## Лекция 7

## Тема « Data Source Views in Depth »

# Data Source Views in Depth

Data warehouse designs consist of several fact tables and all the associated dimension tables. Small data warehouses are usually comprised of 10 to 20 tables, whereas larger data warehouses can have more than a hundred tables. Even though you have a large number of tables in your data warehouse, you will likely work with a small subset of those tables; each of which has relationships between them. For example, assume you have sales, inventory, and human resources (HR) data to analyze and the HR data is not strongly related to the sales and inventory data but there is a desired linkage. Then you might create two cubes, one for Sales and Inventory information and another one for HR. It is quite possible the Sales, Inventory, and HR information could be stored in a single data source — in the ODS or OLTP system.

Employee information (HR) could be related to the sales and inventory information within the company so far as there is a link between a given sales event and the employee who made the sale. You might want to slice the sales data by a specific employee, but to do so you must access information that is a part of a separate cube only accessible to the HR department (for security reasons). You can get around this problem by making a single DSV containing all the tables that store sales, inventory, and HR information of a company. From that DSV, both cubes can be formulated and permissions set such that only members of the HR group can drill down on personal employee data.

Having a lot of tables in the DSV definitely makes the navigation and usability a bit complex. When you are working on HR data you will only want to see the tables related to this alone. For easy manageability you will need customizable views within your DSV that show only certain tables. SSAS 2008 provides you with the capability of having several views within the DSV that each contains a subset of the DSV ' s tables. These views are called diagrams. By default you get a diagram called < All Tables > when you complete the DSV Wizard. You can create additional diagrams and select the tables that you want to include within them. Next, you learn how to create a new diagram and include only the tables you need.

To create a new diagram, you need to do the following:

1. Right - click the Diagram Organizer pane and select New Diagram, as shown in Figure 4 - 17.

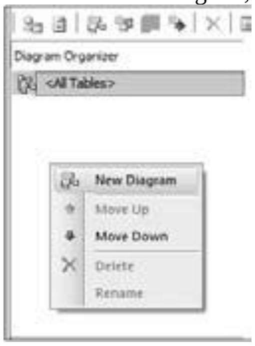
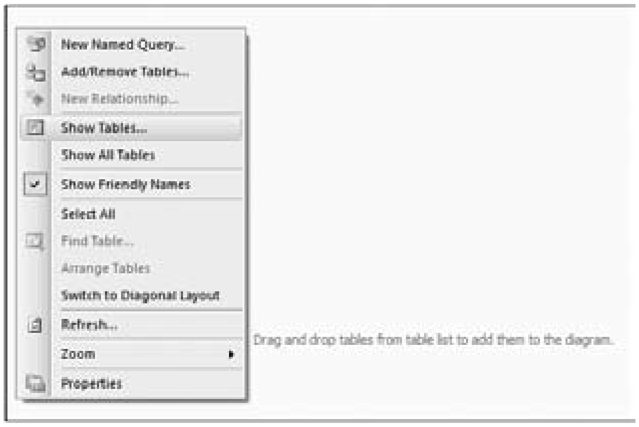


Figure 4-17

2. Name the new diagram “ Internet Sales. ”
3. You now have an empty diagram view. Right - click the diagram view and select Show Tables (see Figure 4 - 18 ). You are presented with a dialog where you can choose the table(s) you want to include in this diagram.



Drag and drop tables from table list to add them to the diagram.

Figure 4-18

4. Select all the tables that are part of the InternetSales fact table and click OK.

This gives you a diagram view of Internet Sales that contains the Internet Sales fact table and the related dimension tables as shown in Figure 4 - 19 . Alternatively you can add the FactInternetSales table to the diagram, right - click it, and select Show Related Tables to achieve the same result. This Internet Sales diagram has seven of the ten tables in the DSV. This makes it much easier to understand the relationship between these tables only.

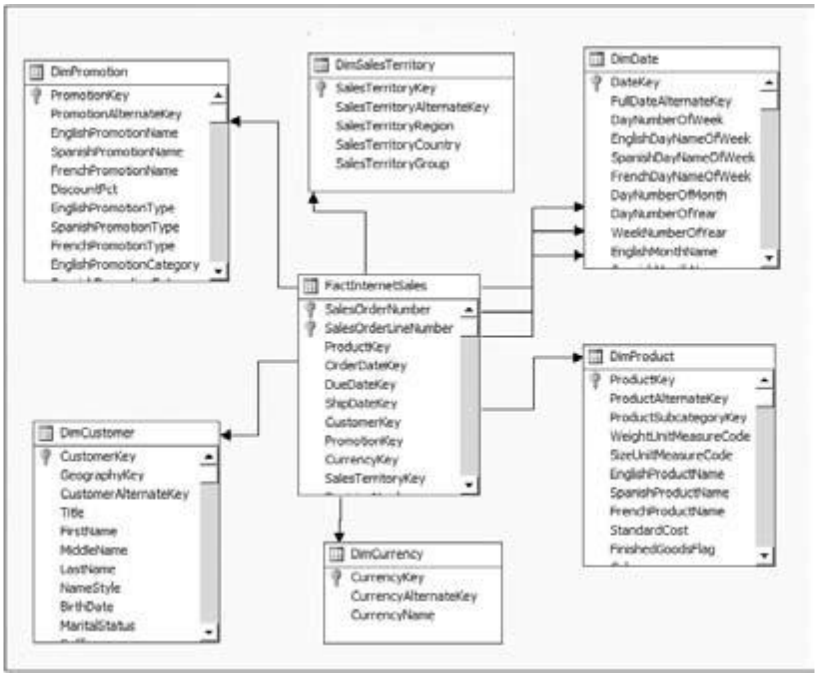


Figure 4-19

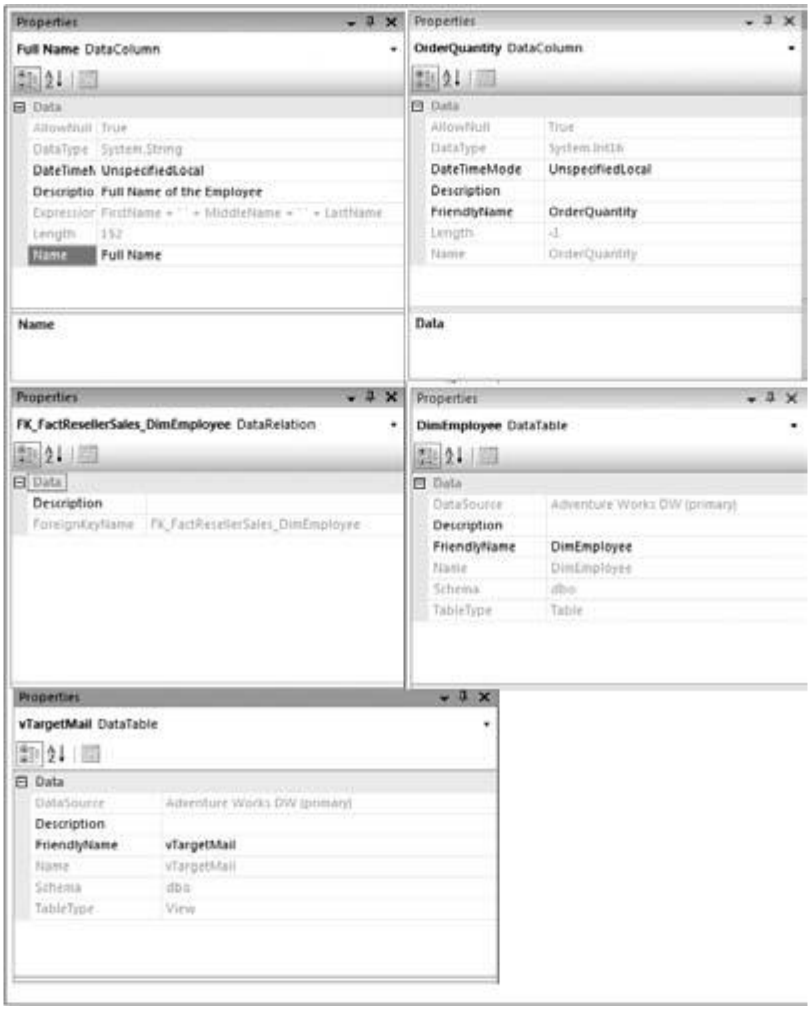
If you do not want a specific table in your diagram view you can right - click the table and select Hide. Instead of steps 3 and 4 in the preceding list you can add tables to the diagram view by dragging and dropping them from the Table pane to the Diagram pane. Create another diagram called Reseller Sales and add the FactResellerSales table and related tables.

## *Data Source View Properties*

Each object created within BIDS has certain properties. Within the DSV you can view the properties of the objects in the DSV such as tables, views, columns, and relationships. Properties of these objects are shown in the Properties window within BIDS, as shown in Figure 4 - 20 .

Figure 4 - 20 shows the properties of a column in a table, calculated column, a table, and a relationship. For the regular columns in a table you have the properties AllowNull, Data Type, Description, Friendly Name, Length, and Name. The properties of a column are populated by retrieving the corresponding property from the data source. The data type of this column is retrieved from the data source. Based on the properties defined in the data source, the properties AllowNull, Data Type, Length, Name, and Friendly Name are populated. The Length property is applicable only for the string data type. For all

other data types the Length property has a value of – 1. You cannot change certain properties; they are not editable in the Properties window and are grayed out. You can change the Friendly Name and provide a description to each column. Often columns of a table in the relational database might not have user - friendly names. User - friendly means the name of the column should indicate clearly or intuitively the data stored in the column. Friendly Name is a property that can be changed so that this friendly name is shown in the DSV for an easier understanding of the model. You can also provide an optional Description to each column if needed. The DSV Designer provides you with the option of switching between the original column names and the friendly names. You can right - click in the DSV diagram view and toggle between the friendly name and the original column name by selecting the Show Friendly Name option.



Properties

**Full Name** DataColumn

| Data | |
|---|---|
| AllowNull | True |
| DataType | System.String |
| DateTimeN | UnspecifiedLocal |
| Descriptio | Full Name of the Employee |
| Expressior | FirstName + ' ' + MiddleName + ' ' + LastName |
| Length | 152 |
| Name | Full Name |

Name

Properties

**OrderQuantity** DataColumn

| Data | |
|---|---|
| AllowNull | True |
| DataType | System.Int16 |
| DateTimeMode | UnspecifiedLocal |
| Description | |
| FriendlyName | OrderQuantity |
| Length | -1 |
| Name | OrderQuantity |

Data

Properties

**FK_FactResellerSales_DimEmployee** DataRelation

| Data | |
|---|---|
| Description | |
| ForeignKeyName | FK_FactResellerSales_DimEmployee |

Properties

**DimEmployee** DataTable

| Data | |
|---|---|
| DataSource | Adventure Works DW (primary) |
| Description | |
| FriendlyName | DimEmployee |
| Name | DimEmployee |
| Schema | dbo |
| TableType | Table |

Properties

**vTargetMail** DataTable

| Data | |
|---|---|
| DataSource | Adventure Works DW (primary) |
| Description | |
| FriendlyName | vTargetMail |
| Name | vTargetMail |
| Schema | dbo |
| TableType | View |

Figure 4-20

Named columns created in the DSV do not have a Friendly Name property because you will define the name to this column and we expect you to provide a name that is intuitive and understandable. Instead, named columns have the Expression property because each named column is created from a SQL expression. You can change this expression only in the Named Column dialog and not in the Properties window.

Tables have the properties Data Source, Description, FriendlyName, Name, Schema, and Table Type. The Data Source indicates the name of the data source of the Table. The Table Type shows whether the object in the underlying data source is a table or a view. Similar to the columns, tables also have the option to specify a friendly name. Relationships between tables are provided with a name that includes the tables that participate in the relationship. Similar to named columns, named queries do not have a Friendly Name property. They have a property called Query Definition that shows the query used to specify the named query object.

## Different Layouts in DSV s

The DSV Designer provides you with two layout types to view the tables in the DSV. When you create a

DSV the default layout type is rectangular layout. In the default layout, the lines representing the relationships between tables are composed of horizontal and vertical lines, and the lines emerge from any of the sides of the table. The second layout type offered by the DSV Designer is called diagonal layout. In diagonal layout, the tables are arranged in a way such that the lines showing the relationships between tables are originating at the end points of the table, so that these lines appear to be along the diagonal of the tables — hence the name " diagonal layout. " You can switch between rectangular layout and diagonal layout in the DSV by right - clicking in the DSV Designer and selecting the layout type of your choice. Figures 4 - 21 and 4 - 22 show rectangular and diagonal layout, respectively, of the Internet Sales diagram.

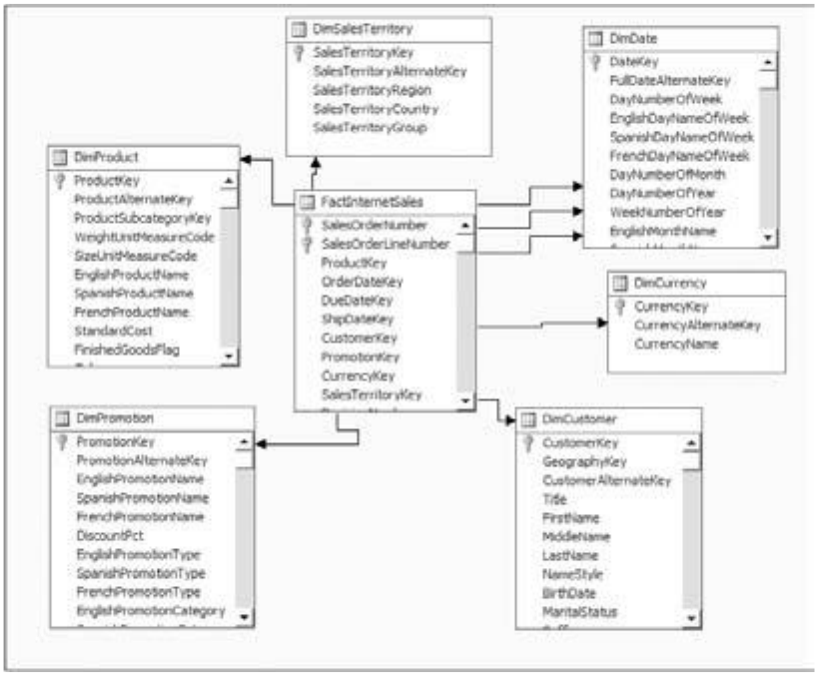


Figure 4-21

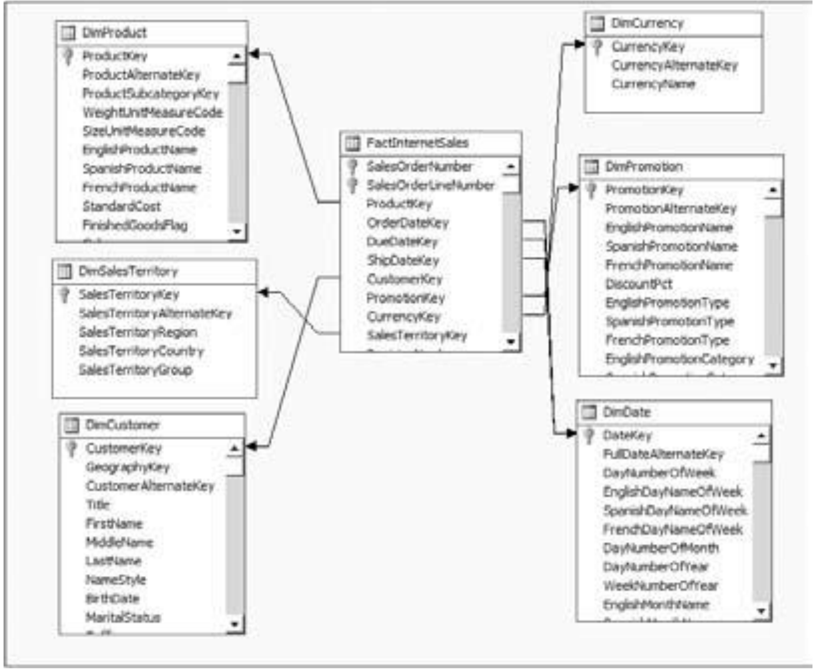


Figure 4-22

## Validating Your DSV and Initial Data Analysis

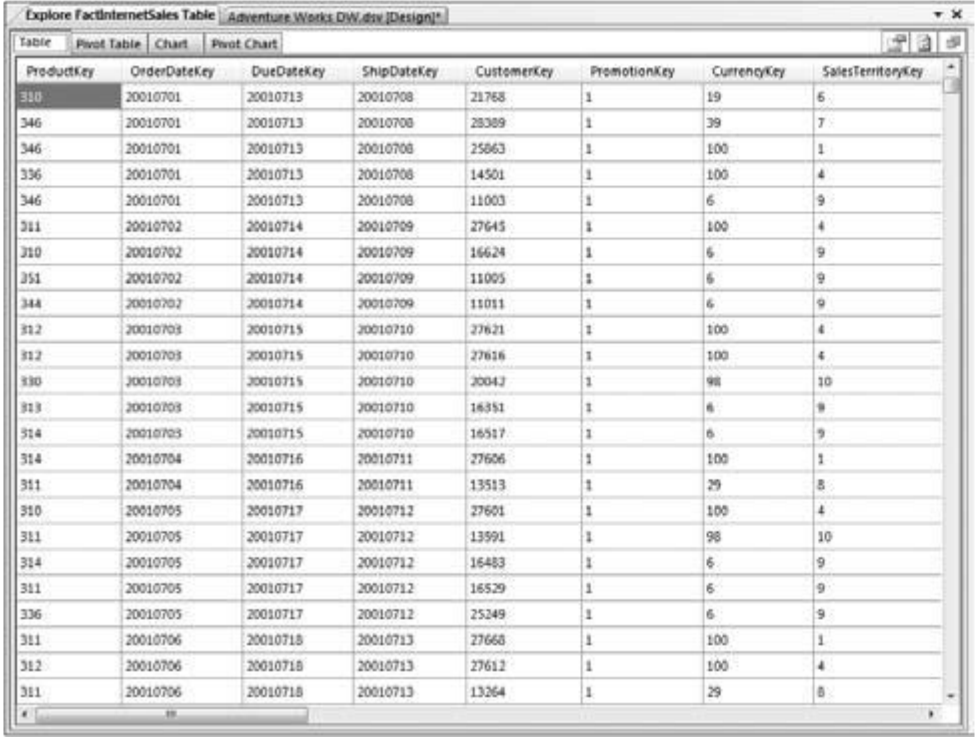
The relationships specified in the DSV will be used in creating your dimensions and cubes. Therefore, validating your DSV is crucial to your data warehouse design. The DSV Designer provides a first level of validation when you specify relationships. If the data types of column(s) involved in the relationship do not match, the DSV will not allow you to establish the relationship. This forces you to make sure you cast the data types of the column(s) involved in the relationships appropriately. You might need another level of validation by looking at the data within each table. You can do this by issuing queries to the tables in the relational data source. The DSV provides a way of looking at sample data for validation. A few validations you can do within the DSV by looking at sample data are as follows:

1. Looking at the fact table data helps you in making sure this table contains fact data, the primary key has been specified correctly, and appropriate relationships needed for dimensions are established.

2. Analyzing a dimension table ' s sample data ensures that you have all the relationships established between the fact and dimension tables and any relationships within each table are established correctly. For example, if you have an Employee table that contains an employee and his manager, you might want to establish a relationship so that your model can take advantage of this.

In addition, a sample of data from the tables in the DSV helps you in identifying the measures of the cube as well as the hierarchies of each dimension. Analyzing sample data in the DSV also helps you identify dimensions that can be created from the fact table data. The analysis of sample data within the DSV is even more important when creating Data Mining models. You learn more about analyzing the data with respect to Data Mining in Chapter 16 .

To see a sample of the data specified by your DSV, right - click a table in the DSV Designer and select Explore Data. You can now see rows from the underlying table presented within the Explore < tablename > Table window as shown in Figure 4 - 23 . The data presented is only a subset of the underlying table data. By default the first 5,000 rows are retrieved and shown within this window. You can change the number of rows retrieved by clicking the Sampling Options button. Clicking the Sampling Options button launches the Data Exploration Options dialog where you can change the sampling method, sample count, and number of states per chart, which is used for displaying data in the chart format. Once you have changed the sample count value you can click the Resample Data button to retrieve data based on the new settings. The Explore Table window has four tabs: Table, Pivot Table, Chart, and Pivot Chart. The Table tab shows the raw sampled data from the data source as rows and columns with column headings.


Explore FactInternetSales Table
Adventure Works DW.dsv [Design]*
Table
Pivot Table
Chart
Pivot Chart

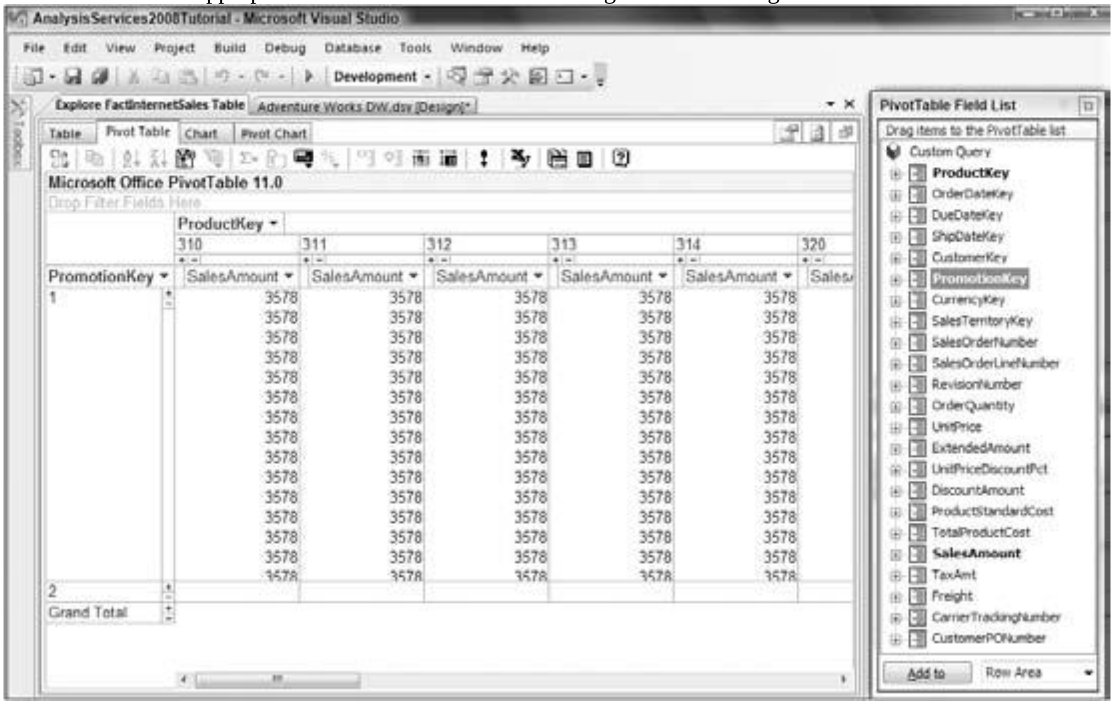
| ProductKey | OrderDateKey | DueDateKey | ShipDateKey | CustomerKey | PromotionKey | CurrencyKey | SalesTerritoryKey |
|---|---|---|---|---|---|---|---|
| 310 | 20010701 | 20010713 | 20010708 | 21768 | 1 | 19 | 6 |
| 346 | 20010701 | 20010713 | 20010708 | 28389 | 1 | 39 | 7 |
| 346 | 20010701 | 20010713 | 20010708 | 25863 | 1 | 100 | 1 |
| 336 | 20010701 | 20010713 | 20010708 | 14501 | 1 | 100 | 4 |
| 346 | 20010701 | 20010713 | 20010708 | 11003 | 1 | 6 | 9 |
| 311 | 20010702 | 20010714 | 20010709 | 27645 | 1 | 100 | 4 |
| 310 | 20010702 | 20010714 | 20010709 | 16624 | 1 | 6 | 9 |
| 351 | 20010702 | 20010714 | 20010709 | 11005 | 1 | 6 | 9 |
| 344 | 20010702 | 20010714 | 20010709 | 11011 | 1 | 6 | 9 |
| 312 | 20010703 | 20010715 | 20010710 | 27621 | 1 | 100 | 4 |
| 312 | 20010703 | 20010715 | 20010710 | 27616 | 1 | 100 | 4 |
| 330 | 20010703 | 20010715 | 20010710 | 20042 | 1 | 98 | 10 |
| 313 | 20010703 | 20010715 | 20010710 | 16351 | 1 | 6 | 9 |
| 314 | 20010703 | 20010715 | 20010710 | 16517 | 1 | 6 | 9 |
| 314 | 20010704 | 20010716 | 20010711 | 27606 | 1 | 100 | 1 |
| 311 | 20010704 | 20010716 | 20010711 | 13513 | 1 | 29 | 8 |
| 310 | 20010705 | 20010717 | 20010712 | 27601 | 1 | 100 | 4 |
| 311 | 20010705 | 20010717 | 20010712 | 13591 | 1 | 98 | 10 |
| 314 | 20010705 | 20010717 | 20010712 | 16483 | 1 | 6 | 9 |
| 311 | 20010705 | 20010717 | 20010712 | 16529 | 1 | 6 | 9 |
| 336 | 20010705 | 20010717 | 20010712 | 25249 | 1 | 6 | 9 |
| 311 | 20010706 | 20010718 | 20010713 | 27668 | 1 | 100 | 1 |
| 312 | 20010706 | 20010718 | 20010713 | 27612 | 1 | 100 | 4 |
| 311 | 20010706 | 20010718 | 20010713 | 13264 | 1 | 29 | 8 |

Figure 4-23

When you click the Pivot Table tab you get an additional window called PivotTable Field List that shows all the columns of the table, as shown in Figure 4 - 24 . You can drag and drop these columns inside the pivot table in the row, column, details, or filter areas. The values in the row and column provide you with an intersection point for which the detailed data is shown. For example, you can drag and drop CustomerKey, SalesTerritoryKey, and Sales Amount to the row, column, and detail data areas, respectively. The pivot table now shows you the sales amount of each product by each customer.

The pivot table actually allows you to view multidimensional data from a single table. You learn more about pivot tables in Chapter 17 .

Analysis Services analyzes the sample data, identifies the most important columns within the table, and provides you distributions in the Chart tab. The Pivot Chart tab provides you functionality similar to the Pivot Table tab but in a chart view. The Chart and Pivot Chart tabs are typically used to do an initial analysis of the data so that appropriate columns can be used to create good Data Mining models.



AnalysisServices2008Tutorial - Microsoft Visual Studio
File Edit View Project Build Debug Database Tools Window Help
Development
Explore FactInternetSales Table
Adventure Works DW.dsv [Design]*
Table
Pivot Table
Chart
Pivot Chart
Toolbox

Microsoft Office PivotTable 11.0

Drop Filter Fields Here

| | ProductKey | | | | | |
|---|---|---|---|---|---|---|
| | 310 | 311 | 312 | 313 | 314 | 320 |
| PromotionKey | SalesAmount | SalesAmount | SalesAmount | SalesAmount | SalesAmount | Sales |
| 1 | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| | 3578 | 3578 | 3578 | 3578 | 3578 | |
| 2 | | | | | | |
| Grand Total | | | | | | |

PivotTable Field List

Drag items to the PivotTable list

Custom Query
- **ProductKey**
- OrderDateKey
- DueDateKey
- ShipDateKey
- CustomerKey
- PromotionKey
- CurrencyKey
- SalesTerritoryKey
- SalesOrderNumber
- SalesOrderLineNumber
- RevisionNumber
- OrderQuantity
- UnitPrice
- ExtendedAmount
- UnitPriceDiscountPct
- DiscountAmount
- ProductStandardCost
- TotalProductCost
- **SalesAmount**
- TaxAmt
- Freight
- CarrierTrackingNumber
- CustomerPONumber

Add to: Row Area

Figure 4-24

# Multiple Data Sources within a DSV

Data warehouses usually consist of data from several data sources. Some examples of data sources are SQL Server, Oracle, DB2, and Teradata. Traditionally, the OLTP data is transferred from the operational data store to the data warehouse — the staging area that combines data from disparate data sources. This is not only time intensive in terms of design, maintainability, and storage but also in terms of other considerations such as replication of data and ensuring data is in sync with the source. SSAS 2008 helps you avoid this and gives you a better return on your investment.

The DSV Designer provides you with the capability of adding tables from multiple data sources, which you can then use to build your cubes and dimensions. You first need to define the data sources that include the tables that are part of your data warehouse design using the Data Source Wizard. Once this has been accomplished, you create a DSV and include tables from one of the data sources. This data source is called the primary data source and needs to be a SQL Server. You can then add tables in the DSV Designer by right - clicking in the diagram view and choosing Add/Remove Tables. You need to have a data source defined in your Analysis Services project to be able to add tables from it to the DSV. The Add/Remove Tables dialog allows you to choose a data source as shown in Figure 4 - 25 so that you can add its tables to the DSV. To illustrate the selection of tables from a second data source, as shown in Figure 4 - 25 , we have created a second data source in the AnalysisServices2008Tutorial project from the SQL Server Master database. You should be aware that there might be performance implications to retrieving data from secondary data sources because all the queries are routed through the primary data source.

Once you have added the tables from multiple data sources to your DSV, you can start creating your cubes and dimensions as if these came from a single data source. The limitation that the primary data source needs to be a SQL Server is due to the fact that Analysis Services instance uses a SQL Server – specific feature called OPENROWSET to retrieve data from other data sources.

## Summary

You now have the skills to deal with the challenges real - world data warehouses will throw at you in terms of multiple data sources. You learned about OLE DB and managed data providers that are used by SSAS 2008 to retrieve data from data sources and the trade - offs of using one versus another. Indeed, you learned to tame the disparate data source beast by using multiple data sources. Then you learned to consolidate the tables and relationships of interest in Data Source Views (DSVs), and finally, to refine the tables and relationships in the DSVs so you only have to deal with what ' s relevant.

Note that when key changes are made in the DSV that is where the changes stay — in the DSV. The changes are not written out to the underlying tables as you might expect. This is a good thing. To see why, take a look at the alternative to using the DSV capability. The alternative method is to create a view in SQL with real relationship transforms in the underlying tables. It ' s not that we strongly oppose this method, but if your data spans multiple databases, you may have to create linked servers and that can become time-consuming. SSAS 2008 provides an easy way to specify these cross - database relationships within a DSV without the overhead of having to use linked servers. However, when multiple data sources are included in a single DSV the primary data source should support the ability to send queries and retrieve results from other servers. You can incur performance degradation due to this method; however, you do have the flexibility of not having to manage the data on multiple servers to create your data warehouse.

You ' re doing great! In fact, you ' re now ready to tackle core business intelligence constructs like dimension design (Chapter 5 ) and cube design (Chapter 6 ). If you already know these topics from working with earlier versions of SQL Server Analysis Services, we recommend working through the chapters anyway. There have been some important changes to the Cube and Dimension Wizards in Analysis Services 2008.

# Dimension Design

Prior to the advent of cable, when you brought home a new television, the first order of business was to manually tune in one of the few existing local channels. To accomplish this you manipulated the dials, rabbit - ear antennae positioning, and other controls to eventually obtain an optimal picture, audio, and vertical hold configuration. The process of designing a data warehouse using Analysis Services 2008 is similar to that. Analysis Services 2008 provides you with various wizards that help you build the initial framework, just like the rotary tuner on the television got you close to the desired channel. With the basic infrastructure in place, some fine - tuning can optimize the initial framework to your needs. In fact, you saw this approach in the previous chapter when you learned about creating data sources and DSVs. Likewise, here you learn how to create dimensions using the Dimension Wizard and then use the Dimension Designer to fine - tune the dimension based on your business needs.

Cubes are made up of dimensions and measures, where the measures are aggregated along each dimension. Without an understanding of dimensions and how measures are aggregated along them, you can ' t create and exploit the power of cubes, so let ' s jump right into learning about building and viewing dimensions. Once the dimensions are created, they need to be added to the cube and the right relationship type between the fact data and dimension needs to be defined. Analysis Services 2008 supports six relationship types (no relationship, regular, fact, referenced, many - to - many, and data mining). You learn about relationship types in this chapter and Chapters 8 and 16 . In addition, you learn about the attributes and hierarchies that form an integral part of dimensions. You learn how to model the Time dimension and Parent - Child dimensions, which are different from regular dimensions and are found in many data warehouses. Finally, you learn how to process and browse dimensions.

## Working with the Dimension Wizard

Dimensions help you define the structure of your cube so as to facilitate effective data analysis. Specifically, dimensions provide you with the capability of slicing data within a cube, and these dimensions can be built from one or more dimension tables. As you learned in Chapter 1 , your data warehouse can be designed as a star or snowflake schema. In a star schema, dimensions are created from single tables that are joined to a fact table. In a snowflake schema, two or more joined dimension tables are used to create dimensions where one of the tables is joined to the fact table. You create both of these dimension types in this chapter.

You also learned in Chapters 1 , 2 , and 3 that each dimension contains objects called hierarchies. In Analysis Services 2008 you have two types of hierarchies to contend with: the attribute hierarchy, which corresponds to a single column in a relational table, and multilevel hierarchies, which are derived from two or more attribute hierarchies where each attribute is a level in the multilevel hierarchy. A typical example of an attribute hierarchy would be Zip Code in a Dim Geography dimension, and a typical example for a multilevel hierarchy would be Country - State - City - Zip Code also in a Geography dimension. In everyday discussions of multilevel hierarchies, most people leave off the “ multilevel ” and just call them “ hierarchies.” For the exercises in this chapter, you use the project you designed in Chapter 2 . If you don ’ t happen to have the project handy, you can download it from www.wrox.com . Regardless of whether you download, you will still need to add the Geography Dimension (dbo.DimGeography) to the DSV. To add this dimension to your DSV, follow these steps:

1. Double - click the DSV named “ AdventureWorksDW.dsv ” in Solution Explorer.

2. Click the Add/Remove Objects toolbar button (the top - left button in the DSV Designer toolbar) as shown in Figure 5 - 1 .

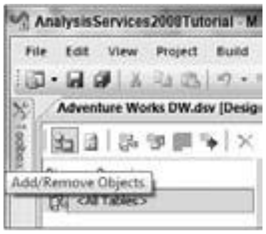
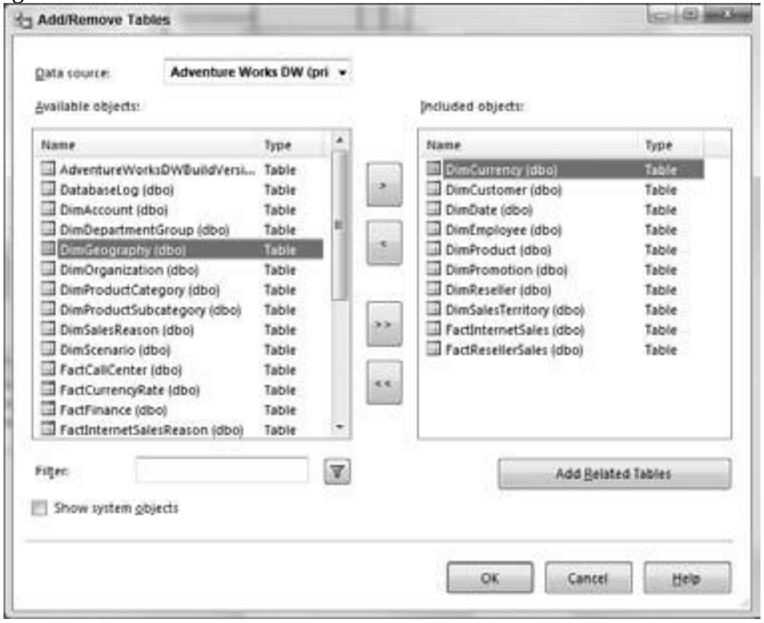

AnalysisServices2008Tutorial - M
File Edit View Project Build
Adventure Works DW.dsv [Desig
Add/Remove Objects
<All Tables>

Figure 5 - 1

3. In the Available objects list, select DimGeography and click the > (right arrow) button as shown in Figure 5 - 2 . This will move the Geography dimension into the Included objects list in the Add/Remove Tables dialog. Click OK to continue.


Add/Remove Tables

Data source: Adventure Works DW (pri

Available objects:

| Name | Type |
|---|---|
| AdventureWorksDWBuildVersi... | Table |
| DatabaseLog (dbo) | Table |
| DimAccount (dbo) | Table |
| DimDepartmentGroup (dbo) | Table |
| DimGeography (dbo) | Table |
| DimOrganization (dbo) | Table |
| DimProductCategory (dbo) | Table |
| DimProductSubcategory (dbo) | Table |
| DimSalesReason (dbo) | Table |
| DimScenario (dbo) | Table |
| FactCallCenter (dbo) | Table |
| FactCurrencyRate (dbo) | Table |
| FactFinance (dbo) | Table |
| FactInternetSalesReason (dbo) | Table |

Included objects:

| Name | Type |
|---|---|
| DimCurrency (dbo) | Table |
| DimCustomer (dbo) | Table |
| DimDate (dbo) | Table |
| DimEmployee (dbo) | Table |
| DimProduct (dbo) | Table |
| DimPromotion (dbo) | Table |
| DimReseller (dbo) | Table |
| DimSalesTerritory (dbo) | Table |
| FactInternetSales (dbo) | Table |
| FactResellerSales (dbo) | Table |

Filter:

□ Show system objects

Figure 5 - 2

4. All tables in the DSV should have the logical primary key set in order to allow Analysis Services to determine the key attribute columns if the table is used in a dimension or the foreign key columns in the case of fact tables. Review the DimGeography table in the DSV designer to make sure the DimGeography column has been set as the primary key as shown in Figure 5 - 3 .

If the primary key column is not retrieved for a table, you can right - click the key column in the table and select Set Logical Primary Key as shown in Figure 5 - 3 . If the primary key for the table has already been set, the option to Set Logical Primary Key will be disabled.

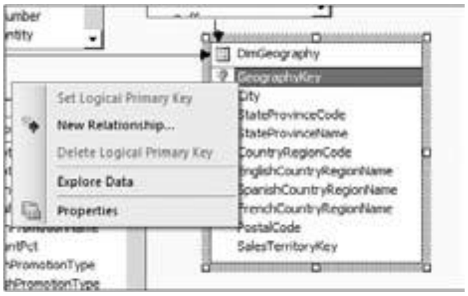


Figure 5 - 3

Now you are ready to explore use of the Dimension Wizard in Analysis Services 2008. Continue to follow these steps to create a Geography dimension.

5. Launch the Dimension Wizard by right - clicking Dimensions in the Solution Explorer and selecting New Dimension as shown in Figure 5 - 4 . If the welcome screen of the Dimension Wizard opens up, click Next.

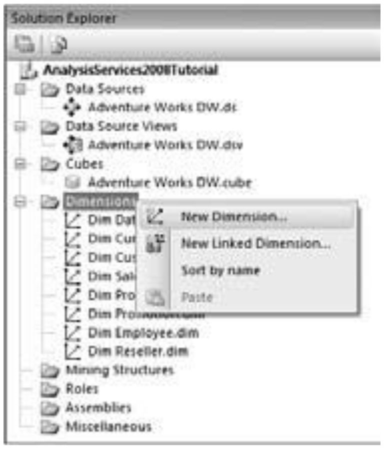


Figure 5 - 4

6. In the Select Creation Method screen, shown in Figure 5 - 5 , you will see four options:

Use an existing table

Generate a time table in the data source

Generate a time table on the server

Generate a non - time table in the data source

Using an existing table in the data source allows for the creation of a standard dimension, which can later be modified to become any sophisticated dimension type. This makes for a great generic starting point.

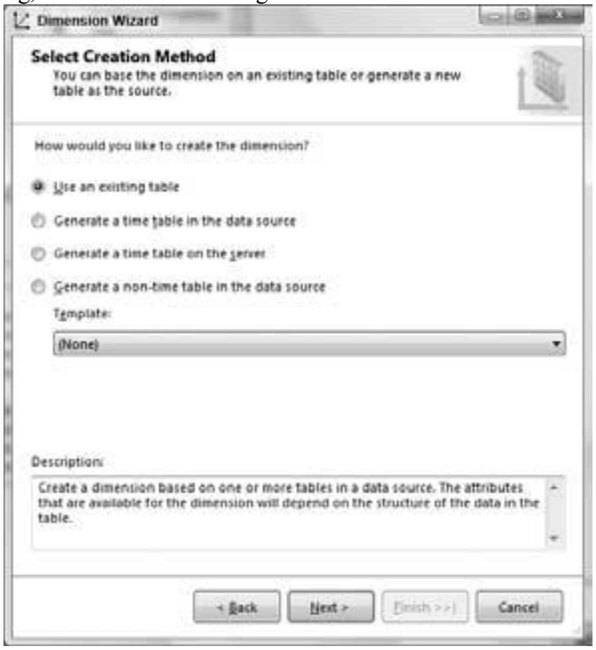
A Time dimension, on the other hand, is a unique type of dimension typically created from a table that contains time information such as year, semester, quarter, month, week, and date.

A Time dimension is unique because its members are fixed (a year always has 12 months in it) and typical business analyses are performed over time. Due to the uniqueness of the Time dimensions and how they are used in business analysis, there are special MDX functions that can be used with time dimensions. Furthermore, aggregation of data on a Time dimension does not have to be a garden variety additive aggregation like sum or count.

Most business decision makers want to analyze their data across a Time dimension to understand, for example, the month with maximum sales for a quarter or some other time period. Analysis Services provides you a distinct way to aggregate measure data across a Time dimension. This is done with semi - additive measures. You learn more about semi - additive measures in Chapters 6 and 9 . In a Time dimension, some standard hierarchies are commonly used, such as fiscal year and calendar year, both of which can be built automatically. The Time dimension can be built using either a table from the data source or without any associated tables in the data source. To create a table to serve as the source of your Time dimension, you choose the second option on the Creation Method screen. To create an Analysis Server – based Time dimension you would select the third option. You learn more about server Time dimensions in Chapter 8 .

In addition to Time, Analysis Services 2008 is also aware of several other common dimension types used in Business Intelligence applications such as Account, Customer, Employee, and

Currency, and can create the necessary tables in the data source to support these dimension types. In this chapter we ' re concerned with creating dimensions from a data source.
In the Select Creation Method dialog, select " Use an existing table " and click Next.



Dimension Wizard

**Select Creation Method**

You can base the dimension on an existing table or generate a new table as the source.

How would you like to create the dimension?

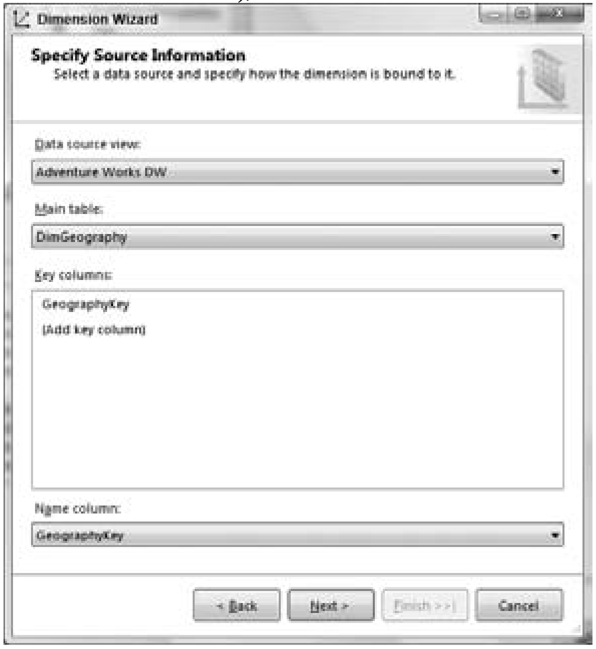
● Use an existing table
○ Generate a time table in the data source
○ Generate a time table on the server
○ Generate a non-time table in the data source

Template: (None)

Description:
Create a dimension based on one or more tables in a data source. The attributes that are available for the dimension will depend on the structure of the data in the table.

Figure 5 - 5

7. In the Specify Source Information page (shown in Figure 5 - 6 ), you need to select the DSV for creating the dimension, select the main table from which the dimension is to be designed, specify the key columns for the dimension, and optionally specify a name column for the dimension key value. By default, the first DSV in your project is selected. Because the current project has only one DSV (the Adventure WorksDW DSV), it is selected.



Dimension Wizard

**Specify Source Information**

Select a data source and specify how the dimension is bound to it.

Data source view: Adventure Works DW

Main table: DimGeography

Key columns:
GeographyKey
(Add key column)

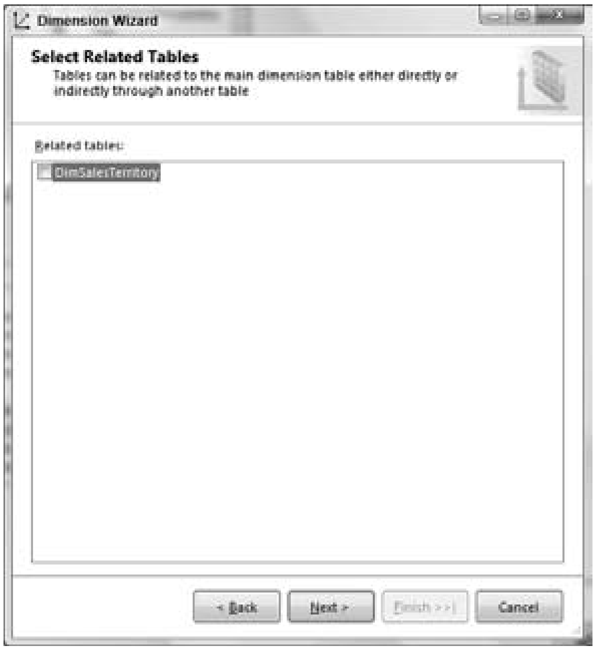
Name column: GeographyKey

Figure 5 - 6

In the Main Table listbox on the screen, you need to select the main table from which the dimension is to be designed. If a dimension is to be created from a star schema, the dimension is created from the single pertinent table. A snowflake schema dimension actually contains several tables, one of which is the primary table of the dimension. This primary table is chosen as the main table in the Main Table selection of the Dimension Wizard.

Select the DimGeography table from the Main table drop - down list as shown in Figure 5 - 6 . After selecting the Main Table for the dimension, the Key Columns list and the Name Column combo box will automatically be set to the logical primary key of the selected table. If no logical primary key is set, you will need to specify the key columns yourself. If the logical key for the main dimension table is a collection of columns, you must select a single column for the Name Column before proceeding. You have already defined a logical primary key column for the DimGeography table in the DSV.
Click the Next button to proceed to the next step in the Dimension Wizard.

8. The Dimension Wizard now analyzes the DSV to detect any outward - facing relationships from the DimGeography table. An outward - facing relationship is a relationship between the DimGeography table and another table, such that a column in the DimGeography table is a foreign key related to another table. The Select Related Tables screen (see Figure 5 - 7 ) shows that the wizard detected an outward relationship between the DimGeography table and the DimSalesTerritory table. In this example you will be modeling the DimGeography table as a star schema table instead of snowflake schema. Deselect the DimSalesTerritory table and click Next.



Select Related Tables

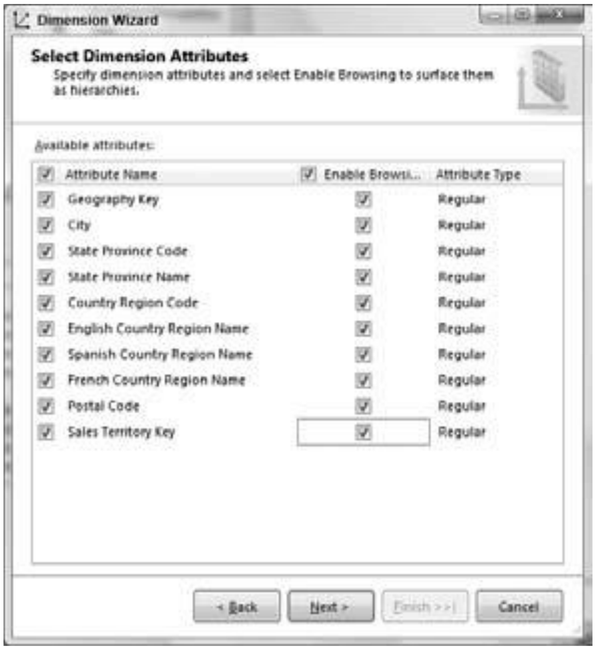
Tables can be related to the main dimension table either directly or indirectly through another table

Related tables:

□ DimSalesTerritory

Figure 5-7

9. The Select Dimension Attributes screen of the Dimension Wizard (see Figure 5 - 8 ) displays the columns of the main table that have been selected for the dimension you ' re creating. Each selected column on this screen results in an equivalent attribute being created in the new dimension. Even though you are building just a dimension here, that dimension is going to be part of a cube, which is described by the Unified Dimensional Model (UDM). The UDM combines the best of the relational and OLAP worlds. One important aspect of the relational model is the ability to query each column for reporting purposes. The columns selected from the relational table are transformed to attributes of a dimension that can then be used for querying from the UDM. You can control which of the attributes are available for browsing and querying by checking or un - checking the Enable Browsing option for each attribute. In addition, you can set the Attribute Type property to allow Analysis Services to provide special functionality based on the attribute ' s type and/or Analysis Services client tools to utilize this property to provide appropriate navigational paths. By default the wizard assigns the column name as the attribute name. You can change the attribute name for each selected attribute in this page.
In the Select Dimension Attributes screen, select all the attributes of the DimGeography table (all the attributes in the screen), leave their Attribute Type as Regular, allow them to be browsed as shown in Figure 5 - 8 , and click Next.



**Select Dimension Attributes**

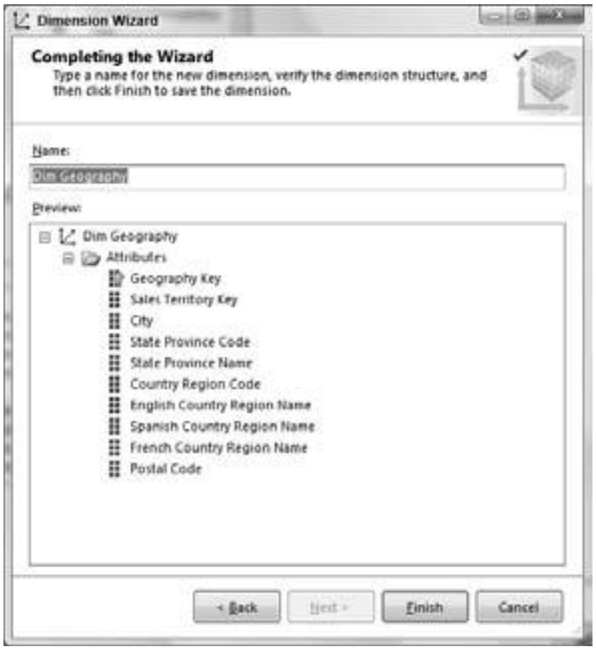
Specify dimension attributes and select Enable Browsing to surface them as hierarchies.

Available attributes:

| ☑ Attribute Name | ☑ Enable Browsi... | Attribute Type |
|---|---|---|
| ☑ Geography Key | ☑ | Regular |
| ☑ City | ☑ | Regular |
| ☑ State Province Code | ☑ | Regular |
| ☑ State Province Name | ☑ | Regular |
| ☑ Country Region Code | ☑ | Regular |
| ☑ English Country Region Name | ☑ | Regular |
| ☑ Spanish Country Region Name | ☑ | Regular |
| ☑ French Country Region Name | ☑ | Regular |
| ☑ Postal Code | ☑ | Regular |
| ☑ Sales Territory Key | ☑ | Regular |

Figure 5-8

10. The final screen of the Dimension Wizard shows the attributes that will be created for the dimension based on your choices in the wizard (see Figure 5 - 9 ). Click the Finish button. The wizard has created the dimension object Dim Geography and opened it up in the Dimension Designer. Congratulations!!! You have successfully created your first dimension using the Dimension Wizard. Next, you learn how to use the Dimension Designer to enhance the dimension to fit your business needs.



**Completing the Wizard**

Type a name for the new dimension, verify the dimension structure, and then click Finish to save the dimension.

Name:

Dim Geography

Preview:

- Dim Geography
  - Attributes
    - Geography Key
    - Sales Territory Key
    - City
    - State Province Code
    - State Province Name
    - Country Region Code
    - English Country Region Name
    - Spanish Country Region Name
    - French Country Region Name
    - Postal Code

Figure 5-9

The wizard has created the dimension object Dim Geography and opened it up in the Dimension Designer. Congratulations!!! You have successfully created your first dimension using the Dimension Wizard. Next, you learn how to use the Dimension Designer to enhance the dimension to fit your business needs.