Analysis Services - An Introduction

In SQL Server, so far we have focused on getting data into a SQL Server database and then later getting the same data out of the database. We know how to create tables, insert data, and use SQL statements, views, and stored procedures to retrieve the data. This pattern of activity, where individual actions deal with small pieces of the database, is sometimes called *online transaction processing*, or OLTP.

But there's another use for databases, especially large databases. Suppose we have an online book store and have sales records for 50 million book sales. Maybe books on introductory biology show a strong spike in sales every September. That's a fact that we can use to our advantage in ordering stock, if only we knew about it.

Searching for patterns like this and summarizing them is called *online analytical processing*, or OLAP.

Microsoft SQL Server 2008 includes a separate program called SQL Server Analysis Services to perform OLAP analysis. In this introduction we will see the basics of setting up and using Analysis Services.

Understanding Analysis Services

The basic idea of OLAP is fairly simple. Let's think about that book ordering data for a moment. Suppose you want to know how many people ordered a particular book during each month of the year. You could write a fairly simple query to get the information you want. The catch is that it might take a long time for SQL Server to churn through that many rows of data.

And what if the data was not all in a single SQL Server table, but scattered around in various databases throughout your organization? The customer info, for example, might be in an Oracle database, and supplier information in a legacy xBase database.

SQL Server can handle distributed heterogeneous queries, but they're slower. What if, after seeing the monthly numbers, you wanted to drill down to weekly or daily numbers? That would be even more time-consuming and require writing even more queries.

This is where OLAP comes in. The basic idea is to trade off increased storage space now for speed of querying later.

OLAP does this by precalculating and storing aggregates. When you identify the data that you want to store in an OLAP database, Analysis Services analyzes it in advance and figures out those daily, weekly, and monthly numbers and stores them away (and stores many other aggregations at the same time). This takes up plenty of disk space, but it means that when you want to explore the data you can do so quickly.

First, though, you need to familiarize yourself with a new vocabulary. The basic concepts of OLAP include:

- Cube
- Dimension table
- Dimension
- Fact table
- Schema

Cube

The basic unit of storage and analysis in Analysis Services is the *cube*. A cube is a collection of data that's been aggregated to allow queries to return data quickly. For example, a cube of order data might be aggregated by time period and by title, making the cube fast when you ask questions concerning orders by week or orders by title.

Cubes are ordered into *dimensions* and *measures*. The data for a cube comes from a set of staging tables, sometimes called a star-schema database. Dimensions in the cube come from *dimension tables* in the staging database, while measures come from *fact tables* in the staging database.

Dimension table

A *dimension table* lives in the staging database and contains data that you'd like to use to group the values you are summarizing. Dimension tables contain a primary key and any other attributes that describe the entities stored in the table. Examples would be a Customers table that contains city, state and postal code information to be able to analyze sales geographically, or a Products table that contains categories and product lines to break down sales figures.

Dimension

Each cube has one or more *dimensions*, each based on one or more dimension tables. A dimension represents a category for analyzing business data: country or product line in the examples above. Frequently, a dimension has a natural hierarchy so that lower results can be "rolled up" into higher results. For example, in a geographical level you might have city totals aggregated into state totals, and state totals into country totals.

Fact table

A *fact table* lives in the staging database and contains the basic information that you wish to summarize. This might be order detail information, payroll records, drug effectiveness information, or anything else that's amenable to summing and averaging. Any table that you've used with a `Sum` or `Avg` function in a totals query is a good bet to be a fact table. The fact tables contain fields for the individual facts as well as foreign key fields relating the facts to the dimension tables.

Schema

Fact tables and dimension tables are related, which is hardly surprising, given thatyou use the dimension tables to group information from the fact table. The relations within a cube form a *schema*.

There are two basic OLAP schemas: star and snowflake.

In a *star schema*, every dimension table is related directly to the fact table.
In a *snowflake schema*, some dimension tables are related indirectly to the fact table.

For example, if your cube includes `OrderDetails` as a fact table, with `Customers` and `Orders` as dimension tables, and `Customers` is related to `Orders`, which in turn is related to `OrderDetails`, then you're dealing with a snowflake schema.

To launch Business Intelligence Development Studio, select Microsoft SQL Server 2008 → SQL Server Business Intelligence Development Studio from the Programs menu.

BIDS shares the Visual Studio shell, so if you have Visual Studio installed on your computer, this menu item will launch Visual Studio complete with all of the Visual Studio project types (such as Visual Basic and C# projects).

Creating a Data Cube

To build a new data cube using BIDS, you need to perform these steps:

- Create a new Analysis Services project
- Define a data source
- Define a data source view
- Invoke the Cube Wizard

We'll look at each of these steps in turn.

You'll need to have the AdventureWorksDW2008 sample database installed to complete the examples in this chapter. This database is one of the samples that's available with SQL Server.

Creating a New Analysis Services Project

To create a new Analysis Services project, you use the New Project dialog box in BIDS. This is very similar to creating any other type of new project in Visual Studio.

Try It!

To create a new Analysis Services project, follow these steps:

1. Select Microsoft SQL Server 2008 → SQL Server Business Intelligence Development Studio from the Programs menu to launch Business Intelligence Development Studio.
2. Select File → New → Project.
3. In the New Project dialog box, select the Business Intelligence Projects project type.
4. Select the Analysis Services Project template.
5. Name the new project AdventureWorksCube1 and select a convenient location to save it.
6. Click OK to create the new project.

Defining a Data Source

A data source provides the cube's connection to the staging tables, which the cube uses as source data. To define a data source, you'll use the Data Source Wizard. You can launch this wizard by right-clicking on the Data Sources folder in your new

Analysis Services project. The wizard will walk you through the process of defining a data source for your cube, including choosing a connection and specifying security credentials to be used to connect to the data source.

Try It!

To define a data source for the new cube, follow these steps:

1. Right-click on the Data Sources folder in Solution Explorer and select New Data Source.
2. Read the first page of the Data Source Wizard and click Next.
3. You can base a data source on a new or an existing connection. Because you don't have any existing connections, click New.
4. In the Connection Manager dialog box, select the server containing your analysis services sample database from the Server Name combo box.
5. Fill in your authentication information.
6. Select the Native OLE DB\SQL Native Client provider (this is the default provider).
7. Select the AdventureWorksDW2008 database. Figure 15-2 shows the filled-in Connection Manager dialog box.
8. Click OK to dismiss the Connection Manager dialog box.
9. Click Next.
10. Select Use the Service Account impersonation information and click Next.
11. Accept the default data source name and click Finish.

Defining a Data Source View

A data source view is a persistent set of tables from a data source that supply the data for a particular cube. It lets you combine tables from as many data sources as necessary to pull together the data your cube needs. BIDS also includes a wizard for creating data source views, which you can invoke by right-clicking on the Data Source Views folder in Solution Explorer.

Try It!

To create a new data source view, follow these steps:

1. Right-click on the Data Source Views folder in Solution Explorer and select New Data Source View.
2. Read the first page of the Data Source View Wizard and click Next.
3. Select the Adventure Works DW data source and click Next. Note that you could also launch the Data Source Wizard from here by clicking New Data Source.
4. Select the `FactFinance(dbo)` table in the Available Objects list and click the > button to move it to the Included Object list. This will be the fact table in the new cube.
5. Click the Add Related Tables button to automatically add all of the tables that are directly related to the `dbo.FactFinance` table. These will be the dimension tables for the new cube.
6. Click Next.
7. Name the new view Finance and click Finish. BIDS will automatically display the schema of the new data source view.

Invoking the Cube Wizard

As you can probably guess at this point, you invoke the Cube Wizard by right clicking on the Cubes folder in Solution Explorer. The Cube Wizard interactively explores the structure of your data source view to identify the dimensions, levels, and measures in your cube.

Try It!

To create the new cube, follow these steps:

1. Right-click on the Cubes folder in Solution Explorer and select New Cube.
2. Read the first page of the Cube Wizard and click Next.
3. Select the option to Use Existing Tables.
4. Click Next.
5. The Finance data source view should be selected in the drop-down list at the top. Place a checkmark next to the `FactFinance` table to designate it as a measure group table and click Next.
6. Remove the check mark for the field `FinanceKey`, indicating that it is not a measure we wish to summarize, and click Next.
7. Leave all Dim tables selected as dimension tables, and click Next.
8. Name the new cube `FinanceCube` and click Finish.

Defining Dimensions

The cube wizard defines dimensions based upon your choices, but it doesn't populate the dimensions with attributes. You will need to edit each dimension, adding any attributes that your users will wish to use when querying your cube.

Try It!

1. In BIDS, double click on `DimDate` in the Solution Explorer.
2. Using below as a guide, drag the listed columns from the right-hand panel (named Data Source View) and drop them in the left-hand panel (named Attributes) to include them in the dimension.

DimDate

CalendarYear
CalendarQuarter
MonthNumberOfYear
DayNumberOfWeek
DayNumberOfMonth
DayNumberOfYear
WeekNumberOfYear
FiscalQuarter
FiscalYear

3. Using below, add the listed columns to the remaining four dimensions.

DimDepartmentGroup

DepartmentGroupName

DimAccount

AccountDescription
AccountType

DimScenario

ScenarioName

DimOrganization

OrganizationName

Adding Dimensional Intelligence

One of the most common ways data gets summarized in a cube is by time. We want to query sales per month for the last fiscal year. We want to see production values year-to-date compared to last year’s production values year-to-date. Cubes know a lot about time.

In order for SQL Server Analysis Services to be best able to answer these questions for you, it needs to know which of your dimensions stores the time information, and which fields in your time dimension correspond to what units of time. The Business Intelligence Wizard helps you specify this information in your cube.

Try It!

1. With your `FinanceCube` open in BIDS, click on the Business Intelligence Wizard button on the toolbar.
2. Read the initial page of the wizard and click Next.
3. Choose to Define Dimension Intelligence and click Next.
4. Choose `DimDate` as the dimension you wish to modify and click Next.
5. Choose Time as the dimension type. In the bottom half of this screen are listed the units of time for which cubes have knowledge. Using the below, place a checkmark next to the listed units of time and then select which field in `DimDate` contains that type of data.

Time Property Name	**Time Column**
Year	CalendarYear
Quarter	CalendarQuarter
Month	MonthNumberOfYear
Day of Week	DayNumberOfWeek
Day of Month	DayNumberOfMonth
Day of Year	DayNumberOfYear
Week of Year	WeekNumberOfYear
Fiscal Quarter	FiscalQuarter
Fiscal Year	FiscalYear

Table 15-1:Time columns for FinanceCube

6. Click Next.