



Tip or Technique

Calculations that Span Dimensions

Product(s): Report Studio, Crosstabs, Dimensional
Expressions

Area of Interest: Reporting

Copyright

Your use of this document is subject to the Terms of Use governing the Cognos software products and related services which you have licensed or purchased from Cognos. The information contained in this document is proprietary information of Cognos Incorporated and/or its licensors and is protected under copyright and other applicable laws. You may use the information and methodologies described in this document 'as is' or you may modify them, however Cognos will not be responsible for any deficiencies or errors that result from modifications which you make. Copyright 2007 (c) Cognos Incorporated. All Rights Reserved.

You can print selected pages, a section, or the whole book. Cognos grants you a non-exclusive, non-transferable license to use, copy, and reproduce the copyright materials, in printed or electronic format, solely for the purpose of providing internal training on, operating, and maintaining the Cognos software.

This document is maintained by the Best Practices, Product and Technology team. You can send comments, suggestions, and additions to BestPractices-ProductandTechnology@cognos.com.



Contents

1	INTRODUCTION	4
1.1	PURPOSE	4
1.2	EXCLUSIONS AND EXCEPTIONS	4
2	EXAMPLE SHOWING ERROR	4
2.1	CURRENTMEMBER() FUNCTION TAKES A HIERARCHY AS A PARAMETER	5
2.2	PARENT() MAY NOT REPRESENT THE ROLLUP OF THE MEMBERS IN THE SET	5
3	MULTIPLE SETS APPROACH	6
4	MEMBER SET APPROACH	9
5	SUMMARY	13



1 Introduction

1.1 Purpose

This document describes two techniques to extend crosstab calculations across disparate dimensions or hierarchies within a dimension. Section 2 below shows a screenshot of what happens if you try to do this with traditional calculations.

These techniques will be useful for report developers that need to create calculations in crosstabs that span more than one dimension. They will find uses in building financial reports. The first technique relies upon the member summary function's ability to take multiple sets as parameters e.g. total(something within set set1, se2, set3 etc)

Whereas the alternate approach to resolving this is to use member sets. So the two techniques will be named "Multiple Set Approach" for the first approach, and "Member Set Approach" for the second approach.

This document will explain a feature of Report Studio that is not readily apparent or well documented to date.

1.2 Exclusions and Exceptions

This is a Report Studio technique for crosstabs only. The work for creating and testing this technique has been performed within the RTM release of Cognos 8.2.

2 Example showing Error

Sample With Error

Quantity sold	2004		2005		2006	
	Quantity sold	Percent of Level	Quantity sold	Percent of Level	Quantity sold	Percent of Level
Americas	210,046	37.4%	221,384	28.0%	256,348	29.7%
Asia Pacific	56,532	10.1%	94,252	11.9%	99,314	11.5%
Central Europe	154,888	27.6%	249,634	31.6%	261,492	30.3%
Northern Europe	67,474	12.0%	131,288	16.6%	146,766	17.0%
Southern Europe	72,766	13.0%	93,118	11.8%	100,052	11.6%
Total(Sales Territory1)	561,706	100.0%	789,676	100.0%	863,972	100.0%
Camping Equipment	181,822	#!Error	306,566	#!Error	377,846	#!Error
Golf Equipment	18,912	#!Error	38,692	#!Error	41,796	#!Error
Mountaineering Equipment	0	#!Error	139,562	#!Error	162,396	#!Error
Outdoor Protection	274,650	#!Error	171,182	#!Error	112,022	#!Error
Personal Accessories	86,322	#!Error	133,674	#!Error	169,912	#!Error
Total(Products1)	561,706		789,676		863,972	

The calculation of the Percent of Level data item in the above sample output is:

```
[Quantity sold] /  
tuple (parent (currentMember ([great_outdoors_company] . [Sales  
Territory] . [Sales Territory])), [Quantity sold])
```

As you can see since the Sales Territory hierarchy is referenced in the CurrentMember() calculation, it does not have the correct context to satisfy both the Products hierarchy and the Sales Territory hierarchy at the same time.

2.1 CurrentMember() function takes a hierarchy as a parameter

When traversing members in a crosstab in a given calculation, the CurrentMember() function is often used to provide context for the calculation. When disparate dimensions or hierarchies are stacked in the crosstab, the CurrentMember(hierarchy) component of the calculation only refers to one of the hierarchies thus giving an error condition in the crosstab for the other hierarchies.

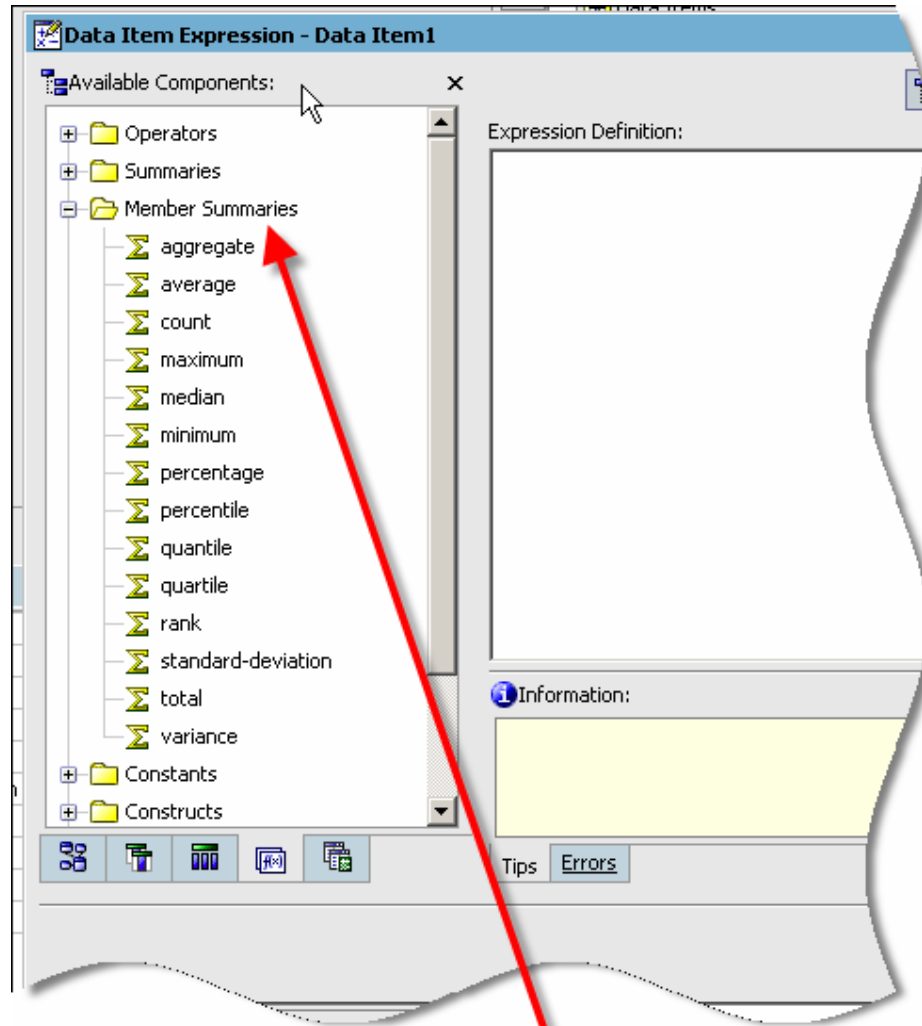
2.2 Parent() may not represent the rollup of the members in the set

When attempting to generate a percent of base calculation the parent member may not contain the appropriate summary for the set itself. There are two reasons why this might take place: the rollup at the parent is not the straight sum of the child members or the set that is being used on the rows is not the complete set of children for the parent member.

We will look at using member summary functions to build a more appropriate expression to define the denominator for our percentage.

3 Multiple Sets Approach

The built-in functions for Member Summaries are accessed from the following location in the Functions list of the Available Components pane of the Expression Editor:



Member Summaries for use in OLAP calculations

Each has a syntax similar to:

```
Total ( < CurrentMeasure | expr > within set set_expr { , set_expr } )
```

It's the ability to add multiple { , set_expr } to this function which allows this first technique to work.

Lets change our example to use the Total() Member Summary.

From

```
[Quantity sold] /
tuple (parent (currentMember ([great_outdoors_company] . [Sales
Territory] . [Sales Territory])), [Quantity sold])
```

To

```
[Quantity sold] /
total ( [Quantity sold] within set [Products1], [Sales
Territory1])
```

Here [Products1] and [Sales Territory1] are the data item references to the existing sets that we have defined on our rows.

This gives:

Example Corrected to use Total() Member Summary

Quantity sold	2004		2005		2006	
	Quantity sold	Data Item1	Quantity sold	Data Item1	Quantity sold	Data Item1
Americas	210,046	0.374	221,384	0.280	256,348	0.297
Asia Pacific	56,532	0.101	94,252	0.119	99,314	0.115
Central Europe	154,888	0.276	249,634	0.316	261,492	0.303
Northern Europe	67,474	0.120	131,288	0.166	146,766	0.170
Southern Europe	72,766	0.130	93,118	0.118	100,052	0.116
Total(Sales Territory1)	561,706	1.000	789,676	1.000	863,972	1.000
Camping Equipment	181,822	0.324	306,566	0.388	377,846	0.437
Golf Equipment	18,912	0.034	38,692	0.049	41,796	0.048
Mountaineering Equipment	0	0.000	139,562	0.177	162,396	0.188
Outdoor Protection	274,650	0.489	171,182	0.217	112,022	0.130
Personal Accessories	86,322	0.154	133,674	0.169	169,912	0.197
Total(Products1)	561,706	1.000	789,676	1.000	863,972	1.000

This of course can be formatted to be a percentage. However, we can make use of the `percentage()` Member Summary function to automatically take care of the percentage formatting

```
percentage([Quantity sold] within set [Product line], [Sales
territory])
```

This gives us:

Example Corrected to use the percentage() Member Summary

Quantity sold	2004		2005		2006	
	Quantity sold	percent of level	Quantity sold	percent of level	Quantity sold	percent of level
Americas	210,046	37%	21,384	28%	256,348	30%
Asia Pacific	56,532	10%	94,252	12%	99,314	11%
Central Europe	154,888	28%	249,634	32%	261,492	30%
Northern Europe	67,474	12%	131,288	17%	146,766	17%
Southern Europe	72,766	13%	93,118	12%	100,052	12%
Total(Sales territory)	561,706	100.0%	789,676	100.0%	863,972	100.0%
Camping Equipment	181,822	32.4%	306,566	38.8%	377,846	43.7%
Golf Equipment	18,912	3.4%	38,692	4.9%	41,796	4.8%
Mountaineering Equipment	0	0.0%	139,562	17.7%	162,396	18.8%
Outdoor Protection	274,650	48.9%	171,182	21.7%	112,022	13.0%
Personal Accessories	86,322	15.4%	133,674	16.9%	169,912	19.7%
Total(Product line)	561,706	100.0%	789,676	100.0%	863,972	100.0%

This approach has the benefit that the drill down and up behaviour on the hierarchy does not need to be modified. i.e. product line to product type to product name.

The drawback to this approach is that listing multiple sets within a member summary function is a request to generate a cross-product of the two sets. There are several issues with this approach. The first is that a cross-product cannot currently be performed between different hierarchies of the same dimension. When attempting to do so you will receive a message similar to the following:

```
OP-ERR-0045 Unable to support query items from two or more
hierarchies of a single dimension within a query:
hierarchy1="[Sales Territory].[]"; hierarchy2="[Sales
Territory].[By Staff name]".
```

The second potential issue is that if you are using a DMR model the relationships to the fact table may be defined with inner joins. This means that a sparse fact table may not have matching records for the intersections of both dimensions. In such cases this may result in a smaller data set than would be encountered when querying from the respective dimensions individually.

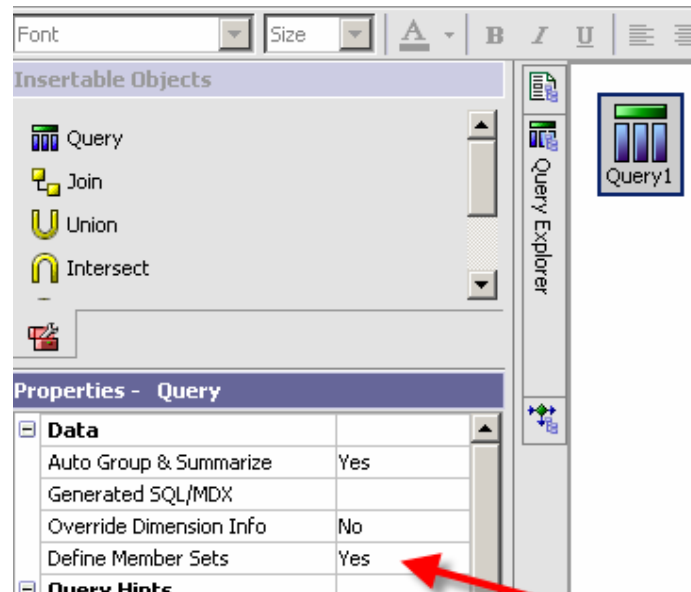
A third consideration occurs in processing such a cross-product of the two sets. When the sets are large there may be additional processing required to generate the product of the two sets. For example, if the first set contains 10 members and the second set contains 20 members then the intermediate cross-product required in the member summaries will be a set of 200 tuples. When working with many sets on an edge this cross-product processing requirement can become an even greater impact on performance.



4 Member Set Approach

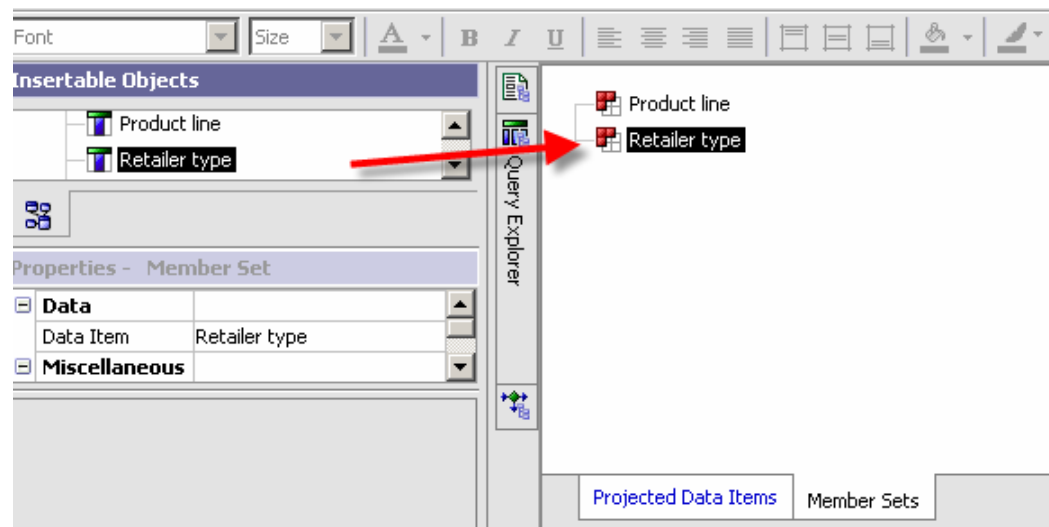
This approach to solving the problem, relies on Report Studio's ability to define Member Sets. A Member Set is simply an alias for an existing set within the query.

To define Member Sets for a Query object,

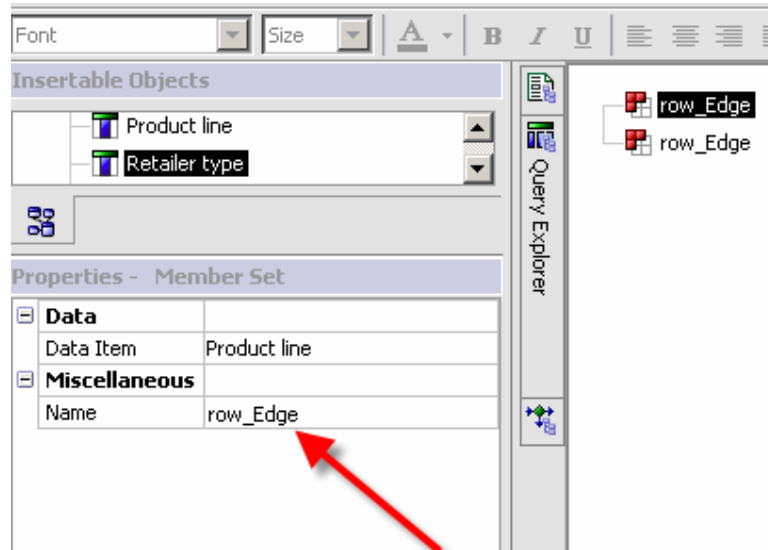


Change this query attribute to Yes

Then click on the member sets tab and drag the data items that appear in the rows of our crosstab:



Under the Member Set properties, change the name attribute to the same "Common Alias". In this case "row_Edge" is being used as the new, common, alias for the two different sets.



Create a "Common Alias"

The name of the new alias is not important, but it has to be identical to others that appear in the crosstab row or column that you want the calculation to span and it should not duplicate the name of an existing data item in the query. The distinction from the other data item names prevents confusion when referencing the alias in expressions. This new common alias is what we will refer to in our calculation.

The calculation, now becomes

```
percentage( [Quantity sold] within set [row_Edge])
```

Change the Solve order on the calculation, so that it is calculated after the row totals. The default solve order for the row totals is 1. Assign a solve order to the calculation that is greater than 1 to obtain the correct rollup of the percentage calculation.

The screenshot shows the 'Properties - Data Item' window for the 'percent of level' data item. The 'Solve Order' is set to 10. To the right, the 'Data Items' list shows 'percent of level' selected, along with other items like 'Year', 'Product line', and 'Retailer type'.

Properties - Data Item	
Data Item	
Name	percent of level
Label	
Expression	percentage(curren...
Aggregate Function	Automatic
Rollup Aggregate Function	Automatic
Pre-Sort	Don't sort
Detail	Automatic
Calculation Intersection	Show value
Solve Order	10

The report is now like this:

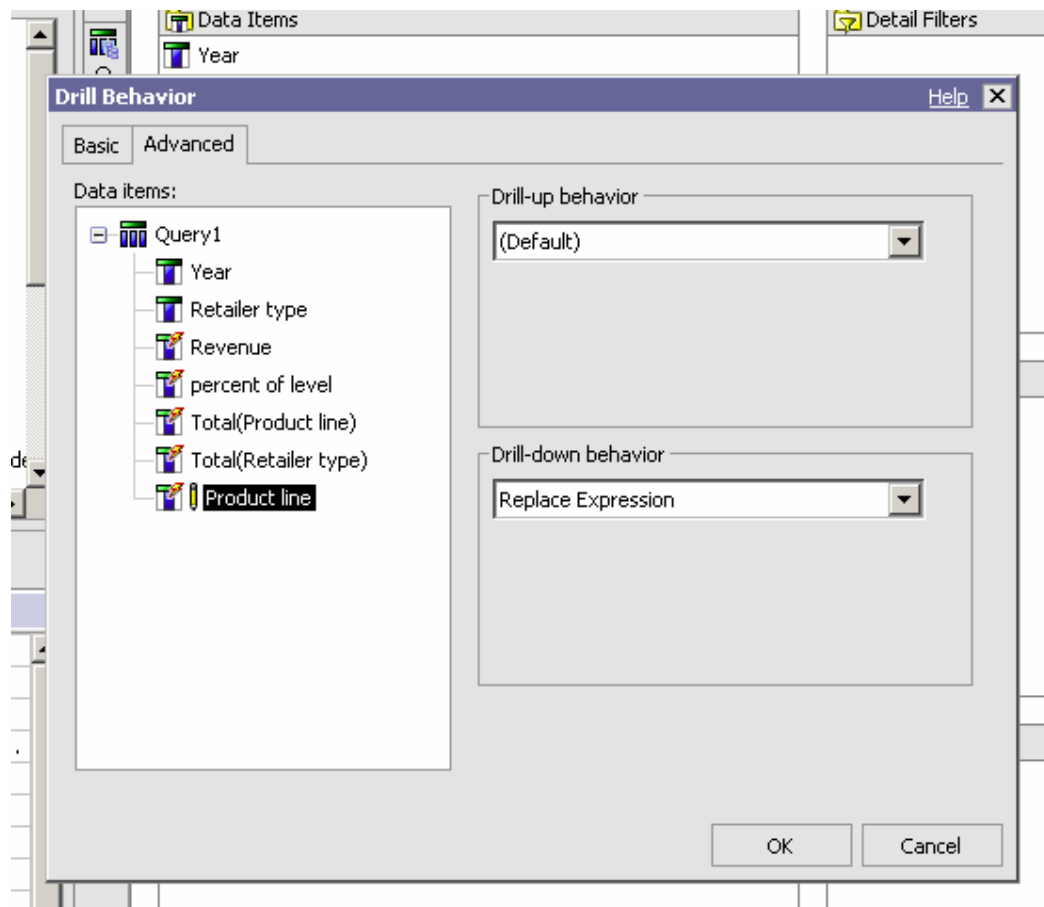
Member Set Approach

Revenue	2004		2005		2006	
	Revenue	percent of level	Revenue	percent of level	Revenue	percent of level
Camping Equipment	\$20,471,328.88	58.9%	\$31,373,293.14	50.2%	\$37,869,368.90	51.0%
Golf Equipment	\$5,597,980.86	16.1%	\$9,580,958.70	15.3%	\$10,726,526.02	14.4%
Mountaineering Equipment	\$0.00	0.0%	\$9,642,674.54	15.4%	\$11,248,676.06	15.1%
Outdoor Protection	\$1,536,456.24	4.4%	\$988,230.64	1.6%	\$646,428.04	0.9%
Personal Accessories	\$7,144,797.52	20.6%	\$10,955,708.04	17.5%	\$13,793,960.30	18.6%
Total(Product line)	\$34,750,563.50	100.0%	\$62,540,865.06	100.0%	\$74,284,959.32	100.0%
Department Store	\$6,946,116.92	20%	\$10,080,403.44	16%	\$11,337,759.04	15%
Direct Marketing	\$1,155,880.62	3%	\$2,924,317.20	5%	\$3,396,560.06	5%
Equipment Rental Store	\$345,612.74	1%	\$369,296.70	1%	\$542,870.22	1%
Eyewear Store	\$135,366.94	0%	\$142,456.48	0%	\$166,381.94	0%
Golf Shop	\$2,886,829.76	8%	\$5,621,018.62	9%	\$6,492,519.78	9%
Outdoors Shop	\$11,566,932.70	42%	\$30,245,152.22	48%	\$35,741,462.70	48%
Sports Store	\$6,018,600.50	17%	\$10,065,090.26	16%	\$12,943,493.34	17%
Warehouse Store	\$2,695,223.32	8%	\$3,093,130.14	5%	\$3,663,912.24	5%
Total(Retailer type)	\$34,750,563.50	100.0%	\$62,540,865.06	100.0%	\$74,284,959.32	100.0%

Note: Drill down not available.

Drill down is not enabled at this point because a member set has been defined for the sets on the row edge of our crosstab. Member sets are the basis of some advanced drill behaviours and once they are defined a drill behaviour must be assigned to the data item to enable drill up and drill down.

In Report Studio, select Drill Behavior from the Data menu and then select the Advanced Tab.



Choose the data items in the cross tab row edge change their Drill-down behaviour to "Replace Expression". This has the same effect as the default drill behaviour when drilling down.

For more info on defining drill behaviour obtain this document called "Drill Up/Down" from the Proven Practice web site.

http://support.cognos.com/supported/tti/public/docs/cognos_pp_reporting_drill_up-down.pdf?lid=cognos_pp_reporting_drill_up-down

So, with this approach there is an extra step to redefine drill behaviours for interactive HTML output. Currently there is also limit to including multiple hierarchies of the same dimension as the member sets for the query. When using multiple hierarchies from the same dimension you will receive a message that reads something like:

PCA-ERR-0028 The query cannot specify more than one hierarchy for "Sales Territory" dimension.



At this time the issue can be resolved by adding the row totals into the “Common alias” that we have defined for the sets on the rows of the crosstab. Assigning the Totals to the “row_Edge” alias in our member sets will ensure that all the sets on the edge are being parsed appropriately. However, the member summary functions require a set to be defined for the “within set” clause and the totals that are on the crosstab rows will only produce a single summary member. To ensure that a set is always referenced in the member summary function it is necessary to modify the expression as follows.

Original expression:

```
percentage([Quantity sold] within set [row_Edge])
```

Modified expression:

```
percentage([Quantity sold] within set set([row_Edge]))
```

This change will ensure that when the single member of the Total row summary is being evaluated within the expression it will be treated as a set (containing a single member) rather than as a single member object alone.

However, the processing of this approach does not require a cross-product and can result in performance improvements over the previous technique when using large data sets.

5 Summary

Based on these two approaches you can generate any number of complex report layouts to present data from multiple dimensions and hierarchies using a single query. Using a single query will facilitate alignment of objects on your report and allow for easier page breaking as the data items will be controlled by a common crosstab container.

Using the cross-product of the two sets to generate the summaries used in the new calculation provides an intuitive approach to designing calculations and allows the current drill behaviours to be preserved without modification.

The member set approach allows for more straight-forward queries to be submitted to the data source but requires that drill behaviours be assigned to the data items to redefine the ability to drill up and down on the sets in the rows.

Regardless of the approach, both approaches increase the power of the reporting techniques at your disposal.

