crystal decisions™

# Crystal Analysis Professional

## Microsoft® SQL Server™ Analysis Services White Paper

## Overview

This white paper explores, at a high level, the key concepts of Microsoft SQL Server Analysis Services (formerly OLAP Services). It identifies key features and terminology, and highlights features supported by Crystal Analysis Professional.

Information in this white paper is based on the versions of Microsoft SQL Server and Crystal Analysis Professional, which were generally available at the time of writing (SQL Server7, SQL Server 2000, Crystal Analysis Professional 8.0). Its validity may change with later releases of software.

## Contents


**INTRODUCTION** ..... **3**

**OLAP TECHNOLOGY** ..... **3**

*Cubes* ..... *3*

Aggregations ..... 4

Partitions ..... 5

Other Cube Features ..... 5

Drill through ..... 6

*Dimensions* ..... *6*

Dimension Storage Options ..... 6

Hierarchies ..... 6

Dimension Members ..... 8

Special Dimension Types ..... 8

**ADMINISTRATION** ..... **9**

*Building Cubes* ..... *9*

*Security* ..... *9*

Roles ..... 9

**OTHER FEATURES** ..... **10**

Actions ..... 10

Data Mining ..... 10

Distinct Count ..... 10

**GETTING DATA OUT OF SQL SERVER OLAP CUBES** ..... **11**



Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

*Open Interfaces* ..... *11*
OLEDB for OLAP ..... 11
MDX ..... 12
XML for Analysis ..... 12
*Client Tools* ..... *12*
Office 2000 ..... 12
Crystal Analysis Professional ..... 13

**ARCHITECTURE ..... 14**
*SQL Server Architecture* ..... *14*
*SQL Server and Crystal Analysis* ..... *14*

**FURTHER READING ..... 15**

 Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved. 

# Introduction

A few years ago Microsoft released its first OLAP Server with the release of SQL Server7.0. The aim was to create an OLAP product for the mass market, reducing the price of the technology and significantly increasing the number of OLAP systems deployed. While the majority of SQL Server users are still not making use of its OLAP capabilities, independent sources report rapid growth in OLAP market share for SQL Server™.

During the development of SQL Server2000, the OLAP server was updated and Data Mining features were added. This resulted in a product name change to SQL Server Analysis Services when SQL Server2000 was released.

# OLAP technology

## Cubes

Microsoft's OLAP solution offers a wide range of options for dealing with multidimensional data. The general term used in Microsoft SQL Server for a multidimensional data set is a cube. This section identifies the main features available when designing OLAP cubes in Microsoft SQL Server™.

Four basic types of cube technology are available:

- MOLAP (Multi-dimensional OLAP).
- ROLAP (Relational OLAP).
- HOLAP (Hybrid OLAP).
- CUB files (local cubes).

For each of the first three cube types, the database administrator can further tune the design to cater for performance and scalability requirements of specific applications, including:

- How aggregations are handled.
- Partitioning cubes.
- Creating virtual cubes

**MOLAP cubes** store all their data in an optimized multi-dimensional database. The database contains the base (fact data), the aggregations derived from the fact data and a set of indexes. In response to a query Analysis Services uses the various indexes to locate the data required for the query.

This database takes the form of proprietary format files on the server's file system, typically found in the directory "program files\Microsoft Analysis Services\Data\Database Name\Cube Name".

**ROLAP cubes** store all their data in a relational database. This can be any ODBC or OLEDB compliant database, which means that in addition to SQL Server tables, data can be stored in other databases such as Oracle, DB2 or Sybase.

The data is stored in many tables, replicating the file structure of the proprietary multi-dimensional database. This includes tables for base (fact) data, aggregations and indexes or storage maps which explain which data exists in the cube. Due to the involved nature of these tables, they are not suitable for general SQL queries.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

As with all ROLAP solutions, the benefits are debatable. On one hand the technology is familiar to RDBMS administrators, who will also like the fact that the data can be backed up with the standard SQL Server tools. Conversely, performance is likely to be noticeably slower than for an equivalent MOLAP cube.

**HOLAP cubes** offer a compromise between the full MOLAP and ROLAP solutions. With this type of cube, the base data remains in the RDBMS with only the aggregations being stored in the multi-dimensional database.

The base data will usually be accessed less often than the aggregates, so the performance penalty will not be as severe. This cube type is most suited to situations where there is a large amount of base data and rapid access is required for the aggregations. However, the base level data of most cubes is almost always a small fraction of the consolidated cube data so unless users are frequently requiring access to the base data there may not be a significant benefit to utilizing this cube type over MOLAP.

**Local Cubes** (or CUB files) are portable cubes which can be used without a connection to the server. Developers can use local cubes to provide a 'takeaway' capability for their applications. MS SQL Server Analysis Services CUB files can be either MOLAP based, in which case the file is self-contained, or use MOLAP technology, which requires a connection to the relational data.

**HTTP Cubes** (Introduced in SQL Server2000) allow a client to connect to an OLAP cube using HTTP or HTTPS for transport and authentication, through the Microsoft Internet Information Server web server.

## Aggregations

The base data in an OLAP cube will typically contain information about sales for a specific product in a particular store on a particular week. However, users will more often want to know about summarized data such as the sales for a product in a given region over a calendar quarter. The efficient pre-calculation and storage of this summarized data, known as aggregations, is the reason for the performance advantage of OLAP cubes.

In MOLAP and HOLAP cubes, these aggregations are stored in an optimized multidimensional database. In the case of ROLAP cubes, the aggregations are stored in the relational database.

The simplest way to deal with aggregations is to calculate all of the aggregations based on what a user is *most likely* to ask for. Pre-calculating all aggregations has the advantage of delivering uniform performance, but at the cost of storage space - aggregations almost always consume significantly more space than the base data they are derived from.

**Aggregate compression** allows SQL Server to compress the aggregations to minimize the space they occupy.

**Aggregation optimization** is a wizard-driven process that helps the administrator balance the storage and performance requirements for cubes. The wizard helps the administrator to decide which aggregations should be pre-calculated, and those which can be derived only when the user requests them


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

**Usage-based optimization** offers a further refinement to this process. SQL Server gathers information about which aggregations are most often required in user queries. Later the administrator can use a wizard to analyze this information and build the aggregations to give the best performance.

## Partitions

All SQL Server OLAP cubes have at least one partition, which contains all of the data and aggregations for a cube. When managing larger cubes, it may be beneficial for the administrator to split the cube into several partitions.

**Multiple Partitions** are available in SQL Server Enterprise Edition, and allow the administrator to split a large cube into smaller partitions, making it easier to manage. As an example, the cube may be partitioned along a time dimension, with each month having a different physical cube. It is possible to partition along more than one dimension, so this example may be extended further to allow each financial year to reside in a different partition. Another option would allow the aggregations to be stored in a different partition from the base data.

Another benefit of partitioning is that it allows cubes to be built from multiple data sources. For example, if financial data is held in one database and HR data in another, it is not possible to build a single cube containing both sets of data unless they are combined in a star schema. However, a partition can be built from each data source and these partitions can then be joined into a single cube.

**Remote Partitions** (introduced in SQL Server2000) allow the partitions of a cube to be held on different physical servers. This may be done for reasons of scalability, but it also allows administrators to combine external data from a data provider, delivered as an HTTP cube for example, which may be included as a partition in a larger cube.

## Other Cube Features

**Virtual cubes** allow you multiple cubes to be combined into a single cube, or the data from a larger cube to be divided into smaller views.

**Write-enabled cubes** are available in all versions of SQL Server™. SQL Server OLAP cubes (other than local cubes) can be defined as write-enabled in the Analysis Manager (OLAP Manager in SQL Server7). When values are written back to the cube, the new value is stored in a separate database table as a delta. For example, if a cell with value 5 is changed to 7 then the delta value of +2 is stored in the table.

These deltas can either be committed or rolled back (in the latter case they are simply deleted from the table). Once values are committed to the table, any access to the modified cell will first read a value of 5 from the cube, then the delta value of +2, giving a result of 7.

If the cell is further modified, for example to 10, the table now contains two delta values for the cell - +2 and +3 (5 + 2 + 3 = 10). Anyone reading that cell value will now read 5 from the cube; the two delta values will then be read and added to give the result of 10.

Response can degrade as these delta values build up: therefore, the administrator should periodically roll these new values into the cube itself, clearing out the delta table.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

It is the responsibility of the application to prevent two users updating the same data cell. In the case of conflicting updates, the last commit will win.

**Allocated Writeback** (introduced in SQL Server2000) allows data to be entered at high levels in a dimension hierarchy and automatically allocated down to the base level, using a selection of allocation rules.

### Drill through

Drill through (introduced with SQL Server2000) is the ability to get from a cell in a cube to the underlying relational data it is composed of. Drill through options are controlled using the Analysis Manager (OLAP manager in SQL Server7). This dialog controls whether drill through is available, which columns of data are returned in the drill through and also allows the rows returned to be filtered with a specific where clause.

## Dimensions

Dimensions are the basic building blocks of OLAP cubes. SQL Server offers the administrator a range of options for dealing with different dimension requirements. Dimensions are built in the Analysis Manager (OLAP Manager in SQL Server7) using wizards.

Dimensions can be created as **shared** or **private**. Shared dimensions can be used in many cubes, while private dimensions can only be used in their own cube. Each dimension can be assigned a **type** (e.g. Time, Geography, Bill of Materials) which can be interpreted by client applications.

Each cube must have a **measures** dimension containing members representing the items to be analyzed, e.g. unit sales, costs etc. This is defined using a separate wizard in the Analysis Manager.

### Dimension Storage Options

**MOLAP dimensions** are stored in an optimized format allowing for maximum performance.

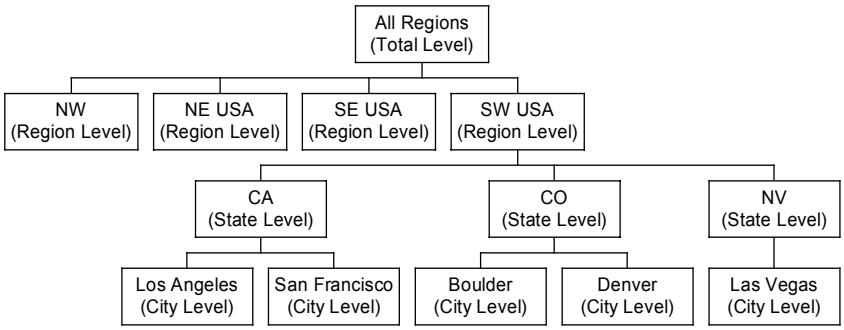
**ROLAP dimensions** do not create a separate optimized dimension, rather they keep all of the dimension's metadata in the relational database. This may be beneficial for large dimensions, and allows real-time updates to dimensions. The trade-off for this is slower performance. ROLAP dimensions are only available with SQL Server enterprise edition.

### Hierarchies

Hierarchies group members together, allowing data to be aggregated and helping the user in drill down operations. Each hierarchy is organized into levels, which may be given a name and may be interpreted by the client tools.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

SQL Server supports several types of hierarchies in its OLAP dimensions.




**Balanced hierarchies** are the only type supported in SQL Server7. A balanced hierarchy must be divided into levels, with all branches of the tree being a uniform depth. A typical balanced hierarchy is a geography dimension showing cities in the USA. This is illustrated in the diagram below. The detail below NW, NE and SE is omitted for clarity, but for each of these regions there would be states then cities.

**Unbalanced Hierarchies** (introduced in SQL Server2000) allow levels of the balanced hierarchy to be skipped. For example the dimension above may be extended to include regions which do not have states, perhaps introducing a UK region with cities of London, Edinburgh and Cardiff.

**Parent Child Hierarchies** (introduced in SQL Server2000) are used when there are no set levels in the hierarchy, and each member is simply defined in terms of which parent it belongs to. This is typically used in hierarchies representing organization structures, product catalogs, bills of materials etc.

**Multiple Hierarchy** support was greatly improved in SQL Server2000. A dimension may have several hierarchies, each of which is identified by a name. For example, the stores in a supermarket chain may be grouped by geography or alternatively by the size of the stores in square feet.

**Custom Rollup formulas** (introduced in SQL Server2000) allow the aggregated values to be based on calculations other than a simple summation. As an example, a time dimension may need to use period end balances or average balances when consolidating weeks to months to quarters. The custom roll-up formula applies to all members in a given level.

**Custom member formulas** (introduced in SQL Server2000) are like custom rollup formulas, except they are applied to a single member rather than to all members at a given level in a hierarchy.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

## Dimension Members

The items within a dimension are known as **members**, for example Sales might be a member of the Measures dimension.

**Member Properties** can be assigned as required by an application, to any member in a dimension. For example in a product dimension one such property might be color.

**Calculated Members** are defined by the database administrator using MDX. They allow data which is not in the source database to be derived when the cube is queried. For example the source data for a cube may contain Sales and Costs, and the administrator may add a Margin member which subtracts one from the other.

**Data Members** (introduced in SQL Server2000) allow fact data, i.e. non-aggregated data, to be held at any level of a hierarchy, rather than just at the base level.

**Named Sets** (introduced in SQL Server2000) allow the administrator to define commonly used subsets of dimension members, using MDX, and give them a name which means something to the application users. Named sets can be static, e.g. five common products, or evaluated at query time, e.g. the top 5 products this month based on sales.

**Default Members** (introduced in SQL Server2000) allow the administrator to define the default view of a dimension for each security role.

## Special Dimension Types

**Virtual dimensions** allow a new dimension to be created based on member properties. A virtual dimension based on the property color would give a new dimension that rolls the products up into a hierarchy based on their color. Once created, virtual dimensions can be used in cubes just like any other dimensions.

**Changing Dimensions** allow for situations where dimension changes can occur at any time, and the changes need to be reflected without fully processing the cube. This capability comes at the expense of slightly slower performance for these dimensions.

**Data mining dimensions** (introduced in SQL Server2000) are a special kind of shared dimension, which is the output of a data mining model and which can only be used to build virtual cubes. For example, a data mining model may create a dimension of customers who are a good credit risk. This could be used to build a virtual cube on the base data, creating a cube containing just those customers who were identified by the mining model.

**Write-enabled dimensions** (introduced in SQL Server2000) allow a client tool or application to modify the members of a dimension. This is controlled in the Analysis Manager (OLAP Manager in SQL Server7), and the ability can be assigned to different security roles.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

# Administration

A GUI for managing the OLAP server is supplied in the SQL Server Analysis Manager (OLAP Manager in SQL Server7).

## Building Cubes

Analysis Services has a simple GUI driven interface to build cubes. Wizards help the administrator to build the dimensions and cubes from database tables containing the fact and dimension data in a star or snowflake schema.

**DSO (Decision Support Objects)** is the administration object model in Analysis Services and can be called from programming environments (like VB) to define, manipulate and process cubes on the server - in fact the GUI cube building interface described above is just a DSO application. Using DSO, the application programmer can automate any of the administration tasks described in this whitepaper.

**Data Transformation Services** is a set of GUI tools and an object model, supplied with SQL Server™, which allow an application programmer to automate the creation of the star or snowflake schema and the cubes.

## Security

SQL Server allows the administrator to define security at various levels. Crystal Analysis Professional is able to reflect all of these security options in the cubes, dimensions and data that it makes available to the user.

### Roles

Roles define groups of users with similar access rights to the SQL Server OLAP data. Once roles have been defined, they can be use to apply security to objects – cubes, dimensions, mining models etc. – and to limit access to features such as dimension writeback.

**Cube Security** allows roles to be defined for specific cubes, and security rights to be assigned to the cubes. This can be used to allow some cubes to be hidden from certain users.

**Dimension member security** restricts the view of a dimension which is available to users in a specific role.

**Cell level security** restricts which cells a user can see and which, if any, can be written to, based on defined roles and security policies.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

# Other Features

This section highlights some other popular features of SQL Server2000 Analysis Services.

## Actions

Actions is a new feature for SQL Server2000, allowing extra workflow to be built into an OLAP application. This enables applications to extend their functionality beyond analysis of the OLAP data. For example a user might be using a supplier performance cube. Having identified an under-performing supplier, the user could right-click on the supplier details and send an email to the contact at the supplying company.

Actions are associated with a cube and are defined by the administrator using the Analysis Manager. They can be assigned at various places in the cube:

- The cube.
- A dimension within the cube.
- A single dimension level.
- A group of cells in the cube.
- Sets defined in the cube.

An action can have several pre-defined results, for example calling a URL. An MDX statement can be used to customize the action based on the selected object, e.g. customizing an email address with the contact details for a supplier.

An action type 'proprietary' exists to enable ISVs to create their own actions, for example we could use this to call a Crystal Report.

## Data Mining

Data mining features have been introduced with SQL Server2000. These allow decision trees and clustering algorithms to be applied to specified relational or OLAP data. Data mining models allow a technique to be set up by the administrator, with parameters pre-defined. Users can then apply this data mining model through client tools.

A full discussion of the data mining features in SQL Server2000 is beyond the scope of this document.

## Distinct Count

Distinct count allows for SQL Server to calculate values that are non-additive or semi-additive. As an example, consider the table below:

| | Product 1 | Product 2 | All Products |
|---|---|---|---|
| **Customer 1** | 12 | | 12 |
| **Customer 2** | | 10 | 10 |
| **Customer 3** | 5 | 8 | 13 |
| **All Customers** | 17 | 18 | 35 |


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

This shows the units sold to each customer for each of the two products. Now imagine that we want to count how many customers bought each product, and how many bought both products. In the table below, this has been added to the measures dimension, and the measures dimension has been pivoted into the rows.

| | Product 1 | Product 2 | All Products |
|---|---|---|---|
| **Customers** | 2 | 2 | 1 |
| **Unit Sales** | 17 | 18 | 35 |
| **Sales Value** | 125 | 247 | 372 |

A **distinct count** has been used to calculate the **Customers** measure, which shows that 2 customers bought 'Product 1', 2 customers bought 'product 2' but only 1 customer bought both products.

# Getting Data out of SQL Server OLAP Cubes

## Open Interfaces

### OLEDB for OLAP

OLEDB for OLAP (also known as **ODBO**) is an object model, defined by Microsoft on top of the existing OLEDB specification, specifically designed for querying OLAP databases. It is the closest thing to a standard API for OLAP servers available today, particularly since the collapse of the OLAP council along with its OLAP API standard.

**OLEDB for OLAP Providers** are OLAP servers which support the ODBO standard, offering the necessary interfaces to allow client tools to query the OLAP cubes.

**OLEDB for OLAP Consumers** are client tools which are able to retrieve and analyze data from any OLEDB for OLAP provider.

The standard has been somewhat successful in the market, with vendors such as SAP and Applix offering as ODBO providers for their OLAP servers, and many ODBO consumers being developed by from large and small ISVs. Tools vendors are also entering the market with middleware to help build OLAP server vendors to become ODBO providers.

Some of the ODBO interfaces are mandatory, while others are optional. SQL Server implements all of the interfaces but other ODBO providers may only support the mandatory features. Therefore, it is not guaranteed that any ODBO client will work with any ODBO provider.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

### MDX

MDX is a complex multi-dimensional query language which allows data to be retrieved from an ODBO provider. It is much more complex than SQL and is not recommended for the faint-hearted. Many ODBO clients claim support for advanced SQL Server features, but expect the user to achieve them through MDX. It is more realistic to say that if a client tool only supports a feature through an MDX editor, the tool does *not* support the feature.

**User Defined Functions (UDFs)** allow customers to extend the function library available in MDX. The UDF is written as part of a user function library, which must then be installed on all clients and SQL Servers which need access to that function.

### XML for Analysis

This is an emerging standard being championed by Microsoft and Hyperion (for their Essbase OLAP server) as the new open interface to OLAP and data mining servers. XML for analysis brings SQL Server Analysis Services into Microsoft's .NET software as a service strategy.

The standard is based on the Simple Object Access Protocol (SOAP), which has been ratified by many industry leaders like Sun and IBM. This gives it more credibility than ODBO, which was based on Microsoft's own COM technology.

SQL Server2000 does not support XML for analysis at the time of writing, nor are any client tools available. It is likely that this situation will change rapidly in the second half of calendar year 2001 as support becomes generally available in SQL Server™.

## Client Tools

Microsoft has not, so far, concentrated on developing client tools for Analysis Services. Office 2000 contained updated pivot tables with direct connection to ODBO cubes, but otherwise Microsoft has been keen to engage Independent Software Vendors (ISVs) in producing client tools.

### Office 2000

This is the default client for most SQL Server OLAP users. Excel provides improved pivot tables which connect directly to the OLAP server, which is a more scalable solution than building pivot tables directly from Excel data. These capabilities are also included in the Office Web Components, a group of ActiveX controls which allow common Office features to be distributed over the web.

Office 2000 is a good first step for customers, but it does not have the same range of analytical capabilities as tools like Crystal Analysis Professional.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

## Crystal Analysis Professional

Crystal Analysis Professional (CA Pro) was written from scratch to be a first class client for SQL Server Analysis Services. Its goal is to allow power users and IT departments to create OLAP applications for unsophisticated users, leveraging all of the capabilities of advanced MDX but without requiring knowledge of MDX. It also put sophisticated analysis capabilities into the hands of the end users of the applications, again without requiring any knowledge of MDX.

The table below highlights key SQL Server features supported by Crystal Analysis Professional 8.

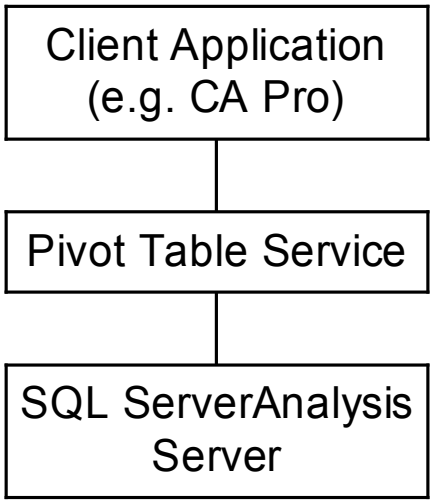
| | V8 |
|---|---|
| **Offline Cube (.CUB) Files** | |
| Read data from .CUB file | ✓ |
| **Calculated Members** | |
| Create calculated members from GUI experts | ✓ |
| Allow use to enter MDX for calculations | ✓ |
| Calculations performed on the server | ✓ |
| Calculations on any dimension | ✓ |
| Calculations from calculations | ✓ |
| Semi-additive variables | ✓ |
| **Dimension Hierarchies** | |
| Balanced hierarchies | ✓ |
| Ragged hierarchies | ✓ |
| Unbalanced hierarchies | ✓ |
| Multiple hierarchies | ✓ |
| **Dimension Member Properties** | |
| Display member properties | ✓ |
| Filter based on member properties | ✓ |
| **HTTP Cubes** | |
| Read data from HTTP cubes | ✓ |
| **Miscellaneous Features** | |
| Distinct Counts | ✓ |
| **Named Sets** | |
| Use Named Sets in calculated members | ✓ |
| **Security** | |
| Honor roles based security | ✓ |
| Cube security | ✓ |
| Dimension member security | ✓ |
| Cell security | ✓ |


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

# Architecture

## SQL Server Architecture

The diagram below shows the key components of SQL Server Analysis Services (or OLAP Services).




**Client Applications** communicate with the Analysis Server using OLEDB for OLAP, OLEDB for Data Mining, and MDX using the public interfaces of the Pivot Table Service.

**Pivot Table Service**, or **PTS**, is the client connector which is used by applications to access OLAP data and Data Mining capabilities on the SQL Server™. PTS exposes these capabilities through OLEDB for OLAP, OLEDB for Data Mining and MDX. It is not possible for applications to get 'inside' PTS - the transport between PTS and the SQL Server is proprietary.

PTS is responsible for caching data and metadata retrieved from the Analysis Server. It can use the local data to satisfy queries and calculations, or it can ask the server for more data to satisfy the requirement. This 'smart caching' is key to the performance of SQL Server Analysis Services.

PTS can also create local cubes and local data mining models, which can be used without a connection to SQL Server™.

**The Analysis Server** holds the OLAP cubes and deals with security, scalability, backup, recovery etc. The only way to talk to the analysis server is through the Pivot Table Service.

## SQL Server and Crystal Analysis

For more details on the Crystal Analysis architecture and technology, please see the CA Pro product home page of the Crystal Decisions web site.


Copyright © 2001 Crystal Decisions, Inc. All Rights Reserved.

# Further Reading

More information about SQL Server Analysis Services is available from a variety of sources. Some are suggested below.

**SQL Server Books online** is provided with SQL Server in on-line help form

**MSDN** contains all of the same information as SQL Server Books and is available over the Internet at http://msdn.Microsoft.com

**Newsgroups** such as comp.databases.olap and Microsoft.sqlserver.olap. This latter newsgroup is available online at MSDN.

**The Microsoft web site** contains a number of technical papers and white papers on SQL Server Analysis Services.
http://www.Microsoft.com/sql/techinfo/olap.htm

SQL Server Analysis Services has generated sufficient interest for a number of publishers to write books on the subject. At the time of writing, a search for OLAP on Amazon.com returned 12 books, 10 of which were specific to SQL Server™.

© 2001 Crystal Decisions Inc. All rights reserved. Crystal Decisions, Crystal Reports, Crystal Enterprise, Crystal Analysis, Crystal Applications – Holos Analytic System, and Seagate Info are trademarks of Crystal Decisions, Inc, or Seagate Technologies, Inc. Microsoft and SQL Server are registered trademarks of Microsoft Corporation in the US and/or other countries. All other trademarks referenced are the property of their respective owner.