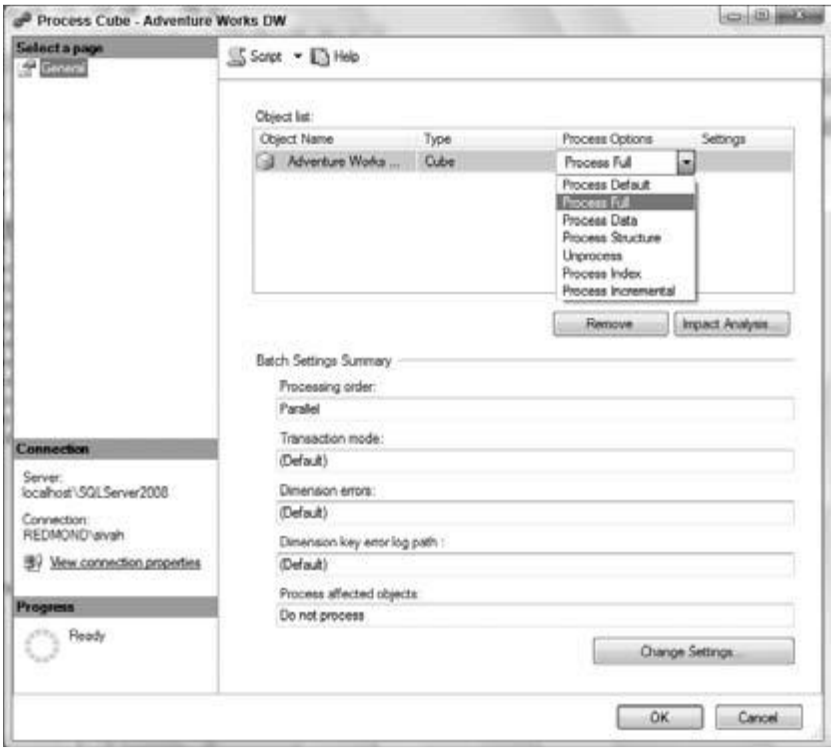
Лекция 14

Тема « Processing a Cube »

Processing a Cube

An Analysis Services database can contain several cubes and dimensions. You have the flexibility to control the processing of individual cubes and dimensions by launching the Process dialog from appropriate cube or dimension objects. There are several processing options for processing a cube, as shown in Figure 7 - 14 . All of the same processing options available for partitions and measure groups are available for the cube because a cube is a collection of measure groups, which in turn is a collection of partitions.



When a cube is created you will typically do a full process (*Process Full* in the Process dialog) of it so that you can browse the cube. Usually the cube structure will not change after the initial design is completed. In this case, you will be processing in order to get additional fact data that you would want to add to the cube. For example, you might have a Sales cube that you have created and you might be getting sales fact data from each store every month. Processing the entire cube whenever new data comes in will take a considerable amount of time, causing end users to have to wait for a long period to see the most up - to - date data. Analysis Services 2008 provides you with an option to process only the new fact data instead of the entire cube. This is called incremental processing. In order to add new fact data to the cube you can add a new partition to the cube and process that partition. Alternately, you can use the *Process Incremental* option in the Process dialog and specify the query that provides the new fact data that needs to be processed. Process Incremental is a common management task for data warehouses. If you specify the *Process Default* option in the Process dialog, the server checks for all the objects that have not been processed and only processes those objects. If the cube data has been processed and if aggregations and indexes are not processed, then those are processed.

When you choose the *Process Full* option for processing a cube, the server performs three internal operations. If the storage mode for the cube is MOLAP, the server first reads the data from the relational data and stores it in a compact format. If there were aggregations defined for the cube, the server will build those aggregations during this processing. Finally, the server creates indexes for the data that helps speed access to data during querying. Even if there were no aggregations specified for the cube, the

server still creates the indexes. The *Process Data* option actually is the first step of the Process Full option where the server reads data from relational data sources and stores it in proprietary format. The second and third steps of processing aggregations and indexes can be separately accomplished by the *Process Index* option. You might be wondering why you have the Process Data and Process Index options when the Process Full and Process Default options actually accomplish the same task. These options provide the administrator with a fine grain of control. These are especially important when you have limited time to access the relational data source and want to optimize the processing on your machine. Having multiple processing operations running in parallel can require more system resources. Specifically on a 32 - bit (X86 machines) system, a large cube that fails on Process Full may be able to be successfully processed by sending Process Data and Process Index commands one after another. In such instances, we recommend you first get the data from your relational backend into SSAS using the Process Data option. Once you have all the data in the Analysis Services instance, you can then create your aggregations and indexes, which do not need access to the relational data source.

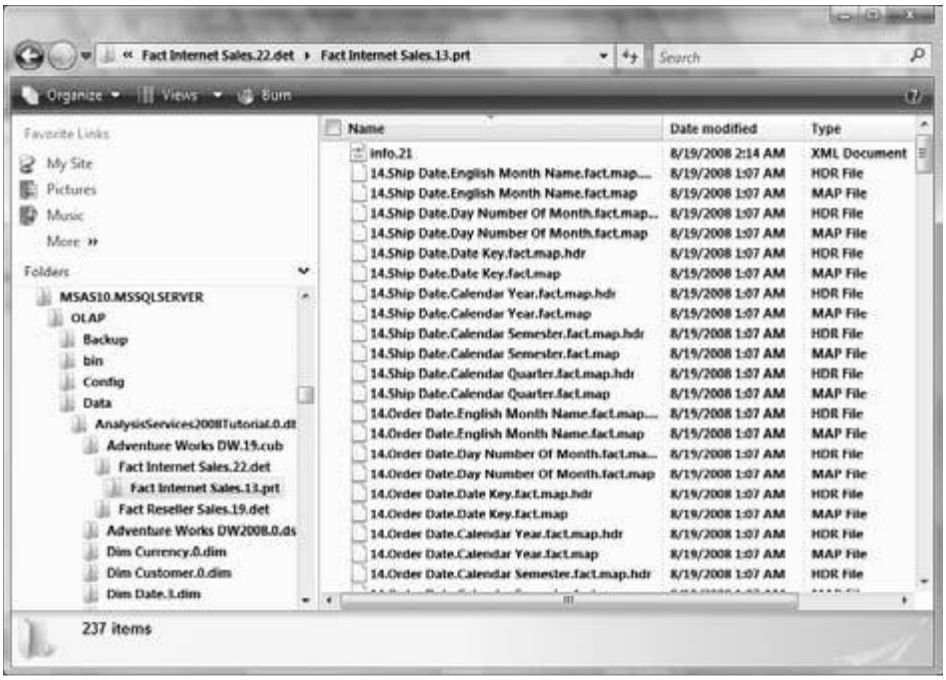
If you choose the *Process Structure* option, the server processes all the cube ' s dimensions and the cube definitions so that the cube ' s structure is processed without any processing of the data. The server will not process the partitions or measure groups of the cube, therefore you cannot see any of the fact data; however, you can browse the cube because the cube definitions are processed. You can retrieve metadata information about the cube (measure names, measure groups, dimensions, KPIs, actions, and so on) after processing the cube ' s structure. However, you will not be able to query the cube data. For a cube that has been processed with Process Structure, you can see the cube in the SQL Server Management Studio MDX query editor when you select the drop - down list for the cube. If your cube contains linked measure groups and if they have been processed successfully, processing the cube with the Process Structure option will allow you to query the measures in linked measure groups. Often when you design your UDM you will want to make sure your design is correct and your customers are able to see the right measures and dimensions. Process Structure is helpful in validating your design. As soon as the data for the cube is available the cube can be processed with the Process Default option so that end users can query the data from the cube.

You can clear the data in the cube using the *Unprocess* option. The processing options provided in the Process dialog are different than the process types that are specified in the process command sent to Analysis Services. The following table shows how the various processing options map to the process types sent to Analysis Services:

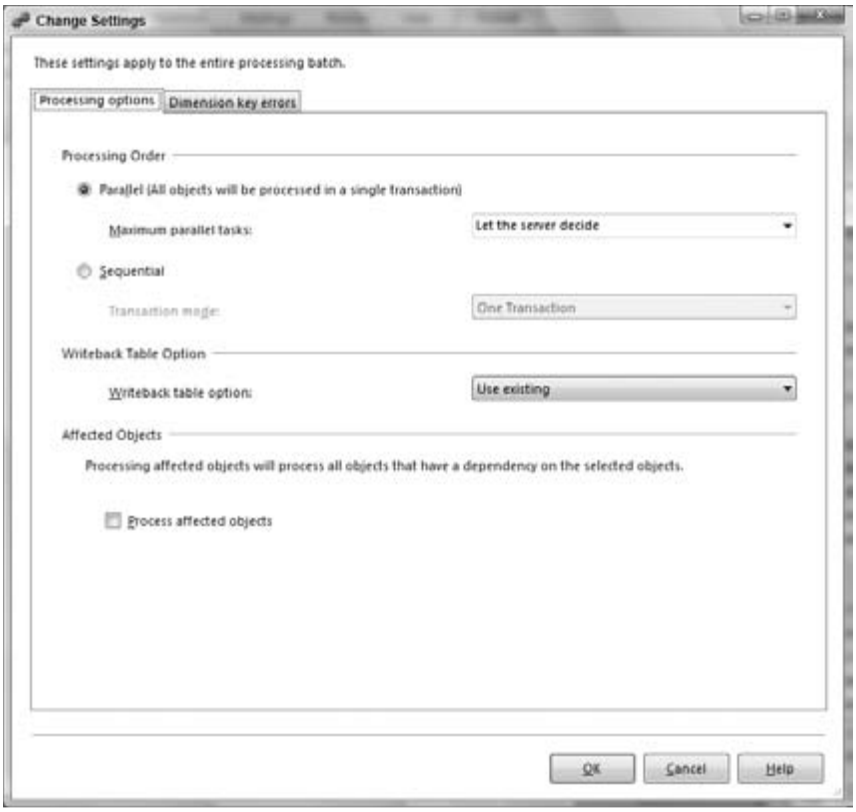
Process Options in Process Dialog Process Type in Process Command

Process Full ProcessFull
Process Default ProcessDefault
Process Data ProcessData
Process Structure ProcessStructure
Unprocess ProcessClear
Process Index ProcessIndexes
Process Incremental ProcessAdd
Process Script Cache ProcessScriptCache

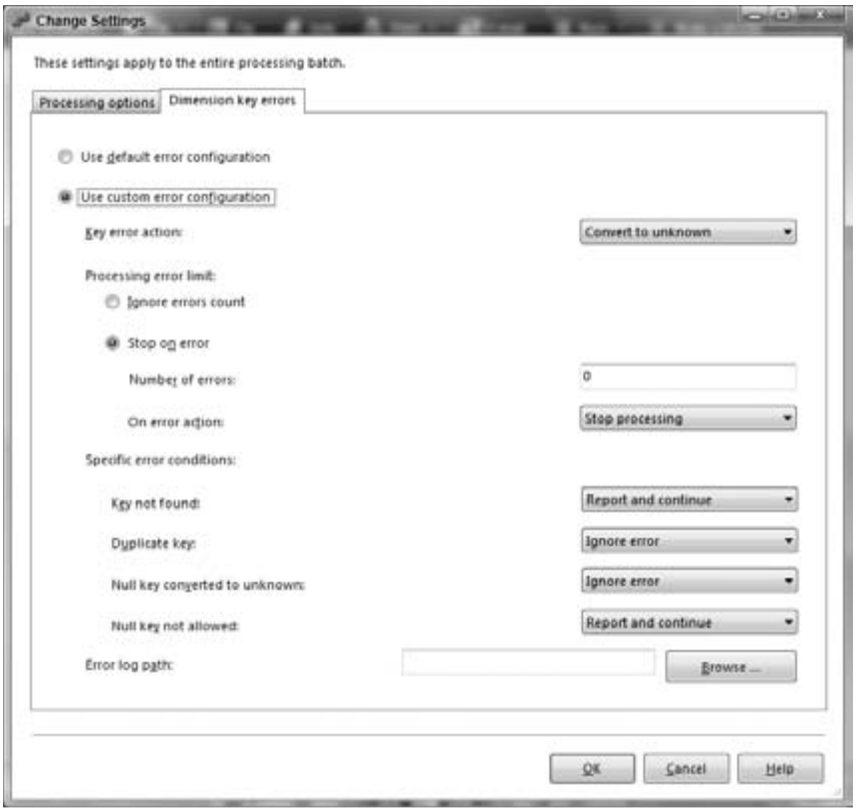
The processed data of a cube are stored in a hierarchical directory structure that is equivalent to the structure you see in the Object Explorer. Figure 7 - 15 shows the directory structure of the processed data of the AnalysisServices2008Tutorial database in Analysis Services 2008. The directory also shows the files within a partition. The metadata information about the cubes and dimensions are stored as XML files, and the data is stored in a proprietary format. Every time an object is processed, a new version number is appended to the object. For example, the files shown in Figure 7 - 15 are under a specific partition directory. The file info. < versionnumber > .xml is used to store the metadata information about the partition. Similar metadata files are stored within the directories of each object, cube, dimension, and measure group. We recommend you browse through each object folder to see the metadata information. The fact data is stored in the file with extension .data. The key to an OLAP database is the fast access to data. You learned about a cell, which was represented by a tuple. A tuple is the intersection of various dimension members. For fast data access, Analysis Services builds indexes to access data across multiple dimensions. The index files in Analysis Services have the extension " map". In Figure 7- 15 you can see the .map files that have the format < version>.< Dimension>.< Hierarchy>.fact.map. There is an associated header file for each map file. Analysis Services stores the data as blocks called segments for fast access. The associated header file contains offsets to the various segments for fast access during queries.



The processing dialog provides you the flexibility of processing objects in parallel or within the same transaction. If errors are encountered during processing, you can set options to handle these errors. You can configure the parallelism and error options by selecting the Change Settings button in the Process dialog. You will see the Change Settings dialog as shown in Figure 7 - 16 , which enables you to configure certain processing options and error settings during processing. Setting the parallelism option is as simple as selecting the appropriate option in the Processing Order section of the dialog. By default all the objects are processed in parallel and within the same transaction. If you do want failure of one object to impact other objects, you should process the objects under different transactions by choosing the sequential option.

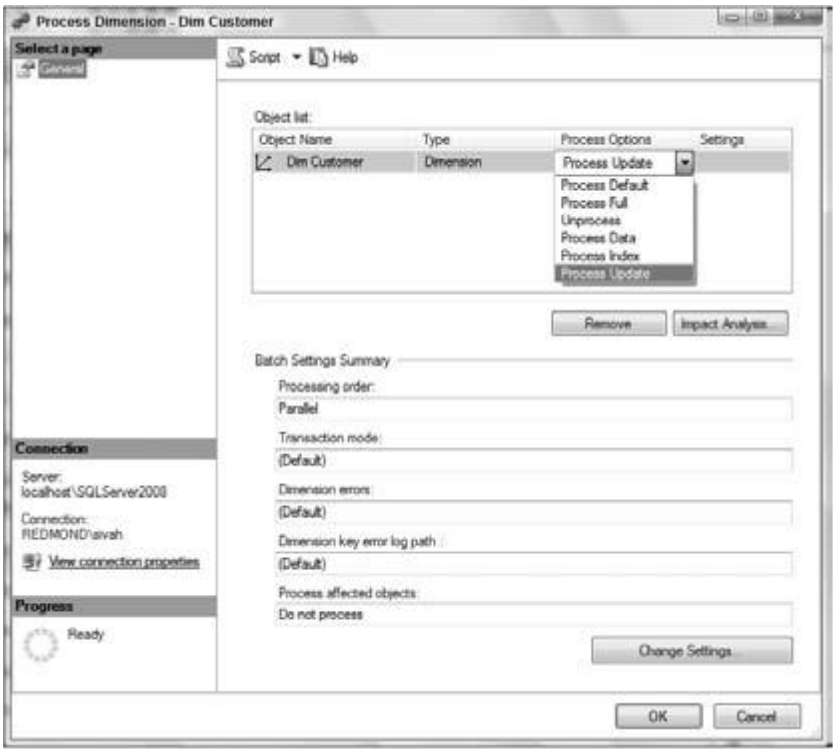


You might encounter errors while processing your Analysis Services objects due to incorrect design or referential integrity problems in the relational data source. For example, if you have a fact record that contains a dimension id that is not available in the dimension table, you will see a " Key not found " error while processing the cube. By default, when an error is encountered during processing, the processing operation fails. You can change the settings in the processing dialog to take appropriate action other than failing the processing operation. The Dimension Key Errors page of the Change Settings dialog shown in Figure 7 - 17 allows changing the error configuration settings for all the objects selected for processing. Whenever you encounter key errors you can either convert the values to unknown or discard the erroneous records. You can run into key errors while processing facts or dimensions. If you encounter a key error while processing a cube, that means Analysis Services was unable to find a corresponding key in the dimension. You can assign the fact value to a member called the Unknown Member for that specific dimension. You can encounter key errors while processing a snowflake dimension when an attribute defined as a foreign key does not exist in the foreign table or when there are duplicate entries. The two most common types of key errors that you might encounter during dimension processing are key not found and duplicate key errors.

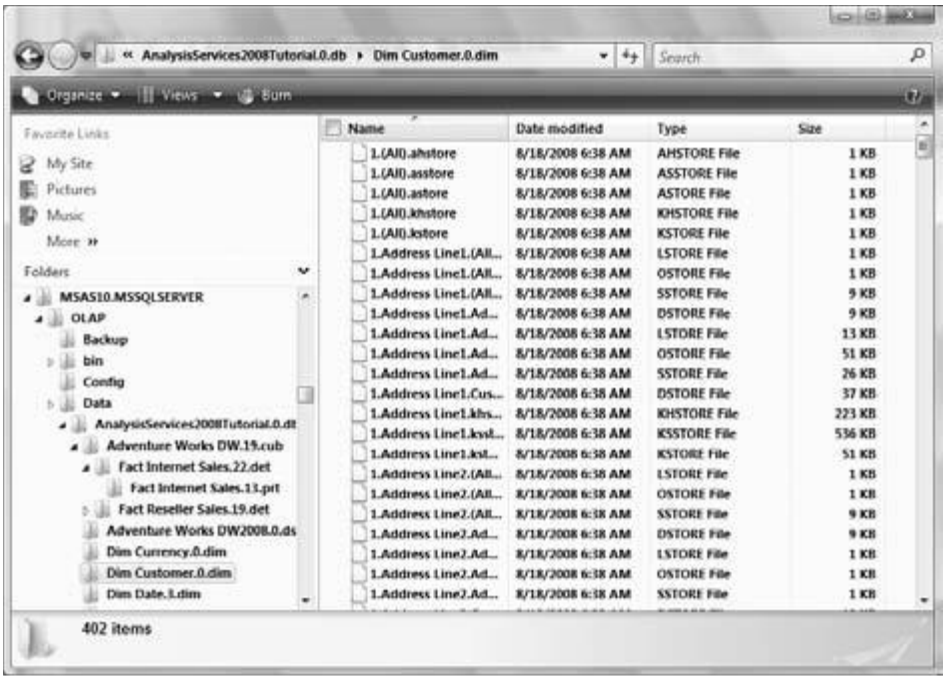


Processing a Dimension

You can process dimensions independent of the cubes they are a part of. After the initial processing of a dimension, you might process the dimensions on a periodic basis if additional records are added in the dimension table or there were changes to columns of an existing row. An example of additions to a dimension is new products being added to the products dimension. You would want this information to be reflected in the dimensions so that you can see the sales information for the new products. Another example of changes in dimension is when an employee moves from one city to another city; the attributes of the employee will need to change. Therefore the Process dialog provides you with various options for processing the dimension, as shown in Figure 7 - 18 .



While processing a dimension, Analysis Services reads data from the dimensions tables. When a dimension is processed, each attribute of the dimension is processed separately. Based on the parallelism specified on Analysis Services, these attributes can be processed in parallel. Each dimension contains an attribute called the All attribute. This is not exposed to the user but used internally by Analysis Services. You can see the files associated with this attribute as < version > .(All). < extension > in Figure 7 - 19 . When each attribute is processed, several files are created. Similar to fact data, dimension data is stored in a proprietary format. Each attribute of a dimension has a key column and a named column. These directly map into two different files with the extensions kstore and sstore, which refer to key store and string store, respectively. In addition, there are additional files that get created for each attribute of the dimension, which help in fast access to name, key, and levels of attributes and hierarchies. Files with the extension .map are created when indexes are processed for each attribute and help in fast retrieval of related attributes of the dimension for a dimension member.



The amount of time it takes to process a dimension depends on the number of attributes and hierarchies in the dimension as well as the number of members in each hierarchy. When a processing command is sent to the Analysis Services instance, the server reads the data from the relational data source and updates the dimension. When a dimension is processed, each attribute of the dimension is processed separately. Some attributes can be processed in parallel, whereas some cannot. The order of processing of various attributes is dependent on the relationships between the attributes in the dimensions and resources available on the machine. The relationships between attributes are defined at the dimension design time using the Attribute Relationships tab of the dimension designer, which you learned about in Chapter 5 . For example, say you have a Customer dimension that contains the attributes Customer Name, SSN, City, State, and Country. Assume SSN is the Key attribute for this dimension and by default all attributes within the dimension are related to the key attribute. In addition, assume additional attribute relationships have been established. They are Country State, State City, City Customer Name, State Customer Name, and Country Customer Name. Based on the preceding relationships, the order of processing of the attributes in the Customer dimension is Country, State, City, Customer Name, and SSN. This is because Analysis Services needs to have information about Country in order to establish the member property relationship while processing the State, Customer Name, or SSN. The key attribute is always the last attribute processed within a dimension.

When the Process Default option is chosen for processing, the dimension ' s data or indexes are processed if they have not been processed or are out-of-date. If the Process Full option is chosen, the entire dimension is re - processed. When the Process Full option is used, dimension data and indexes that have been processed initially will be dropped and data is retrieved from the data source. The dimension processing time depends on the dimension size (number of dimension members as well as number of attributes and hierarchies in the dimension) and your machine resources.

Similar to incremental processing of the cubes you can incrementally process dimensions using the *Process Update* option. The *Process Update* option in the Process dialog maps to the *ProcessUpdate* process

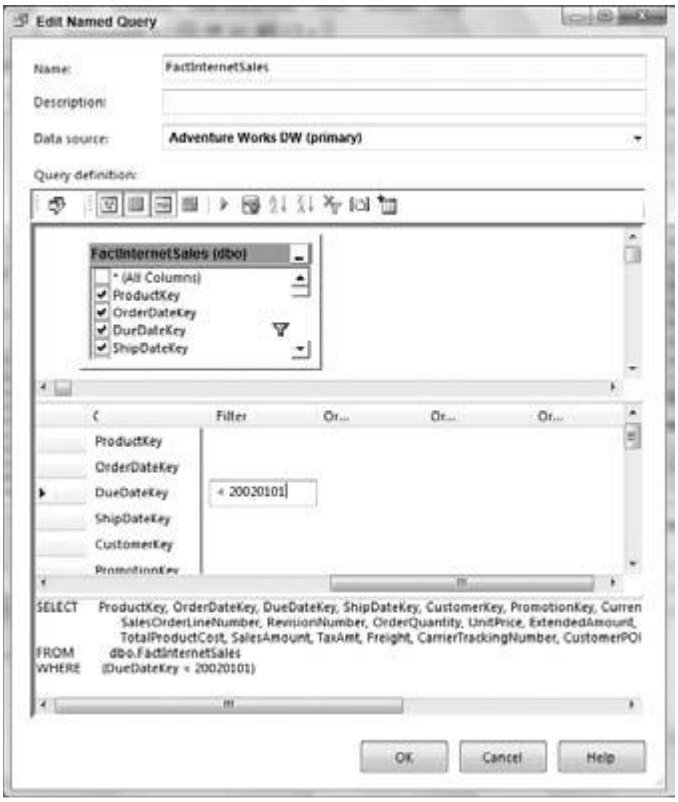
or Customers or Products can potentially contain a large number of members. Additional members may have been added to these dimensions or some attributes of these dimension members might have changed. Often a full processing of any dimension is not only unnecessary but cannot be afforded due to business needs. Under these circumstances incremental processing of the dimension or an update of the attributes of the dimension should be sufficient. When you choose the *Process Update* option for the dimension, the server scans all the dimensions in the dimension table. If there were changes to the dimension ' s properties, such as caption or description, they are updated. If new members are added to the dimension table, these members are added to the existing dimension using incremental processing. The attributes of each dimension member will also be updated. The key of each dimension member is assumed to be the same, but expect some attributes to be updated. The most important attribute that is updated is the member property for each member. When you have a parent - child

hierarchy in a dimension and if the parent attribute has been changed, that information is updated during the Process Update processing option.
The Process Data option for dimensions is used to process the dimension data. The indexes will not be processed when the Process Data option is used. The Process Index option is used to create indexes for attributes in the dimensions. If the ProcessMode dimension property is set to LazyAggregations, Analysis Services builds indexes for new attributes of the dimension as a lazy operation in the background thread. If you want to rebuild these indexes immediately you can do so by choosing the Process Index option. The Unprocess option is used to clear the data within the dimension.

Managing Partitions

Partitions enable you to distribute fact data within Analysis Services and aggregate data so that the resources on a machine can be efficiently utilized. When there are multiple partitions on the same server, you will reap the benefits of partitions because Analysis Services reads/writes data in parallel across multiple partitions. Fact data on the data source can be stored as several fact tables — Sales_Fact_2002, Sales_Fact_2003, and so on — or as a single large fact table called Sales Fact. You can create multiple partitions within a measure group; one for each fact table in the data source or by splitting data from a single large fact table through several queries. Partitions also allow you to split the data across two or more machines running Analysis Services, which are called Remote partitions. As an administrator you might be thinking what the size of each partition should be to achieve the best results. Microsoft recommends each partition to be 3 – 5GB or 20 million records. You learn more about optimizing partitions in Chapter 14 .
A sales cube ' s partitions usually contain data spread across time, that is, a new partition might be created for every month or a quarter. As an administrator you would create a new partition from SQL Server Management Studio and process it so that it is available for users. To create a new partition, perform the following steps in BIDS:

1. Open the AnalysisServices2008Tutorial project you have used in previous chapters.
2. Change the FactInternetSales table to a named query so that there is a where condition DueDateKey < 20020101. In case you don ' t recall how this is done, we ' ve included the steps here:
   a. Open Adventure Works DW.dsv under the Data Source Views folder.
   b. Right - click the FactInternetSales table in diagram view and select Replace Table With New Named Query menu item.
   c. In the Create Named Query dialog, in the DueDateKey Filter text entry box, enter < 20020101. Your change will automatically be reflected in the query window as shown in Figure 7 - 20 . Click OK to continue.



3. In the DSV, right - click in the diagram view and select Add/Remove Tables from the context menu.
4. Add the FactInternetSales table to " Included objects: " list and click OK.
5. In the diagram view, replace the FactInternetSales table with a named query.
6. In the named query, set Filter to DueDateKey > =20020101.
7. Rename the named query as FactInternetSalesNew.
8. Deploy the AnalysisServices2008Tutorial project to your Analysis Services instance.
9. Connect to the AnalysisServices2008Tutorial database using SSMS.
10. Navigate to the measure group FactInternetSales.
11. Right - click the Partitions folder and select New Partition as shown in Figure 7 - 21 .

12. Click Next on the welcome screen of the Partition Wizard.

13. Choose the named query FactInternetSalesNew to create a new partition as shown in Figure 7 - 22 and click Next. Select the checkbox " Specify a query to restrict rows " . As suggested by the warning in the Restrict Rows page (Figure 7 - 23) you may need to specify a restriction on the query to filter appropriate data for a partition. In this example FactInternetSalesNew already has the appropriate query restriction.

14. Click the Next button.

15. One way Analysis Services provides scalability is by use of remote partitions, where the partitions reside in two or more Analysis Services instances. On the Processing and Storage Locations page, as shown in Figure 7 - 24 , you can specify where to store the partition. You can specify the remote Analysis Services instance on this page, but the data source to the remote Analysis Services instance should have been defined in this database. You can also change the storage location where you want the data for the partition to reside on any of the Analysis Services instances. Choose the default options as shown in Figure 7 - 24 and click Next.

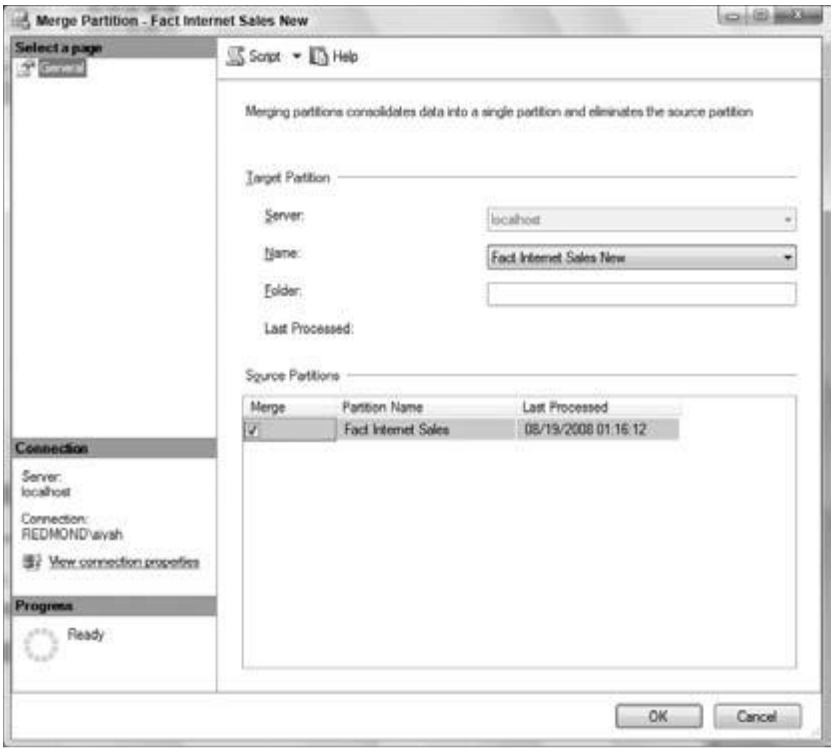
16. In the final page of the Partition Wizard, select Design aggregations later, Process Now as shown in Figure 7 - 25 and click Finish.

17. In the Process Partition dialog, click OK to process the FactInternetSalesNew partition.

The partition will be processed and you can browse the cube data. The number of partitions for a specific cube typically increases over time. Users might not be browsing historical data with the same granularity as that of the recent data. For example, you might be more interested in comparing Sales data for the current month to that of the previous month rather than data from five years ago. However, you might want to compare year - over - year data for several years. By merging the partition data you can see some benefits during query performance. You learn about the considerations you should take into account to merge partitions in Chapter 14 .

There are two main requirements to merge partitions: The partitions should be of the same storage type, and they need to be on the same Analysis Services instance. Therefore if you have remote partitions, they can be merged together only if they are on the same Analysis Services instance. To merge partitions, do the following:

1. Launch the Merge Partition dialog by right - clicking the Partitions folder under the Fact Internet Sales measure group.
2. In the Merge Partition dialog shown in Figure 7 - 26 , select the Target partition that will contain the merged data and the list of partitions to merge data and click OK.



deleted due to this operation. SSMS sends the following command to Analysis Services to merge the partitions:

```
< MergePartitions xmlns="http://schemas.microsoft.com/analysisservices/
2003/engine" >
< Sources >
< Source >
< DatabaseID > AnalysisServices2008Tutorial < /DatabaseID >
< CubeID > Adventure Works DW < /CubeID >
< MeasureGroupID > Fact Internet Sales < /MeasureGroupID >
< PartitionID > Fact Internet Sales < /PartitionID >
< /Source >
< /Sources >
< Target >
< DatabaseID > AnalysisServices2008Tutorial < /DatabaseID >
< CubeID > Adventure Works DW < /CubeID >
< MeasureGroupID > Fact Internet Sales < /MeasureGroupID >
< PartitionID > Fact Internet Sales New < /PartitionID >
< /Target >
< /MergePartitions >
```