member that is a specified number of positions prior to a specified member.

Example:

lag([Tents],1)
result: Cooking Gear

lag([Tents],-2)
result: Packs

3.28 lastChild

Syntax:

lastChild (member)

Description:

Returns the last child of a specified member.

Example:

lastChild(Cooking Gear)
result: TrailChief Utensils

lastChild([By Product Line])
result: Golf Equipment

3.29 lastPeriods

Syntax:

lastPeriods (integer_exp , member)

Description:

Returns the set of members from the same level that ends with the specified member. The number of members returned is the absolute value of "integer_exp". If "integer_exp" is negative then members following and including the specified member are returned. Typically used with a time dimension.

Example:

lastPeriods(2,[2006 Q 4])
result: 2006 Q 3
2006 Q 4

lastPeriods(-3,[2006 Q 4])
result: 2006 Q 2
2006 Q 3
2006 Q 4

3.30 lastSibling

Syntax:

lastSibling (member)

Description:

Returns the last child of the parent of a specified member.

Example:

lastSibling ([Camping Equipment])
result: Golf Equipment

3.31 lead

Syntax:

lead (member , index_exp)

Description:

Returns the sibling member that is a specified number of positions following a specified member.

Example:

lead ([Outdoor Protection],1)
result: Golf Equipment

lead ([Outdoor Protection],-2)
result: Mountaineering Equipment

3.32 level

Syntax:

level (member)

Description:

Returns the level of a member.

Example:

level ([Golf Equipment])
result: Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment

level ([Outdoor Protection])
result: Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment

3.33 levels

Syntax:

levels (hierarchy , index)

Description:

Returns the level's members in the hierarchy whose distance from the root is specified by "index".

Example:

levels([great_outdoors_company].[Products].[Products],2)
result: Cooking Gear
Sleeping Bags
Packs
Tents
...
Irons
Putters
Woods
Golf Accessories

levels([great_outdoors_company].[Products].[Products],1)
result: Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment

3.34 member

Syntax:

member (value_exp [, string1 [, string2 [, hierarchy]]])

Amember + b Member , unique id, caption, hierarchy

Description:

Defines a member based on the specified expression in the specified hierarchy. "string1" is used to identify the member created by this function it must be unique in the query, and must be different from any other member in the same hierarchy. "string2" is used as the caption of the member; if it is absent, the caption is empty.

If the hierarchy is omitted, the measure dimension is assumed. Note: All calculations used as grouping items whose sibling items are other calculations or member sets should be explicitly assigned to a hierarchy using this function, otherwise the results are not predictable. The only exception to this is where the calculation involves only members of the same hierarchy as the siblings. In that case the calculation is assumed to belong to that hierarchy.

Example:

member(total([great_outdoors_company].[Measures].[Quantity sold] within set [2005]),'Quantity2005','Quantity sold 2005')

	<#Quantity sold#>	<#Quantity sold 2005#>
<#Year#>	<#1234#>	<#1234#>
<#Year#>	<#1234#>	<#1234#>

result:

	Quantity sold	Quantity sold 2005
2004	561,706	789,730
2005	789,730	789,730
2006	863,918	789,730

Or rolling up all the products that start with the letter B

member(total(currentMeasure within set filter([great_outdoors_company].[Products].[Products].[Product

name],caption([great_outdoors_company].[Products].[Products].[Product name]) starts with 'B')),'BProducts','B Products',[great_outdoors_company].[Products].[Products])

result:

	Quantity sold	Revenue
B Products	*335,610*	*$6,411,860.14*

3.35 members

Syntax:

members (hierarchy | level)

Description:

Returns the set of members in a hierarchy or level. In the case of a hierarchy, the order of the members in the result is not guaranteed; if a predictable order is required, an explicit ordering function (such as hierarchize) must be used.

Example:

members([great_outdoors_company].[Years].[Years])

result: Years
2004
2004 Q 1
2004/Jan
2004/Feb
2004/Mar
2004 Q 2
2004/Apr
...
2006 Q 3
2006/Jul
2006/Aug
2006/Sep
2006 Q 4
2006/Oct
2006/Nov
2006/Dec

members([great_outdoors_company].[Products].[Products].[Product line])

result: Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment

3.36 nestedSet

Syntax:

nestedSet (set_expr1 , set_expr2)

Description:

Returns the set of members of set_expr2 evaluated in the context of the current member of set_expr1.

Example:

The following example calculates the top 2 products by revenue for each product line.

nestedSet(members([Product line]), topCount(descendants(currentMember([great_outdoors_company].[Products].[Products]), [great_outdoors_company].[Products].[Products].[Product name]), 2, [Revenue]))
result:

		Revenue
Camping Equipment	Star Dome	$14,948,640.60
	Star Gazer 3	$13,044,951.40
Golf Equipment	Hailstorm Titanium Woods Set	$6,264,117.96
	Hailstorm Steel Woods Set	$4,519,643.36
Mountaineering Equipment	Husky Rope 100	$4,613,953.30
	Husky Rope 60	$2,559,977.34
Outdoor Protection	BugShield Extreme	$426,718.18
	BugShield Lotion Lite	$335,255.84
Personal Accessories	Mountain Man Extreme	$4,334,879.56
	Glacier GPS Extreme	$3,089,784.26

3.37 nextMember

Syntax:

nextMember (member)

Description:

Returns the next member in the level to which the specified member exists.

Example:

nextMember([Outdoor Protection])
result: Golf Equipment

3.38 openingPeriod

Syntax:

openingPeriod (level [, member])

Description:

Returns the first sibling member among the descendants of a member at a specified level. Typically used with a time dimension.

Example:

openingPeriod([great_outdoors_company].[Years].[Years].[Month])
result: 2004/Jan

openingPeriod([great_outdoors_company].[Years].[Years].[Year])
result: 2004

openingPeriod([great_outdoors_company].[Years].[Years].[Month],[2006 Q 4])
result: 2006/Oct

3.39 order

Syntax:

order (set_exp , value_exp [, ASC | DESC | BASC | BDESC])

Description:

Arranges members of a specified set, as determined from the set of values created by evaluating "value_exp" for each value of the set, and modified by the third parameter.

There are two varieties of order: hierarchized (ASC or DESC) and non-hierarchized (BASC or BDESC, where B stands for "break hierarchy"). The hierarchized ordering first arranges members according to their position in the hierarchy. Then it orders the children of each member according to "value_exp". The non-hierarchized ordering arranges members in the set without regard to the hierarchy. In the absence of an explicit specification, ASC is the default.

Example:

order(members([Great Outdoors Company].[Product].[Product].[Product type]),[Quantity sold], BASC)

and

order(members([Great Outdoors Company].[Product].[Product].[Product type]),[Quantity sold], ASC)

result:

Default Order

		Quantity
Camping Equipment	*Cooking Gear*	*198,676*
	Lanterns	*345,096*
	Packs	*95,552*
	Sleeping Bags	*96,246*
	Tents	*130,664*
Golf Equipment	*Golf Accessories*	*47,988*
	Irons	*14,244*
	Putters	*23,244*
	Woods	*13,924*
Mountaineering Equipment	*Climbing Accessories*	*142,150*
	Rope	*40,678*
	Safety	*22,332*
	Tools	*96,798*
Outdoor Protection	*First Aid*	*72,348*
	Insect Repellents	*270,074*
	Sunscreen	*215,432*
Personal Accessories	*Binoculars*	*43,330*
	Eyewear	*53,510*
	Knives	*134,134*
	Navigation	*56,666*
	Watches	*102,268*

ORDER BASC

	Quantity
Woods	*13,924*
Irons	*14,244*
Safety	*22,332*
Putters	*23,244*
Rope	*40,678*
Binoculars	*43,330*
Golf Accessories	*47,988*
Eyewear	*53,510*
Navigation	*56,666*
First Aid	*72,348*
Packs	*95,552*
Sleeping Bags	*96,246*
Tools	*96,798*
Watches	*102,268*
Tents	*130,664*
Knives	*134,134*
Climbing Accessories	*142,150*
Cooking Gear	*198,676*
Sunscreen	*215,432*
Insect Repellents	*270,074*
Lanterns	*345,096*

ORDER ASC

	Quantity
Woods	*13,924*
Irons	*14,244*
Putters	*23,244*
Golf Accessories	*47,988*
Safety	*22,332*
Rope	*40,678*
Tools	*96,798*
Climbing Accessories	*142,150*
Binoculars	*43,330*
Eyewear	*53,510*
Navigation	*56,666*
Watches	*102,268*
Knives	*134,134*
First Aid	*72,348*
Sunscreen	*215,432*
Insect Repellents	*270,074*
Packs	*95,552*
Sleeping Bags	*96,246*
Tents	*130,664*
Cooking Gear	*198,676*
Lanterns	*345,096*

3.40 ordinal

Syntax:

ordinal (level)

Description:

Returns the zero-based ordinal value (distance from the root level) of the specified level.

Example:

ordinal([great_outdoors_company].[Products].[Products].[Product line])
result: 1

ordinal([great_outdoors_company].[Products].[Products].[Product type])
result: 2

3.41 parallelPeriod

Syntax:

parallelPeriod (level , int_exp , member)

Description:

Returns a member from a different period in the same relative position as a specified member. This function is similar to the "Cousin" function, but is more closely related to time series. It takes the ancestor of "member" at "level" (call it "ancestor"); then it takes the sibling of "ancestor" that is offset (follows) by "int exp" positions, and returns the descendants of that sibling in the same relative position as the specified member as under "ancestor".

Example:

parallelPeriod([great_outdoors_company].[Years].[Years].[Quarter],-1,[2006/Aug])
result: 2006/Nov

parallelPeriod([great_outdoors_company].[Years].[Years].[Quarter],1,[2006/Aug])
result: 2006/May

parallelPeriod([great_outdoors_company].[Years].[Years].[Year],2,[2006/Aug])
result: 2004/Aug

3.42 parent

Syntax:

parent (member)

Description:

Returns the member that is the parent of the specified member.

Example:

parent([Cooking Gear])
result: Camping Equipment

3.43 periodsToDate

Syntax:

periodsToDate (level , member)

Description:

Returns a set of sibling members from the same level as a given member, as constrained by a specified level.

It locates the ancestor of "member" at "level", and returns that ancestor's descendants at the same level as "member", up to and including "member". Typically used with a time dimension.

Example:

periodsToDate([great_outdoors_company].[Years].[Years].[Year], [2004/Mar])
result: returns the value for [2004/Jan], [2004/Feb], [2004/Mar]

3.44 prevMember

Syntax:

prevMember (member)

Description:

Returns the member that immediately precedes the specified member in the same level.

Example:

prevMember ([Outdoor Protection])
result: Personal Accessories

prevMember ([2005])
result: 2004

3.45 roleValue

Syntax:

roleValue (string [, member | set_exp])

Description:

Returns the value of the attribute that is associated with the role whose name is specified by "string" within the specified context. The second argument is optional only in a number of limited circumstances, where it can be derived from other context. Applications can be made portable across different data sources and models by accessing attributes by role, rather than by query item ID.

(For dimensionally modelled relational data sources, assignment of roles is the modeller's responsibility.)

Intrinsic roles that are defined for members of all data source types include: _businessKey, _memberCaption, _memberDescription, _memberUniqueName.

Additional roles can be defined in Framework Manager for each level in a hierarchy. For example, a Product type level may have an attribute column called "Type Shipping Container" and the Product level may have a "Product Shipping Container" attribute. Each of these could be assigned a custom role in Framework Manager called "Container". The property could then be referenced independently of the actual column name by using the roleValue function.

Example:

roleValue('_businessKey',[great_outdoors_company].[Years].[Years].[Year])
result: ("2004-01-01","2004-12-31")
("2005-01-01","2005-12-31")
("2006-01-01","2006-12-31")

roleValue('_memberUniqueName',[great_outdoors_company].[Years].[Years].[Year])

result: [great_outdoors_company].[Years].[Years].[Year]->:[PC].[Years (Root)].[20040101-20041231]
[great_outdoors_company].[Years].[Years].[Year]->:[PC].[Years (Root)].[20050101-20051231]
[great_outdoors_company].[Years].[Years].[Year]->:[PC].[Years (Root)].[20060101-20061231]

3.46 rootMemebers

Syntax:

rootMembers (hierarchy)

Description:

Returns the root members of a hierarchy.

Example:

rootMembers([great_outdoors_company].[Years].[Years])
result: By Time

3.47 set

Syntax:

set (member { , member })

Description:

Returns a list of members belonging to the same hierarchy

Example:

set([Golf Equipment], [Irons], [TrailChef Cup])
result: Golf Equipment
Irons
TrailChef Cup

3.48 siblings

Syntax:

siblings (member)

Description:

Returns the children of the parent of the specified member.

Example:

siblings([Golf Equipment])
result: Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories

3.49 subset

Syntax:

subset (set_exp, index_exp1 [, index_exp2])

Description:

Returns a subset of members from a specified set starting "index_exp1" from the beginning. If the count "index_exp2" is specified, that many members (if available) are returned. Otherwise, all remaining members are returned.

Example:

subset(members([great_outdoors_company].[Products].[Products].[Product line]), 2)
result: Mountaineering Equipment
Outdoor Protection
Personal Accessories

subset(members([great_outdoors_company].[Products].[Products].[Product line]), 2, 2)
result: Mountaineering Equipment
Outdoor Protection

3.50 tail

Syntax:

tail (set_exp [, index_exp])

Description:

Returns the last "index_exp" elements of "set exp". The default for "index_exp" is 1.

Example:

tail(members([great_outdoors_company].[Products].[Products].[Product line]))
result: Personal Accessories

tail(members([great_outdoors_company].[Products].[Products].[Product line]),2)
result: Outdoor Protection
Personal Accessories

3.51 topCount

Syntax:

topCount (set_exp , index_exp , numeric_exp)

Desciption:

This function sorts a set according to the values of "numeric_exp" evaluated at each of the members of "set_exp", and returns the top "index_exp" members.

Example:

Based on a crosstab report using the intersection of [2006] and the default measure [Revenue] to determine numeric_exp.

topCount(set([Camping Equipment],[Golf Equipment],[Mountaineering Equipment]),2,[2006])

Revenue	*<#Function#>*	<#Camping Equipment#>	<#Golf Equipment#>	<#Mountaineering Equipment#>
<#2004#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2005#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2006#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>

result:

Revenue	*Camping Equipment*	*Mountaineering Equipment*	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	*$20,471,328.88*	*$0.00*	$20,471,328.88	$5,597,980.86	$0.00
2005	*$31,373,606.46*	*$9,642,674.54*	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	*$37,869,055.58*	*$11,248,676.06*	$37,869,055.58	$10,709,215.84	$11,248,676.06

Based on a list report using a direct reference to the [Revenue] measure for numeric_exp.

topCount([great_outdoors_company].[Products].[Products].[Product line],2,[Revenue])
result: Camping Equipment $89,713,990.92
Personal Accessories $31,894,465.86

3.52 topPercent

Syntax:

topPercent (set_exp , numeric_exp1, numeric_exp2)

Description:

This function is similar to topSum, but the threshold is "numeric_exp1" percent of the total.

This function works as follows: For a given set find the members whose sum percentage is greater than equal to a value based on a tuple

numeric_exp1 ranges from 0 to 100.

Example:

topPercent(set([Camping Equipment],[Golf Equipment],[Mountaineering Equipment]),40,[2006])

For the set of Camping Equipment, Golf Equipment and Mountaineering Equipment return the members whose percentage total are greater than or equal to 40% for the tuple 2006

Revenue	*<#Function#>*	<#Camping Equipment#>	<#Golf Equipment#>	<#Mountaineering Equipment#>
<#2004#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2005#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2006#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>

Revenue	*Camping Equipment*	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	*$20,471,328.88*	$20,471,328.88	$5,597,980.86	$0.00
2005	*$31,373,606.46*	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	*$37,869,055.58*	$37,869,055.58	$10,709,215.84	$11,248,676.06

topPercent(set([Camping Equipment],[Golf Equipment],[Mountaineering Equipment]),70,tuple([2006],[great_outdoors_company].[Measures].[Gross profit]))
result:

Revenue	*Camping Equipment*	*Golf Equipment*	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	*$20,471,328.88*	*$5,597,980.86*	$20,471,328.88	$5,597,980.86	$0.00
2005	*$31,373,606.46*	*$9,598,268.88*	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	*$37,869,055.58*	*$10,709,215.84*	$37,869,055.58	$10,709,215.84	$11,248,676.06

3.53 topSum

Syntax:

topSum (set_exp , numeric_exp1 , numeric_exp2)

Description:

This function sorts on "numeric_exp2", evaluated at the corresponding members of "set_exp", and picks up the topmost elements whose cumulative total is at least "numeric_exp1".

Example:

Revenue	*<#Function#>*	<#Camping Equipment#>	<#Golf Equipment#>	<#Mountaineering Equipment#>
<#2004#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2005#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>
<#2006#>	*<#1234#>*	<#1234#>	<#1234#>	<#1234#>

Based on a crosstab report

topSum(children([Products]),16000000,tuple([2006],[great_outdoors_company].[Measures].[Gross profit]))

result:

Revenue	*Camping Equipment*	*Personal Accessories*	*Golf Equipment*	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	*$20,471,328.88*	*$7,144,797.52*	*$5,597,980.86*	$20,471,328.88	$5,597,980.86	$0.00
2005	*$31,373,606.46*	*$10,955,708.04*	*$9,598,268.88*	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	*$37,869,055.58*	*$13,793,960.30*	*$10,709,215.84*	$37,869,055.58	$10,709,215.84	$11,248,676.06

3.54 tuple

Syntax:

tuple (member { , member })

Description:

Identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. Implicitly includes the current member from all dimensions not otherwise specified in the arguments. The current member of any dimension not specified in the evaluating context is assumed to be the default member of that dimension. The value of this cell can be obtained with the "value" function.

Example:

tuple([Mountaineering Equipment], [Fax])

Revenue	*Mountaineering Equipment Sales by Fax*
2004	*$0.00*
2005	*$662,612.52*
2006	*$557,716.86*

3.55 union

Syntax:

union (set_exp1 , set_exp2 [, ALL])

Description:

This function returns the union of 2 sets "set_exp1" and "set_exp2". The result retains duplicates only when the optional keyword ALL is supplied as the third argument.

Example:

union(set([Camping Equipment], [Golf Equipment]), set([Golf Equipment], [Mountaineering Equipment]))
result:

Revenue	Camping Equipment	Golf Equipment	Mountaineering Equipment
2004	$20,471,328.88	$5,597,980.86	$0.00
2005	$31,373,606.46	$9,598,268.88	$9,642,674.54
2006	$37,869,055.58	$10,709,215.84	$11,248,676.06

union(set([Camping Equipment], [Golf Equipment]), set([Golf Equipment], [Mountaineering Equipment]), all)
result:

Revenue	Camping Equipment	Golf Equipment	Golf Equipment	Mountaineering Equipment
2004	$20,471,328.88	$5,597,980.86	$5,597,980.86	$0.00
2005	$31,373,606.46	$9,598,268.88	$9,598,268.88	$9,642,674.54
2006	$37,869,055.58	$10,709,215.84	$10,709,215.84	$11,248,676.06

3.56 value

Syntax:

value (tuple)

Description:

Returns the value of the cell identified by a tuple. Note that the default member of the Measures dimension is the Default Measure.

Many times there will be an implicit value returned and the value function may be unnecessary. In the examples below it is possible to remove the value function and achieve the same results.

Example:

value(tuple([great_outdoors_company].[Years].[Years].[Year]->:[PC].[Years (Root)].[20040101-20041231],[great_outdoors_company].[Measures].[Revenue]))
result: $34,750,563.50

value(tuple([2004], [Camping Equipment] , [Revenue]))
result: $20,471,328.88

4 Appendix A – Function Grouping

4.1 Block

_firstFromSet
_remainderSet

4.2 Date

closingPeriod
lastPeriods
openingPeriod
parallelPeriod
periodsToDate

4.3 Family

ancestor
ancestors
children
cousin
descendants
firstChild
firstSibling
lastChild
lastSibling
parent
siblings

4.4 Hierarchy

hierarchy
level
levels
rootMembers

4.5 Member

currentMember
defaultMember
lag
lead
member
nextMember
prevMember

4.6 Set

emptySet
except
filter
generate
head
hierarchize
intersect
item

members
nestedSet
order
set
subset
tail
union

4.7 Top and Bottom

bottomCount
bottomPercent
bottomSum
topCount
topPercent
topSum

4.8 Value

caption
completeTuple
ordinal
roleValue
tuple

5 Deployment of Function Samples

The attached zip file is a deployment export file for Cognos 8 MR1 containing report samples of the functions listed above. Please follow the product documentation regarding the steps to import a deployment file into Cognos 8 and set up the sample "Great Outdoors Company" PowerCube required for these reports.