Optimizing TMVGate Performance by modifying Cube View(s)

Technical Note: TA2019003

Contents

1.0 Overview

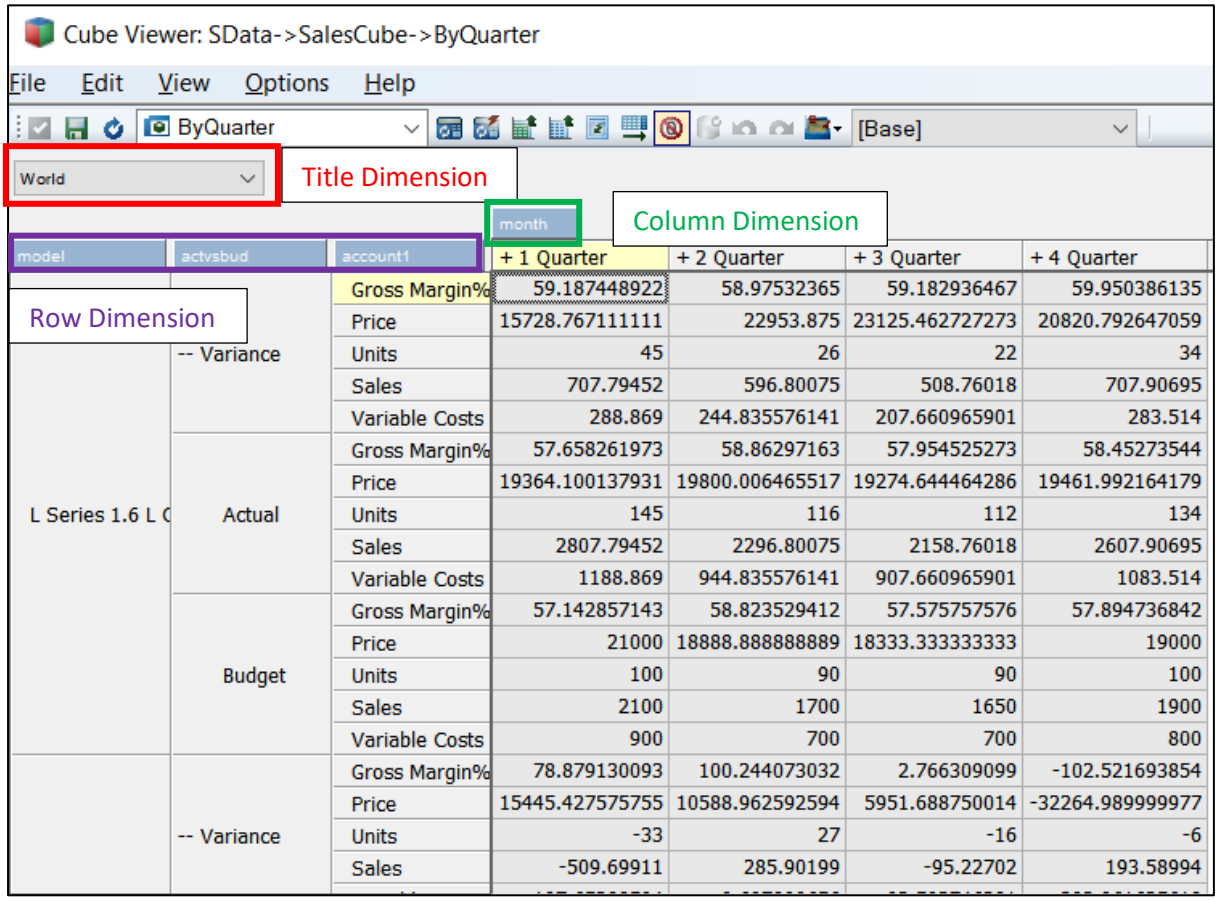
When dealing with millions of data cells to be extracted from Planning Analytics/TM1 using TMVGate, the layout of the Cube View has a significant impact to the performance and the initial memory utilization.

This technical document illustrates an optimization technique which can improve performance and reduce the initial memory requirement for large data extraction using TMVGate.

Note that this technique is not only useful for large data sets, but it will also be applicable to smaller data sets in the range of hundred thousand entries.

2.0 Optimizing Cube View with a balance rows and columns

A typical TM1 Cube view comprises the Title, Row and Column dimensions as shown below:



			+ 1 Quarter	+ 2 Quarter	+ 3 Quarter	+ 4 Quarter
		Gross Margin%	59.187448922	58.97532365	59.182936467	59.950386135
		Price	15728.767111111	22953.875	23125.462727273	20820.792647059
	-- Variance	Units	45	26	22	34
		Sales	707.79452	596.80075	508.76018	707.90695
		Variable Costs	288.869	244.835576141	207.660965901	283.514
		Gross Margin%	57.658261973	58.86297163	57.954525273	58.45273544
		Price	19364.100137931	19800.006465517	19274.644464286	19461.992164179
L Series 1.6 L C	Actual	Units	145	116	112	134
		Sales	2807.79452	2296.80075	2158.76018	2607.90695
		Variable Costs	1188.869	944.835576141	907.660965901	1083.514
		Gross Margin%	57.142857143	58.823529412	57.575757576	57.894736842
		Price	21000	18888.888888889	18333.333333333	19000
	Budget	Units	100	90	90	100
		Sales	2100	1700	1650	1900
		Variable Costs	900	700	700	800
		Gross Margin%	78.879130093	100.244073032	2.766309099	-102.521693854
		Price	15445.427575755	10588.962592594	5951.688750014	-32264.989999977
	-- Variance	Units	-33	27	-16	-6
		Sales	-509.69911	285.90199	-95.22702	193.58994

Notice that in the above example, the Row Dimensions comprise 3 dimensions. Assuming the number of elements for these 3 dimensions are 1,000 respectively, the potential total number of rows will be 1,000*1,000*1,000 = 1,000,000,000.

The column dimension comprises only 1 dimension with 4 columns.

During TMVGate's initial processing of the axes, the maximum records (maximum records of either Rows or Columns) to be handled will be 1,000,000,000.

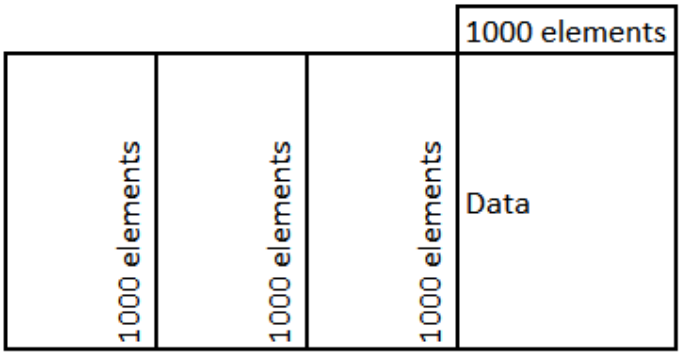
To improve efficiency, we modify the cube view by switching one of the Row dimensions over as a Column dimension. This will result in a significantly lower total number of rows which will be 1000*1000 = 1,000,000.

The total number of columns will be 4*1000 = 4,000.

This results in a 1000-fold reduction compared to the initial Cube View layout.

This optimized cube view will now have 2 dimensions on the Rows and 2 dimensions on the Columns. The data to be extracted is not affected as the number of cells remain unchanged. But the initial processing of the axes information will be significantly faster with lower memory utilization.

In summary, when you have a Cube view with the following layout as an example:



You can improve the resource (memory) utilization and performance by changing the Cube View to: