

## CHAPTER

## 2

## Introduction to MDDBs

---

<i>What Is OLAP?</i>	3
<i>What Is SAS/MDDB Server Software?</i>	4
<i>What Is an MDDB?</i>	4
<i>Understanding the MDDB Structure</i>	5
<i>How Can I Use MDDBs?</i>	7
<i>Why Should I Use MDDBs?</i>	8

---

### What Is OLAP?

Online Analytical Processing (OLAP) is a technology used in creating decision support software that allows 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 can provide the benefit of increased performance over traditional database access tools: a great deal of the complex calculation that is required to summarize the data is done long before a query is submitted.

In order to fully understand the benefits of OLAP and the details of its effective implementation, it helps to examine the technology from two perspectives: (1) from that of the users and (2) from that of the information technology (IT) administrators who are responsible for OLAP implementation. The users, typically business managers and analysts, see OLAP as a way to gain rapid access to business data. They expect that the data will be organized according to categories that reflect the way in which they think about the enterprise. For IT administrators, OLAP can present a long list of technical issues, including concerns about

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

The challenge for IT administrators charged with implementing OLAP is to find a solution that will both supply the functionality that users expect and address these technical considerations. SAS/MDDB Server software provides all the functionality expected by those who will use the system, while meeting all the technical requirements of IT. For users, SAS/MDDB Server provides fast access to large amounts of shared data through a wealth of OLAP client capabilities. IT administrators can use SAS/MDDB Server to implement an OLAP solution that allows them to seamlessly integrate MDDBs with transactional systems, existing warehouse data, and powerful client applications.

This document details the capabilities and functionality of the SAS/MDDB Server and explains how you can implement MDDBs as part of your organization's OLAP solution.

---

## What Is SAS/MDDB Server Software?

SAS/MDDB Server software enables users to create and manage MDDBs to provide fast and easy data access by tools such as multidimensional data viewers. Used as a data summarization, storage, and retrieval mechanism for an organization's OLAP solution, MDDBs enable better decision making by giving business users quick, unlimited views of multiple relationships in large quantities of summarized data at any cross section of business dimensions. A business dimension can be any vision of the data that is useful in making business decisions—for example, time, geography, or product.

Taking advantage of a graphical user interface or a programming interface, you can use SAS/MDDB Server software to quickly and easily build, maintain, and optimize multidimensional data. For example, you can

- use warehouse data to design multidimensional data models based on the least amount of summarization, the most amount of summarization, or projections of the summarizations most likely to be requested by users
- build multidimensional data structures for interactive or batch submission
- change the way that multidimensional data is structured
- control which statistics are stored with which multidimensional data structures
- control the balance between storage and performance
- perform incremental updates.

---

## What Is an MDDB?

A multidimensional database, or MDDB, is a specialized storage facility that allows data to be pulled from a data warehouse or other data sources for storage in a matrix-like format. The process of building an MDDB summarizes the raw data; the data stored in the MDDB is thus said to be *presummarized*. The MDDB enables users to quickly retrieve multiple levels of presummarized data through a multidimensional view. An MDDB is *not* a SAS data file; instead of the traditional structure used for SAS tables, an MDDB stores its data as an *NWAY cube* and zero or more *subcubes*.

An *NWAY cube* is a multidimensional data model that specifies all the classification variables that can be used to define *crossings*. A crossing is a definition of one or more classification variables that exist in the data from which the MDDB is created. Each crossing represents a grouping on which summary statistics could be calculated.

While an MDDB can consist of an *NWAY cube* only, it is usually created with one or more subcubes. Subcubes are created using simple statements of hierarchy that define one or more additional specific crossings derived from the *NWAY cube*. Subcubes are built to enhance reporting speed and are projections of the most likely aggregates that business managers and other users will expect to see. If a subcube does not exist for a particular aggregate query, that is, if no subcube defines the exact crossings required to answer the query, the aggregate data will be derived from the smallest subcube that can provide the data. If no subcube can provide the data, it is derived from the *NWAY cube*.

By default, a minimum number of statistics are stored in the MDDB. During creation of the MDDB, you can specify up to 8 statistics for each analysis variable. Depending on which of the 8 statistics are stored, up to 13 additional statistics can be calculated by SAS/MDDB Server software at run time, allowing for up to 21 available statistics. For more information, see “Stored and Derived Statistics” on page 15.

Other characteristics common to all SAS MDDBs include the following:

- MDDBs can be created interactively or in batch.
- There is no practical limitation on the number of subcubes/crossings.
- There is no practical limitation on the number of analysis variables.
- MDDBs can be updated incrementally.

---

## Understanding the MDDB Structure

The structure of an MDDB, including the NWAY cube and its subcubes, is best illustrated through an example.

The SAS table from which an MDDB is created is called the base table. For this example, assume that our base table contains the following fields:

- classification variables: YEAR, MONTH, PRODUCT
- analysis variables: SALES, RETURNS.

The base table might look like this:

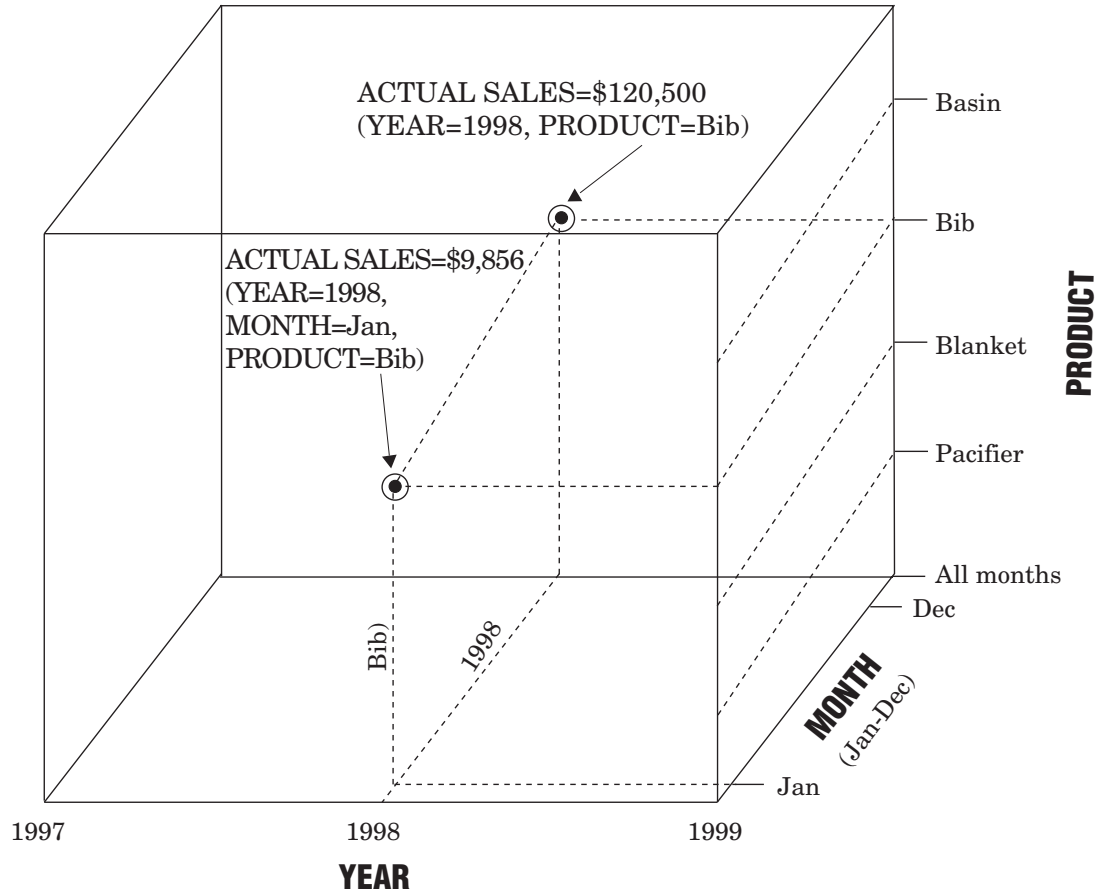
**Table 2.1** Example Base Table

YEAR	MONTH	PRODUCT	SALES	RETURNS
1999	Jan	Bib	\$10396	\$1281
1999	Jan	Blanket	\$11241	\$1145
1999	Feb	Bib	\$9823	\$789
1999	Feb	Blanket	\$14605	\$1358
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
1998	Jan	Bib	\$9846	\$673
1998	Jan	Blanket	\$10263	\$1183
1998	Feb	Bib	\$10793	\$978
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

When you define your NWAY cube, you will include all the classification variables, thus defining a crossing for each combination of classification variable values for which input data is available. A multidimensional report based on this MDDB could allow users to research the total sales of a product in a given year, and then drill down to total sales of the product in a given month. Because these summary statistics are already stored in the NWAY cube, little processing at run time is required.

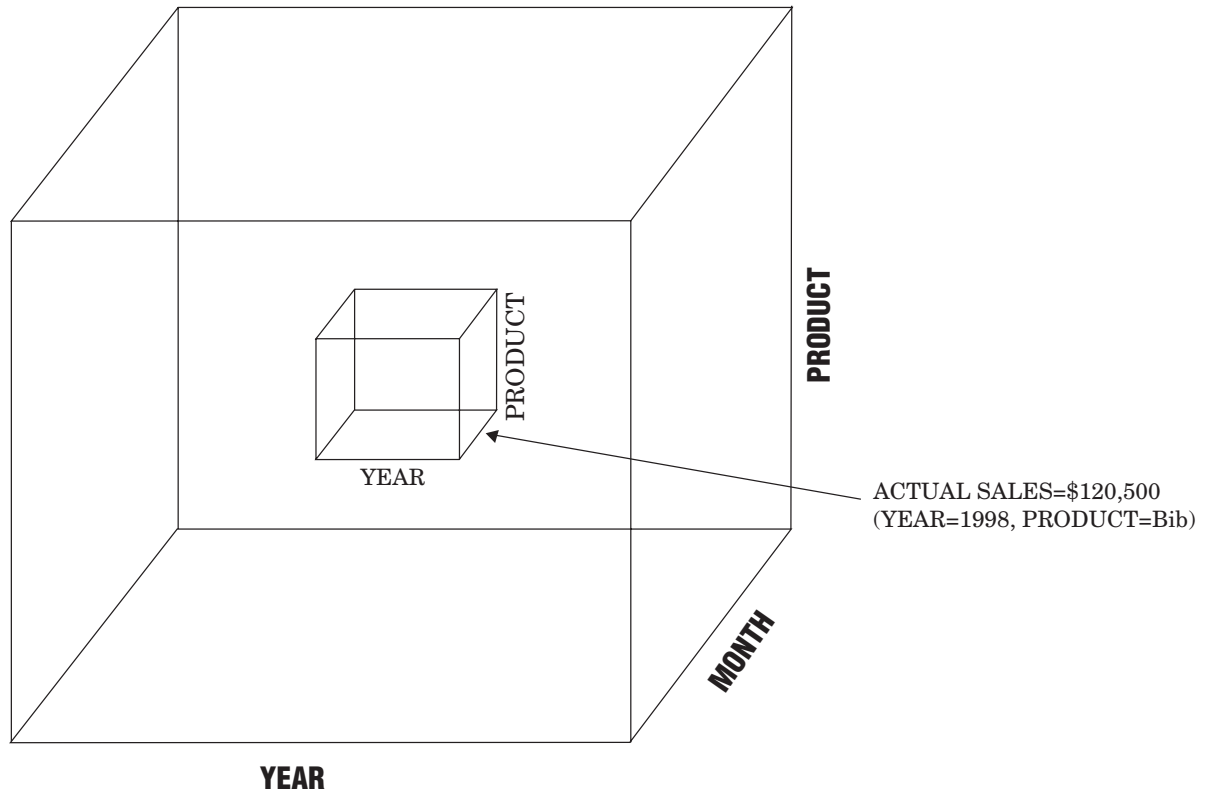
Conceptually, your NWAY cube would look like this:

Figure 2.1 NWAY Cube



If you know of common queries that can be answered using a smaller set of crossings, you could create a subcube that specifies the exact crossing required. For example, assume that you know that the most commonly requested multidimensional report will be one that allows the user to research total sales in a given year for a given product. You could define a subcube that contains only the YEAR and PRODUCT variables.

Figure 2.2 NWAY Cube and Subcube



Of course, the crossing required to answer this query can be summarized from the NWAY cube, but because you know that the MONTH variable will not be needed, you would define the subcube to improve reporting performance. Achieving optimal performance from your MDDB depends in part on the subcubes you choose to define. Chapter 3, “Building and Updating MDDBs,” on page 11 explains this and other performance considerations in detail.

*Note:* The preceding example is highly simplified. The real power of MDDBs lies in their ability to handle data in many dimensions, not just two or three. This example was used because it is easy to illustrate; it is difficult to imagine or draw an object with more than three dimensions. So, although an object with a YEAR dimension and a PRODUCT dimension does not really make a “cube” as in the previous diagram, the important thing to remember is that you can define a subcube on your MDDB with whatever dimensions that you want. △

---

## How Can I Use MDDBs?

Once you have created the MDDBs that are required to meet the needs of your organization’s users, you can create applications that access the MDDBs by using SAS/EIS software or SAS/AF software. Alternatively, you can give users access to MDDBs via your organization’s intranet by deploying the MDDB Report Viewer application.

SAS/EIS software provides a host of MDDB-enabled objects for use in creating multidimensional reports. These reporting objects, which provide a choice of styles for dynamic display and interactive analysis, include a variety of charts, maps, and tables.

The objects have multiple levels with built-in drill-down capabilities for fast navigation through data. With these customizable viewers, users can

- navigate through the data
- perform analyses by adding calculations at run time
- dynamically reach through to detail data anywhere on the network.

For details on using an MDDB with SAS/EIS objects, refer to “Using an MDDB with SAS/EIS Software” on page 27, or *Getting Started with SAS/EIS Software*.

It is also possible to use an MDDB in a SAS/AF application. However, if you want to access MDDBs with a SAS/AF application, it is usually best to subclass one of the MDDB-enabled objects from SAS/EIS software. For more details on using an MDDB with SAS/AF software, refer to “Using an MDDB with SAS/AF Software” on page 28.

Finally, if you want to access MDDBs without creating and deploying your own application, you can use the MDDB Report Viewer application, which is available as a component of SAS/IntrNet Application Dispatcher software. The MDDB Report Viewer enables users to generate and view reports and graphs of data that are stored in MDDBs from a remote client. For more information on using an MDDB with the MDDB Report Viewer, refer to “Using an MDDB with the MDDB Report Viewer” on page 30.

---

## Why Should I Use MDDBs?

There are several advantages to creating and using MDDBs, including the following:

### Reach-through

Using MDDBs with SAS/EIS software allows users to retrieve the underlying detail data that make up a multidimensional view.

### Integration

The OLAP functionality achieved with the SAS/MDDB Server extends the decision support environment by providing multidimensional views of business data within a data warehouse architecture. This means superior integration with warehouse managed data, analytic capabilities, and presentation capabilities.

### Optimized Storage and Performance

SAS/MDDB Server software minimizes the amount of space needed to store multidimensional data. This reduction is due largely to the aggregation of relational warehouse data into dimensions in the MDDB, but other factors include

- *Controlling the storage of statistics.* The size of an MDDB can be affected significantly by whether statistics are stored or derived at run time. By default, a minimum number of statistics are stored in MDDBs. Keeping the size of the MDDBs to a minimum by storing only simple statistics means that MDDBs require short build times and can be easily and quickly maintained by data warehouse administrators.
- *Preventing data expansion through efficient handling of sparse data.* The best measurement of the efficiency of a multidimensional data model is in the number of values that it can store within a given amount of space. MDDBs do not store data in the NWAY cube or in subcubes for crossings that are not populated in the input data. Therefore, the actual input cells are smaller than the number of potential cells proportional to the sparsity of the data. By storing only the cells where data actually exists, the SAS/MDDB Server software keeps the size of the MDDBs to a minimum.
- *Preventing data explosion through user-defined subtables.* Data explosion is related to the way in which multidimensional data operations can cause

extensive increases in data volume. SAS/MDDB Server software controls data explosion by allowing users to control which aggregations are stored as subtables.

- *Storing dimensional labels separately from summaries.* By storing labels separately, SAS/MDDB Server saves storage space for each record.

#### Security

Security is a key issue in the implementation of an OLAP solution. SAS/MDDB Server software provides applications-level security by enabling password protection. Also, when using SAS/EIS software to access your multidimensional data, you can use the SAS/EIS Access Control facility to implement cell-level security. For details on the SAS/EIS Access Control Facility, see the *SAS/EIS Software: Administrator's Guide*.

You should consider each of these factors when deciding how and when to use MDDBs and the SAS/MDDB Server in your application development environment.





The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/MDDDB® Server Administrator's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

**SAS/MDDDB® Server Administrator's Guide, Version 8**

Copyright © 1999 SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-504-3

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute, Inc.

**U.S. Government Restricted Rights Notice.** Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

IBM®, ACF/VTAM®, AIX®, APPN®, MVS/ESA®, OS/2®, OS/390®, VM/ESA®, and VTAM® are registered trademarks or trademarks of International Business Machines Corporation.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.