KIMBALL GROUP KIMBALL UNIVERSITY

www.kimballgroup.com Number 56, July 12, 2004

Kimball Design Tip #56: Dimensional Modeling for Microsoft Analysis Services

By Joy Mundy

Over the past few years, an increasing number of our students and clients have been asking us about Microsoft SQL Server 2000 Analysis Services (AS). Analysis Services, as a server based OLAP product, fits very well with the dimensional modeling techniques that the Kimball Group has long espoused. But is that fit perfect? Are there things you should be aware of as you're designing a dimensional data model that will be accessed from an AS cube? Of course there are!

The first and strongest connection between an Analysis Services cube and a relational dimensional model is that the cube should be built from a relational dimensional model. The relational dimensional source can be in any relational database, though extracts perform better from some databases than others.

Dimensional models in a relational platform and AS cubes both have dimension and facts; both love surrogate keys; both emphasize the importance of using well thought out hierarchies to assist navigation through the dimensional data. Conformed dimensions in the Kimball vernacular are called shared dimensions in AS. More advanced techniques like junk dimensions, dimension roles and factless fact tables all translate smoothly.

The most important difference between the dimensional world on a relational platform and Analysis Services OLAP is how dimensions behave. The more familiar and comfortable you are with relational-based dimensions, the weirder you'll find AS dimensions. The problem is that they look so similar to each other on first glance, but actually behave quite differently. What we consider a single relational dimension, like customer, typically becomes several dimension-like structures in the cube. Any attribute or hierarchy you want to slice on must have its own dimension in the cube. For example, if you want to compare sales by gender by geographic region, you would need a gender dimension and geography dimension hierarchy in the cube, where these could both be part of the Customer dimension in the relational-based dimensional structures.

In the example just described, you don't have to change your relational dimensional model. In AS, you would define a geography hierarchy in the Customer dimension. You would first create the Gender attribute as a member property in the Customer dimension, and then explicitly convert that member property to a virtual dimension. This conversion is simple to do, requiring only a few mouse clicks, but it seems strange and unnecessary to those of us who are deeply familiar with the way the relational model behaves. Microsoft has recognized this issue and announced that dimensions will behave much more like relational dimensions in the next version of AS.

We'll quickly summarize some other differences and common "gotchas" you may encounter when working with Analysis Services.

The Good:

- Parent-Child dimensions are sometimes easier to manage and query in AS than relational dimensional. In relational-based dimensional, you would navigate a ragged hierarchy by using a bridge table. In Analysis Services, you would simply model the relational dimension with a parent-child structure and create the AS dimension as "parent-child."
- Analysis Services holds all dimension members in memory, so it's best to stay under 5-7 million

total members (across all dimensions) on the 32-bit platform. You can stretch that number, but with reasonably priced 64-bit systems available, it hardly seems worth the trouble.
- You can define multidimensional expressions to calculate anything on any level of the cube. A powerful example of a calculation is defining inventory balances to be semi-additive across time. On the downside, it requires some thought and effort to define complex calculations correctly.
- Incremental cube processing requires the ability to identify the new rows added to the fact table. Tagging the fact rows with an audit dimension or other metadata allows them to be filtered for incremental cube processing.
- Type 2 slowly changing dimensions are processed smoothly by AS.
- Analysis Services dimensions can be built from either a star or snowflake schema. If queries go through AS, it doesn't matter much how the dimension tables are structured on the relational side, though stars are generally easier to maintain. Of course, if queries from other tools will be issued directly against the relational-based dimensional model, stars typically deliver better ease of use and query performance than snowflakes. As in the relational dimensional model, AS dimensions must correspond to the level of granularity. For example, forecast data at the brand level will require a brand-level dimension while the actual data may be at the SKU level.

The Not So Good:
- Type 1 slowly changing dimension dimensions can be a problem if the column being updated is part of a dimension hierarchy. If the updated Type 1 attribute is simply a member property or a virtual dimension, there's no issue. The problem with restating history on a hierarchical attribute is the aggregates. Analysis Services faces exactly the same problem that you do if you were to managing the aggregates by hand in the relational database. There are several approaches which are discussed in the Performance Guide reference listed below.
- Many to many dimensions are ugly in AS 2000. We've seen people use a variety of techniques to make them work, but none of them is particularly appealing.
- There is no way, short of fully reprocessing a cube partition, to update a fact row in an AS cube. If your data volumes are small, you can fully reprocess the entire cube. With large data volumes, you may be able to isolate changeable rows in a partition (perhaps a partition for the most recent 30 days) and fully reprocess only that partition. Partitions require the expensive Enterprise Edition.

Here are some pointers to Microsoft-published Analysis Services content:
Analysis Services Performance Guide
Analysis Services Operations Guide
Creating Large-Scale, Highly Available OLAP Sites
Advantages of 64-bit to SQL Server 2000 Enterprise Edition BI Customers
SQL Server 2000 (64-bit) Analysis Services: Why Migrate and What to Expect