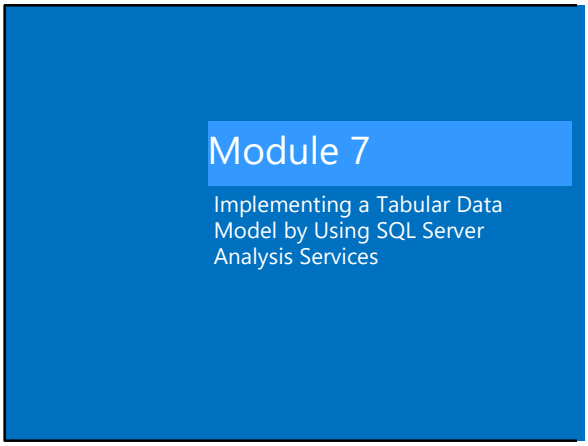
7: Implementing a Tabular Data Model by Using SQL Server Analysis Services



The **20768C-MIA-SQL** virtual machine that is used in the lab for this module includes software and services that can take a while to start. For the best experience, have students start the **20768C-MIA-DC** and **20768C-MIA-SQL** virtual machines at the beginning of the module so that the services are ready before they begin the lab.

20768C

7: Implementing a Tabular Data Model by Using SQL Server Analysis Services

Module Overview

- Introduction to Tabular Data Models in SQL Server Analysis Services
- Creating a Tabular Data Model
- Using a SQL Server Analysis Services Tabular Model in an Enterprise BI Solution

20768C

7: Implementing a Tabular Data Model by Using SQL Server Analysis Services

Lesson 1: Introduction to Tabular Data Models in SQL Server Analysis Services

- What Is a Tabular Data Model?
- Options for Creating a Tabular Data Model Project in SQL Server Analysis Services
- Using SQL Server Data Tools to Develop a Tabular Data Model in SQL Server Analysis Services
- The Workspace Database

What Is a Tabular Data Model?

- An in-memory database that uses xVelocity storage and processing engine
- Exposes data in a relational format
- Easier to learn for IT professionals who have relational database experience
- Offers a faster deployment time due to simplicity
- Uses DAX
- Can be created by using the PowerPivot add-in in Excel, or SQL Server Data Tools

Options for Creating a Tabular Data Model Project in SQL Server Analysis Services

- Install SSAS instance that runs in tabular mode
- Create a tabular data model project by using:
  - SQL Server Data Tools for BI templates:
    - Analysis Services Tabular Project
    - Import from PowerPivot
    - Import from Server (Tabular)
  - SQL Server Management Studio:
    - Restore from PowerPivot
- Key features in Analysis Services tabular databases
  - DirectQuery mode
  - Row-level and object-level security
  - Partitions

Using SQL Server Data Tools to Develop a Tabular Data Model in SQL Server Analysis Services

- Single development environment for BI developers that supports a range of features:
  - Templates to create projects
  - Integrated project life cycle management and source control
  - Debugging tools
  - Building and deployment directly to test and production servers
- Tabular Model Designer:
  - Data view and diagram view
  - Measure grid
- Get Data

Mention that the Get Data functionality automatically creates metadata structures to support models—there is no need to create data source views, dimensions, or cubes.

The Workspace Database

- Created on SSAS instance when tabular data model project is created
- Contains all data and metadata of project
- Can use local or remote SSAS instance
- Configure the workspace database:
  - **Workspace Server**
  - **Workspace Retention**

Explain the limitations of using a remote instance of SQL Server Analysis Services to host the workspace database.

Question

What are the advantages and disadvantages of tabular models when compared to multidimensional models?

Answer

Tabular models are easier to create for most users and do not require as much specialist knowledge as multidimensional cubes. They can also be easily created from PowerPivot workbooks, meaning that business users can use Excel for self-service analysis, and the models that they create can then be refined and brought under IT management as they mature.

On the other hand, multidimensional data modeling is a well-understood professional BI technique, and there are many tools and client applications that can take advantage of it. There are some capabilities of multidimensional models that are not supported, or are difficult to implement in a tabular data model. These limitations include the ability to define specific dimension types such as account dimensions, implement translations, or perform write-back analysis.

7: Implementing a Tabular Data Model by Using SQL Server Analysis Services

Lesson 2: Creating a Tabular Data Model

- Importing Tables
- Defining Measures
- Demonstration: Creating a Tabular Data Model Project
- Managing Relationships
- Configuring Columns
- Demonstration: Managing Relationships and Columns
- Creating Hierarchies
- Demonstration: Creating a Hierarchy

Importing Tables

- Create data source connections for a wide range of connection options including common third-party databases
- Filter out columns that are not required for analysis:
  - Improves performance
  - Simplifies user experience
- Provide table aliases for ease of use

Defining Measures

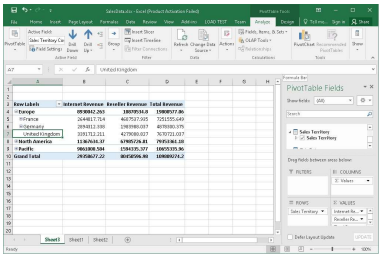
- Measures are numeric aggregations of business metrics
- They are usually based on a simple DAX function that aggregates a numeric column value

```
Revenue:=Sum([SalesAmount])
```

5: Introduction to MDX

Using DAX from Excel and SSRS

- Connecting to Analysis Services from Excel
- Designing an Query



The screen shot on the slide shows a PivotTable in Excel that displays data from an OLAP cube in SQL Server Analysis Services. Tell the students that they will use Excel to perform this task in the last exercise of the lab.

Demonstration: Creating a Tabular Data Model Project

In this demonstration, you will see how to:

- Create a tabular data model project
- Import tables into a tabular model
- Define measures in a data model
- Test the data model in Excel

Preparation Steps

Start the **20768C-MIA-DC** and **20768C-MIA-SQL** virtual machines.

Demonstration Steps

Create a Tabular Data Model Project

1. Ensure that the **20768C-MIA-DC** and **20768C-MIA-SQL** virtual machines are both running, and then log on to **20768C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.

2. In the **D:\Demofiles\Mod07** folder, run **Setup.cmd** as **Administrator**.

3. In the **User Account Control** dialog box, click **Yes**.

4. Start **SQL Server Data Tools 2015**, and then on the **File** menu, point to **New**, and then click **Project**.

5. In the **New Project** dialog box, type the following values, and then click **OK**:

   - **Project Template**: Analysis Services Tabular Project
   - **Name**: TabularDemo
   - **Location**: D:\Demofiles\Mod07

6. If the **Tabular model designer** dialog box is displayed, type the following values, and then click **OK**:

   - **Workspace server**: localhost\SQL2
   - **Compatibility level**: SQL Server 2017 / Azure Analysis Services (1400)

7. On the **Tools** menu, click **Options**.

8. In the **Options** dialog box, expand **Analysis Services Tabular Designers**, and then click **Workspace Database**. Note the name of the default workspace server that is used to host the data model during development, and then click **Cancel**.

(More notes on the next slide)

Import Tables into a Tabular Model

1. On the **Model** menu, click **Import From Data Source**.
2. In the **Get Data** window, select **SQL Server database**, and then click **Connect**.
3. On the **SQL Server database** page, type the following values, and then click **OK**:
   - **Server**: MIA-SQL
   - **Database (optional)**: AdventureWorksDW
4. On the Impersonation Information page, change the **Impersonation Mode** to **Impersonate Service Account**, then click **Connect**.
5. In the **Encryption Support** dialog box, click **OK**
6. On the **Navigator** page, select the following source tables, and then click **OK**:
   - **DimCustomer**
   - **DimDate**
   - **DimGeography**
   - **DimProduct**
   - **DimProductCategory**
   - **DimProductSubcategory**
   - **FactInternetSales**
7. In the Query Editor, in the **Queries** list, click **DimCustomer** table and change the **Name** property to **Customer**.

(More notes on the next slide)

8. On the **Columns** menu, click **Choose Columns**, clear the check box for each of the following columns, and then click **OK**:
   - Title
   - MiddleName
   - NameStyle
   - Suffix
   - SpanishEducation
   - FrenchEducation
   - SpanishOccupation
   - FrenchOccupation
9. In the **Queries** list, click **DimDate** and change the **Name** property to **Date**.
10. On the **Columns** menu, click **Choose Columns**, clear the check box for each of the following columns, and then click **OK**:
    - SpanishDayNameOfWeek
    - FrenchDayNameOfWeek
    - SpanishMonthName
    - FrenchMonthName
    - CalendarSemester
    - FiscalSemester
11. In the **Queries** list, click **DimGeography** and change the **Name** property to **Geography**.
12. In the **Queries** list, click **DimProduct** and change the **Name** property to **Product**.
13. In the **Queries** list, click **DimProductCategory** and change the **Name** property to **Product Category**.

(More notes on the next slide)

14. In the **Queries** list, click **DimProductSubCategory** and change the **Name** property to **Product Subcategory**.

15. In the **Queries** list, click **FactInternetSales** and change the **Name** property to **Internet Sales**.

16. On the toolbar, click **Import**, and then wait for the data to be imported. When the data has been imported successfully, click **Close**.

17. Click each of the tabs in the model designer to see the data that has been imported into each table.

Define Measures in a Data Model

1. Click the **Internet Sales** tab, and then click the first empty cell in the measure grid under the **SalesAmount** column.

2. On the **Column** menu, point to **AutoSum**, and then click **Sum**.

3. In the formula bar, modify the expression that has been generated to change the name of the measure to **Revenue**, as shown in the following example:

   Revenue:=SUM([SalesAmount])

4. Widen the **SalesAmount** column, and then note that the measure is calculated and displayed as a currency value. The formatting has been inherited from the column.

5. Select the cell that contains the **Revenue** measure, and then press F4 to display the properties pane. View the **Format** property, which is set to **Currency**.

6. Create a second measure named **Cost** in the measure grid under the **TotalProductCost** column based on the following expression, and format the **Cost** measure as currency:

   Cost:=Sum([TotalProductCost])

7. On the **File** menu, click **Save All**.

Test the Data Model in Excel

1. On the **Model** menu, click **Analyze in Excel**.

2. In the **Analyze in Excel** dialog box, ensure that **Current Windows User** is selected and that

(More notes on the next slide)

the **(Default)** perspective is specified, and then click **OK**.

3. When Excel opens, in the **PivotTable Fields** pane, note that the measures that you defined are displayed under an **Internet Sales** measure group. The **Internet Sales** table is also shown in this pane, and contains the columns in the table, including those on which the aggregated measures are based. Some users might find this confusing so, in a future refinement of the data model, the columns that are not useful for analysis should be hidden.

4. Select the **Revenue** measure so that it is summarized in the PivotTable.

5. In the **Date** table, select **CalendarYear** so that the revenue in the PivotTable is aggregated by year. Note that there is no indication in the **Date** table that tells the user if the values that are displayed represent the order date, the due date, or the ship date that is associated with the order.

6. In the **Date** table, select **EnglishMonthName** so that the revenue is further broken down into monthly totals. Note that the months are shown in alphabetical order instead of chronological order.

7. Close Excel without saving the workbook.

8. Keep SQL Server Data Tools open for the next demonstration.

(More notes on the next slide)

Managing Relationships

- Automatically recognize relationships based on foreign keys
- Manually create relationships when they are not explicitly defined
- Only one relationship between two tables can be active
- To define role-playing dimensions, import the same table multiple times

Configuring Columns

- Specify column data types
- Create calculated columns based on DAX expressions
- Specify sort orders for columns
- Hide and rename columns

Demonstration: Managing Relationships and Columns

In this demonstration, you will see how to:

- Manage relationships
- Rename and hide columns
- Configure column sort order
- Review your changes in Excel

Preparation Steps

Complete the previous demonstration in this module.

Demonstration Steps

Manage Relationships

1. Ensure that you have completed the previous demonstration in this module.
2. In SQL Server Data Tools, with the **Model.bim** pane visible, on the **Model** menu, point to **Model View**, and then click **Diagram View**. The tables are shown as a schema diagram, with lines between them to denote relationships. Note that there are three relationships between the **Internet Sales** table and the **Date** table.
3. Double-click the solid line between the **Internet Sales** and **Date** tables. Note the columns that are used to define this relationship, and that this is an active relationship, and then click **Cancel**.
4. Double-click each of the dotted lines between the **Internet Sales** and **Date** tables, and then click **Cancel**. In each case, note the columns that are used to define these relationships and that these are inactive relationships, and then click **Cancel**.
5. Right-click each dotted relationship line in turn, and then click **Delete**. When you are prompted to permanently delete the relationship from the model, click **Delete from Model**.
6. Double-click the remaining solid line between the **Internet Sales** and **Date** tables.
7. In the **Edit Relationship** dialog box, in the **Columns** list for the **Internet Sales** table, select **OrderDateKey** and then click **OK** (if it is already selected, click **Cancel**), This ensures that the active relationship is based on the order date, not the shipping date or due date.
8. Right-click the **Date** table title bar, click **Rename**, and then rename the table to **Order Date**.
9. In the Tabular Model Explorer, expand **Data Sources**, right-click **SQL\mia-sql:AdventureWorksDW**, and then click **Import New Tables**.
10. On the **Navigator** page, select the **DimDate** table, and then click **OK**.

(More notes on the next slide)

11. In the Query Editor, in the **Queries** list, click **DimDate** table and change the **Name** to **Ship Date**.

12. On the **Columns** menu, click **Choose Columns**, clear the following columns, and then click **OK**:

    - SpanishDayNameOfWeek
    - FrenchDayNameOfWeek
    - SpanishMonthName
    - FrenchMonthName
    - CalendarSemester
    - FiscalSemester

13. Click **Import**, and then wait for the table to be imported. When it has been imported successfully, click **Close**.

14. Arrange the diagram so that you can see the **Ship Date** and **Internet Sales** tables, and then drag the **ShipDateKey** column from the **Internet Sales** table to the **DateKey** column in the **Ship Date** table to create the relationship. If the **Ship Date** table does not appear, refresh the page.

Rename and Hide Columns

1. Click the title bar of the **Order Date** table, and then click its **Maximize** icon.

2. In the maximized **Order Date** table, click the **DateKey** column, hold down CTRL, click the **DayNumberOfWeek** and **MonthNumberOfYear** columns to select them, right-click any of the selected columns, and then click **Hide from Client Tools**.

3. In the maximized **Order Date** table, right-click the **FullDateAlternateKey** column, click **Rename**, and then rename the column to **Date**.

4. Repeat the previous step to rename the following columns:

    - EnglishDayNameOfWeek (rename to **Weekday**)

(More notes on the next slide)

- EnglishMonthName (rename to **Month**)

1. Click the **Restore** icon for the **Order Date** table.

6. Maximize the **Internet Sales** table, hide all columns other than the **Revenue** and **Cost** measures that you created in the previous exercise, and then restore the **Internet Sales** table.

Configure Column Sort Order

1. On the **Model** menu, point to **Model View**, and then click **Data View**.
2. On the **Order Date** tab, click the **Weekday** column heading.
3. On the **Column** menu, point to **Sort**, and then click **Sort by Column**.
4. In the **Sort by Column** dialog box, in the **Sort** column, ensure that **Weekday** is selected, in the **By** column, select **DayNumberOfWeek**, and then click **OK**.
5. Click the **Month** column heading.
6. On the **Column** menu, point to **Sort**, and then click **Sort by Column**.
7. In the **Sort by Column** dialog box, in the **Sort** column, ensure that **Month** is selected, in the **By** column, select **MonthNumberOfYear**, and then click **OK**.
8. On the **File** menu, click **Save All**.

Review Your Changes in Excel

1. On the **Model** menu, click **Analyze in Excel**.
2. In the **Analyze in Excel** dialog box, ensure that **Current Windows User** is selected and that the **(Default)** perspective is specified, and then click **OK**.
3. When Excel opens, in the **PivotTable Fields** pane, note that the measures that you defined are still displayed under an **Internet Sales** measure group, but that the **Internet Sales** table is no longer shown here because all other columns have been hidden.
4. Select the **Revenue** measure so that it is summarized in the PivotTable.
5. In the **Order Date** table, select **CalendarYear** so that the revenue in the PivotTable is aggregated by year based on the order date.
6. In the **Order Date** table, select **Month** so that the revenue is further broken down into monthly totals.

(More notes on the next slide)

Note that the months are now shown in chronological order.

7. In the **Order Date** table, select **CalendarQuarter**, and then note that the quarter is shown under each month.

8. In the **PivotTable Fields** pane, in the **Rows** area, drag **CalendarQuarter** so that it is above **Month**, and the PivotTable shows the correct hierarchy for year, quarter, and month. This user experience could be improved by enabling users to select a hierarchy that automatically enables them to drill down to the correct levels.

9. Close Excel without saving the workbook.

10. Keep SQL Server Data Tools open for the next demonstration.

Creating Hierarchies

- Hierarchies enable drill-up and drill-down aggregations
- Common examples:
  - Product Category, Product Subcategory, and Product Name
  - Year, Quarter, Month, and Date
- Create hierarchies in the diagram view
- Ragged hierarchies

Demonstration: Creating a Hierarchy

In this demonstration, you will see how to:
- Create a hierarchy
- Browse the hierarchy in Excel

Preparation Steps

Complete the previous demonstrations in this module.

Demonstration Steps

Create a Hierarchy

1. Ensure that you have completed the previous demonstrations in this module.
2. In SQL Server Data Tools, with the **Model.bim** pane visible, on the **Model** menu, point to **Model View**, and then click **Diagram View**.
3. Maximize the **Order Date** table, and then on its title bar, click the **Create Hierarchy** icon. Name the new hierarchy **Calendar Date**.
4. Drag the **CalendarYear** column to the **Calendar Date** hierarchy.
5. Drag the **CalendarQuarter** column to the **Calendar Date** hierarchy.
6. Drag the **Month** column to the **Calendar Date** hierarchy.
7. Drag the **Date** column to the **Calendar Date** hierarchy.
8. Restore the **Order Date** table, and then on the **File** menu, click **Save All**.

Browse the Hierarchy in Excel

1. On the **Model** menu, click **Analyze in Excel**.
2. In the **Analyze in Excel** dialog box, ensure that **Current Windows User** is selected and that the **(Default)** perspective is specified, and then click **OK**.
3. When Excel opens, select the **Revenue** measure so that it is summarized in the PivotTable.
4. In the **Order Date** table, select **Calendar Date** so that the revenue in the PivotTable is aggregated by year.
5. In the PivotTable, expand the years to show the revenue by quarter, expand the quarters to view revenue by month, and then expand the months to view revenue for individual dates.
6. Close Excel without saving the workbook.
7. Keep SQL Server Data Tools open for the next demonstration.

Question

You are implementing an order date and a ship date role-playing dimension based on a date dimension table. You can achieve this by importing the fact table and the date dimension table once, and then creating a relationship between the order date key in the fact table and the date key in the date dimension table, and also creating a relationship between the ship date key in the fact table and the date key in the date dimension table.

( )False

( )True

Answer

(√)False

( )True

Lesson 3: Using a SQL Server Analysis Services Tabular Model in an Enterprise BI Solution

- Perspectives
- Partitions
- Analysis Services Processing Types
- DirectQuery Mode
- Security for Tabular Data Models
- Deploying a Tabular Data Model
- Demonstration: Deploying a Tabular Data Model

Perspectives

- Provide a simplified view of complex data models
- Select tables, measures, and columns to include in perspective
- Not a method of implementing security

Partitions

- Divide tables into logical partitions
- Separate frequently changing data from static data
- Create partitions in workspace and deployed databases

Analysis Services Processing Types

- Process Types:
  - Process Default
  - Process Full
  - Process Data
  - Process Clear
  - Process Add
- Processing databases, tables, and partitions

DirectQuery Mode

- Bypass xVelocity in-memory storage
- Retrieve data directly from a relational database
- Take advantage of powerful server hardware
- Appropriate for very large data sets
- Some feature limitations to consider

Describe the feature limitations to consider as shown in the manual. Also, point out the forward link that shows feature limitations in more detail.

Security for Tabular Data Models

- Create roles to group users according to security requirements
- Apply database-level permissions to roles:
  - Read
  - Read and Process
  - Process
  - Administrator
  - None
- Use DAX to create filters that define row-level security
- Use JSON-based metadata, TMSL, or TOM to configure table-level and column-level security

Deploying a Tabular Data Model

- Copies data and metadata in workspace database to new database on SQL Server Analysis Services instance
- Enables connections from client tools
- Deployment options
- Deployment server
- BI semantic model connection

Explain deployment options and deployment server settings as shown in the manual.

Demonstration: Deploying a Tabular Data Model

In this demonstration, you will see how to:

- Deploy a tabular data model project
- Use a tabular SQL Server Analysis Services database in Excel

Preparation Steps

Complete the previous demonstrations in this module.

Demonstration Steps

Deploy a Tabular Data Model Project

1. Ensure that you have completed the previous demonstrations in this module.
2. In SQL Server Data Tools, in Solution Explorer, right-click the **TabularDemo** project, and then click **Properties**.
3. In the **TabularDemo Property Pages** dialog box, type the following values, and then click **OK**:
   - **Server**: localhost\SQL2
   - **Database**: DemoDB
   - **Model Name**: Internet Sales
4. On the **Build** menu, click **Deploy TabularDemo**. If you are prompted for impersonation credentials, type the user name **ADVENTUREWORKS\ServiceAcct** and the password **Pa55w.rd**, and then click **OK**.
5. In the **Deploy** dialog box, when deployment has completed, click **Close**.
6. Close SQL Server Data Tools.

Use a Tabular SQL Server Analysis Services Database in Excel

1. Start Excel, and then create a new blank workbook.
2. On the **Data** tab, in **Get External Data** area, in the **From Other Sources** drop-down list, select **From Analysis Services**.
3. In the **Data Connection Wizard** dialog box, in the **Server name** box, type **MIA-SQL\SQL2**, ensure that **Use Windows Authentication** is selected, and then click **Next**.
4. On the **Select Database and Table** page, ensure that the **DemoDB** database and the **Internet Sales** cube are selected, and then click **Next**.

(More notes on the next slide)

5. On the **Save Data Connection File and Finish** page, click **Finish**.

6. In the **Import Data** dialog box, ensure that the **Existing worksheet** option is selected, and then click **OK**.

7. In the **PivotTable Fields** pane, under **Internet Sales**, select **Revenue**.

8. In the **PivotTable Fields** pane, under **Order Date**, select the **Calendar Date** hierarchy.

9. In the **PivotTable Fields** pane, under **Geography**, drag **EnglishCountryRegionName** to the **Columns** area.

10. Explore the data in the PivotTable. When you have finished, close Excel without saving the workbook.

Question

Which of the following statements regarding DirectQuery mode is correct?

( )Option 1: A tabular data model that uses DirectQuery mode will always have a better performance than a tabular data model that uses in-memory storage.

( )Option 2: A tabular data model that uses in-memory storage will always have a better performance than a tabular data model that uses DirectQuery mode.

( )Option 3: A tabular data model that uses DirectQuery mode will always contain up-to-date information.

( )Option 4: A tabular data model that uses DirectQuery mode will only support a SQL Server relational database data source.

( )Option 5: You cannot use MDX in a tabular data model that uses DirectQuery mode.

Answer

(√) Option 3: A tabular data model that uses DirectQuery mode will always contain up-to-date information.

7: Implementing a Tabular Data Model by Using SQL Server Analysis Services

Lab: Implementing a Tabular Data Model in SQL Server Analysis Services

- Exercise 1: Creating a Tabular Data Model Project in SQL Server Analysis Services
- Exercise 2: Configuring Columns and Relationships
- Exercise 3: Deploying a Tabular Data Model Project in SQL Server Analysis Services

Logon Information
Virtual machine: **20768C-MIA-SQL**
User name: **ADVENTUREWORKS\Student**
Password: **Pa55w.rd**

Estimated Time: 60 minutes

Point out that the lab instructions are deliberately designed to be high level so that students need to think carefully about what they are trying to accomplish and work out how best to proceed. Encourage students to read the scenario information carefully and collaborate with each other to meet the scenario requirements. Remind students that, if they find a particular task or exercise too challenging, they can find step-by-step instructions in the lab answer key.

Point out that, like all other labs in this course, students must start by running a setup script to prepare the lab environment.

Exercise 1: Creating a Tabular Data Model Project in SQL Server Analysis Services

Adventure Works Cycles has a data warehouse that contains fact and dimension tables for sales. You plan to use the tables that relate to reseller sales as the basis of your data model.

Exercise 2: Configuring Columns and Relationships

You have created an initial tabular data model, but have identified some problems. The column names are not user-friendly, some of the relationships are ambiguous, and some date attributes are being sorted alphabetically rather than chronologically. You are now going to address these issues.

Exercise 3: Deploying a Tabular Data Model Project in SQL Server Analysis Services

You have configured the columns, relationships, and hierarchies of your tabular data model. Now you must improve the processing performance by separating the more recent data and the older data into partitions. You must then set up a perspective for an analysis team who are interested in breaking down sales only by product dimensions and not any other dimensions. Next, you must set up a security role for users who must only have access to data that relates to bike sales in the United States. Finally, you must deploy the tabular model to the server so that users can connect to it by using client tools.

7: Implementing a Tabular Data Model by Using SQL Server Analysis Services

Lab Scenario

Adventure Works needs an enterprise data model for reseller sales analysis, and you have decided to implement it by using a tabular model.

Lab Review

- In this lab, you have created a tabular data model in SQL Server Analysis Services. You have configured relationships, created hierarchies, partitions, and perspectives, and then deployed and tested the model.

Module Review and Takeaways

- In this module, you have learned how to develop and deploy a tabular data model for SQL Server Analysis Services by using SQL Server Data Tools.