



CHAPTER

1

OLAP Introduction and Overview

<i>What Is OLAP?</i>	1
<i>Data Storage and Access</i>	1
<i>Benefits of OLAP</i>	2
<i>OLAP Variations</i>	3
<i>What Is a Cube?</i>	3
<i>Understanding the Cube Structure</i>	4
<i>Dimensions, Levels and Hierarchies</i>	4
<i>Members</i>	4
<i>Measures</i>	4
<i>Calculated Measures</i>	5
<i>Aggregations</i>	5
<i>Why You Should Use Cubes</i>	5
<i>Cube Usage and Storage Space Reduction</i>	5
<i>Multi-Threading Capabilities</i>	5
<i>Easy Setup and Maintenance</i>	6
<i>Data Management: Choosing Your Own Tool</i>	6

What Is OLAP?

Online Analytical Processing (OLAP) is a technology that is used to create decision support software. OLAP enables application users to quickly analyze information that has been summarized into multidimensional views and hierarchies. By summarizing predicted queries into multidimensional views prior to run time, OLAP tools provide the benefit of increased performance over traditional database access tools. Most of the resource-intensive calculation that is required to summarize the data is done before a query is submitted.

Data Storage and Access

Decision makers are asked to make timely and accurate decisions that are based on the past performance and behavior of an organization as well as on future trends and directives. To make effective business decisions, business analysts must have access to the data that their company generates and responds to. This access must include timely queries, summaries, and reviews of numerous levels and combinations of large, recurrent amounts of data. The information that business analysts review determines the quality of their decisions.

Organizations usually have databases and data stores that maintain repeated and frequent business transaction data. These data storage systems provide simple yet detailed storage and retrieval of specific data events. However, these systems are not

well suited for analytical summaries and queries that are typically generated by decision makers. For decision makers to reveal hidden trends, inconsistencies, and risks in a business, they must be able to maintain a certain degree of momentum when querying the data. An answer to one question usually leads to additional questions and review of the data. Simple data stores do not successfully support this type of querying.

A second type of storage, the data warehouse, is better suited for this. Data is maintained and organized so that complicated queries and summaries can be run. OLAP further organizes and summarizes specific categories and subsets of data from the data warehouse. This results in a robust and detailed level of data storage with efficient and fast query returns. SAS OLAP cubes can be built from either partially or completely denormalized data warehouse tables. Stored, precalculated summarizations called *aggregations* can be added to the cube to improve cube access performance. Aggregations can either be pre-built relational tables, or you can let the cube create its own optimized aggregates.

Benefits of OLAP

The ability to have coherent, relevant, and timely information is the reason OLAP has gained in popularity. OLAP systems can help reveal evasive inconsistencies and trends in data that might not have been seen before. OLAP users can intuitively search data that has been consolidated and summarized within the OLAP structure. In addition, OLAP tools allow for tasks such as sales forecasting, asset analysis, resource planning, budgeting, and risk assessment. OLAP systems also provide the following benefits:

- fast access, calculations, and summaries of an organization's data
- support for multiple user access and multiple queries
- the ability to handle multiple hierarchies and levels of data
- the ability to presummarize and consolidate data for faster query and reporting functions
- the ability to expand the number of dimensions and levels of data as a business grows

To fully understand the benefits of OLAP and the details of its effective implementation, it helps to examine the technology from two perspectives— that of the users, and that of the information technology (IT) administrators who are responsible for OLAP implementation. The users, typically business analysts and executives, expect the data to be organized according to categories that reflect the way they think about the enterprise. For IT administrators, OLAP can present a long list of technical issues, including these concerns:

- storage requirements and associated costs
- client and server capabilities
- maintenance activities such as update and backup
- performance considerations such as the amount of time that is required to build a multidimensional model
- the ability of the OLAP solution to integrate with current or planned data warehouse strategies and architectures
- security requirements for cube data

The SAS OLAP Server and SAS OLAP Cube Studio provide resources and functionality to address these concerns. When building SAS OLAP cubes, you can perform functions and specify settings that affect the following:

- cube aggregation storage and query performance
- cube dimension security and identity-driven security

- updates of cube data
- maintenance of cubes (such as adding calculated members or changing an OLAP schema.)

Because SAS OLAP is a component of the SAS Intelligence Platform, it works in conjunction with other SAS applications to provide an overall solution to the access and maintenance of a company's data.

OLAP Variations

OLAP technology can be further defined by the methods for storing and accessing data and by the performance of queries against that data. SAS OLAP supports three different variations of OLAP technology:

MOLAP (multidimensional OLAP)

is a type of OLAP that stores summaries of detail data (aggregates) in multidimensional database structures. MOLAP cubes are most suited for slicing and dicing of data and are used when performance and query speed is critical. For further information on MOLAP, see the topic “MOLAP Aggregation Storage” on page 28.

ROLAP (relational OLAP)

is a type of OLAP in which multidimensional data is stored in a relational database such as a SAS table or an ORACLE table. ROLAP is more scalable than other OLAP types and handles extensive amounts of data well. Although performance can be somewhat slow, ROLAP is limited only by the size of the relational database it is identified with. For further information on ROLAP, see the topic “ROLAP Aggregation Storage” on page 28.

HOLAP (hybrid OLAP)

is a type of OLAP in which relational OLAP (ROLAP) and multidimensional OLAP (MOLAP) are combined. In HOLAP, the source data is usually stored using a ROLAP strategy, and aggregations are stored using a MOLAP strategy. It combines the best features of both ROLAP and MOLAP. This combination usually results in the smallest amount of storage space. In HOLAP, aggregates can be precalculated and can be linked into a hybrid storage model.

What Is a Cube?

One of the advantages of OLAP is how data and its relationships are stored and accessed. OLAP systems house data in structures that are readily available for detailed queries and analytics. Cubes are central to the OLAP storage process.

A *cube* is a set of data that is organized and structured in a hierarchical, multidimensional arrangement. The cube is usually derived from a subset of a data warehouse. Unlike relational databases that use two-dimensional data structures (often in the form of columns and rows in a spreadsheet), OLAP cubes are logical, multidimensional models that can have numerous dimensions and levels of data. Also, an organization typically has different cubes for different types of data.

One of the challenges of OLAP cube data storage and retrieval is the growth of data and how that growth affects the number of dimensions and levels in a cube hierarchy. As the number of dimensions increases over time, so does the number of data cells on an exponential scale. To maintain the efficiency and speed of the OLAP queries, the cube data is often presummarized into various consolidations and subtotals (aggregations).

Note: The SAS OLAP Server term *cube* is synonymous with the terms *hyper-cube* and *multi-cube*. \triangle

Understanding the Cube Structure

A SAS OLAP cube stores data in a method that enables fast retrieval of summarized data. Data summarization in this context means condensing large numbers of detail records into meaningful numbers such as counts, sums, averages, or other statistical measures. The structure of a cube is hierarchical in nature and is derived from the associations between the different columns and rows of data in a data source. SAS OLAP cubes are comprised of dimensions, levels, hierarchies, members, and member properties. This structure enables you to easily select data subsets and navigate the cube structure when querying the cube.

Dimensions, Levels and Hierarchies

SAS OLAP cubes organize data in a hierarchical arrangement, according to dimensions and measures. *Dimensions* group the data along natural categories and consist of one or more levels. Each level represents a different grouping within the same dimension. For example, a time dimension can include levels such as years, months, and days. Or an organization dimension of a bank's customer service centers can include levels such as branches, states, and regions.

Levels are organized into one or more *hierarchies*, typically from a coarse-grained level (for example, Year) down to the most detailed one (for example, Day). The individual category values (for example, 2002 or 21Jan2002) are called *members*.

A dimension can also have *multiple hierarchies* to provide different sequences of groupings. For example, a "Time" dimension can have a "Fiscal Year" hierarchy and a "Calendar Year" hierarchy.

Members

Each combination of values within a dimension is called a member. Some examples of members are shown here.

[Time].[2003]

[Time].[2004].[January]

[Time].[2004].[February].[12th]

For each dimension, there is also the special member called the *ALL member*, which represents the total for all members (for example, [Time].[All Time]). Not all categorical data attributes need to become a member of a hierarchy level. Some grouping information is needed only as additional information for a member or for applying subsets to data. These attributes can be loaded into *member properties*. Member properties can be associated with any level in a hierarchy.

Measures

Measures are the cube data values that are summarized and analyzed. A measure is the combination of a numeric input column with a roll-up rule or statistic. Measures are loaded from the data source that you summarize from. One input column can load one or more measures. For example, you can create the measures "Sum of Amount" and "Maximum of Amount" from the input column "Amount."

Calculated Measures

Not all measures are directly derived from input columns. You can create *calculated measures*, which are formulas that are based on the values of other measures. Calculated measures are designed and stored with the cube.

Aggregations

An aggregation is a summary of detail data that is stored with or referred to by a cube. They are the basis for fast response to data queries in OLAP applications. An aggregation is possible at each intersection of a level of one or more dimensions. Any combination of dimension levels can become a stored aggregation, as long as it is appropriate within the defined hierarchies. One of the major factors that influences query response time is which aggregations you create and use to query your cube. The aggregations that are being stored with the cube affect cube build time, the absolute cube file size, SAS OLAP Server CPU usage, and query response times. As a result, determining and building your cube aggregations is a crucial component of good cube design.

Why You Should Use Cubes

SAS cubes are designed to offer efficient data storage, fast data access, easy data maintenance, and flexibility in data management. The following sections explore cubes and multidimensional storage.

Cube Usage and Storage Space Reduction

While cubes are the format of choice to guarantee fast query response times against your data warehouse, SAS OLAP cubes are also often a very space-efficient choice for data storage. In many cases, a basic cube without additional aggregations can be smaller than the input data because the process of creating the cube consolidates records. SAS OLAP cubes use the hierarchy information for efficient aggregations storage. SAS OLAP cubes also deal efficiently with data sparsity by using virtual placeholders for empty cells. This removes the need for any physical representation of empty cells. A good rule of thumb is, the larger your input data, the greater the storage gain by loading data into a cube.

Multi-Threading Capabilities

Loading data into cubes and executing queries against the cube take advantage of the multi-threading capabilities of your server machine. Aggregations are created in parallel at cube build time. The creation of individual aggregations takes advantage of the Parallel Group-By capabilities of the SAS data engine. At query execution, the multi-threading capabilities of your server machine are fully used to concurrently serve queries by multiple users. Both query evaluation and data access are executed in parallel. To further increase query performance and reduce disk access, you can allocate additional memory on your server to be used for an in-memory aggregation cache.

Easy Setup and Maintenance

A cube is the physical representation of your logical dimensional model. The tools that are provided to update and maintain the cube reflect the multidimensional model, which makes both setup and maintenance of your cube as intuitive as possible. SAS Management Console, a Web-based administrator interface, enables you to set up and manage OLAP servers. SAS OLAP Cube Studio provides the workspace and cube designer tools that you need to create and maintain cubes. You can also use the SAS OLAP procedure to create and maintain cubes in a batch environment.

Data Management: Choosing Your Own Tool

If you create your own aggregations by using data management tools such as SQL, PROC SUMMARY, or the tools of your preferred relational database management system (RDBMS), then you can link those aggregations to your cubes without replicating the data within the cube. Any queries against those aggregations are executed by the appropriate SQL engine, and take advantage of any capabilities that engine might have. This allows you the flexibility to use the data management tools of your choice. It also allows you to distribute your data for your cube aggregations across multiple database systems, servers, and platforms. If you choose to let the cube builder create the aggregations, then you can control where to store the data and index files for each aggregation.