than one way to complete a particular task, and to challenge you by using skills you learned in previous tasks.

The purpose of the lessons is to guide you through authoring a basic tabular model running in Tabular mode by using many of the features included in SQL Server Data Tools. Because each lesson builds upon the previous lesson, you should complete the lessons in order. Once you have completed all of the lessons, you will have authored and deployed the Adventure Works Internet Sales sample tabular model on an Analysis Services server.

After you complete the tutorial, you can add to your model, or create additional models using the same AdventureWorksDW2012 sample database. The database includes an extensive collection of tables and data that can apply to a wide range of sample models.

Note
This tutorial does not provide lessons or information about managing a deployed tabular model database by using SQL Server Management Studio, or using a reporting client application to connect to a deployed model to browse model data.

Prerequisites

In order to complete this tutorial, you must have the following prerequisites installed:

- SQL Server 2012 Analysis Services (running in Tabular mode).
- SQL Server Data Tools (SSDT) - installed as part of SQL Server 2012.
- AdventureWorksDW2012 sample database. This sample database includes the data necessary to complete this tutorial. To download the sample database, see http://go.microsoft.com/fwlink/?LinkID=220093.
- Microsoft Excel 2003 or later (for use with the Analyze in Excel feature in lesson 11)

Lessons

This tutorial includes the following lessons:

Lesson	Estimated time to complete
Lesson 1: Create a New Tabular Model Project	10 minutes
Lesson 2: Add Data	20 minutes
Lesson 3: Rename Columns	20 minutes
Lesson 4: Mark as Date Table	3 minutes
Lesson 5: Create Relationships	10 minutes

Lesson	Estimated time to complete
Lesson 6: Create Calculated Columns	15 minutes
Lesson 7: Create Measures	30 minutes
Lesson 8: Create Key Performance Indicators	15 minutes
Lesson 9: Create Perspectives	5 minutes
Lesson 10: Create Hierarchies	20 minutes
Lesson 11: Create Partitions	15 minutes
Lesson 12: Create Roles	15 minutes
Lesson 13: Analyze in Excel	20 minutes
Lesson 14: Deploy	5 minutes

Supplemental Lessons

This tutorial also includes Supplemental Lessons. Topics in this section are not required to complete the tutorial, but can be helpful in better understanding advanced tabular model authoring features.

This tutorial includes the following supplemental lessons:

Lesson	Estimated time to complete
Implement Dynamic Security by Using Row Filters	30 minutes

Next Step

To begin the tutorial, continue to the first lesson: Lesson 1: Create a New Tabular Model Project.

Lesson 1: Create a New Tabular Model Project

In this lesson, you will create a new, blank tabular model project in SQL Server Data Tools (SSDT). Once your new project is created, you can begin adding data by using the Table Import Wizard. In addition to creating a new project, this lesson also includes a brief introduction to the tabular model authoring environment in SQL Server Data Tools.

To learn more about the different types of tabular model projects, see Tabular Model Projects (SSAS). To learn more about the tabular model authoring environment, see Tabular Model Designer (SSAS).

Estimated time to complete this lesson: **10 minutes**

Prerequisites

This topic is the first lesson in a tabular model authoring tutorial. To complete this lesson, you must have the AdventureWorksDW2012 database installed on a SQL Server instance. For more information, see Tabular Modeling (Adventure Works Tutorial).

Create a New Tabular Model Project

To create a new tabular model project

1. In SQL Server Data Tools, on the **File** menu, click **New**, and then click **Project**.
2. In the **New Project** dialog box, under **Installed Templates**, click **Business Intelligence**, then click **Analysis Services**, and then click **Analysis Services Tabular Project**.
3. In **Name**, type **AW Internet Sales Tabular Model**, then specify a location for the project files.

   By default, **Solution Name** will be the same as the project name, however, you can type a different solution name.
4. Click **OK**.

Understanding the SQL Server Data Tools Tabular Model Authoring Environment

Now that you've created a new tabular model project, let's take a moment to explore the tabular model authoring environment in SQL Server Data Tools (Visual Studio 2010).

After your project is created, it opens in SQL Server Data Tools. An empty model will appear in the model designer and the **Model.bim** file will be selected in the **Solution Explorer** window. When you add data, tables and columns will appear in the designer. If you don't see the designer (the empty window with the Model.bim tab), in **Solution Explorer**, under **AW Internet Sales Tabular Model**, double click the **Model.bim** file.

You can view the basic project properties in the **Properties** window. In **Solution Explorer**, click **AW Internet Sales Tabular Model**. Notice in the **Properties** window, in **Project File**, you will see **AW Internet Sales Tabular Model.smproj**. This is the project file name, and in **Project Folder**, you will see the project file location.

In **Solution Explorer**, right-click the **AW Internet Sales Tabular Model** project, and then click **Properties**. The **AW Internet Sales Tabular Model Property Pages** dialog box appears. These are the advanced project properties. You will later set some of these properties when you are ready to deploy your model.

Now, let's look at the model properties. In **Solution Explorer**, click **Model.bim**. In the **Properties** window, you will now see the model properties, most important of which is the **DirectQuery Mode** property. This property specifies whether or not the model is deployed in In-Memory mode (Off) or DirectQuery mode (On). For this tutorial, you will author and deploy your model in In-Memory mode.

When you create a new model, certain model properties are set automatically according to the Data Modeling settings that can be specified in the Tools\Options dialog box. Data Backup, Workspace Retention, and Workspace Server properties specify how and where the workspace database (your model authoring database) is backed up, retained in-memory, and built. You can change these settings later if necessary, but for now, just leave these properties as they are.

When you installed SQL Server Data Tools, several new menu items were added to the Visual Studio 2010 environment. Let's look at the new menu items that are specific to authoring tabular models. Click on the **Model** menu. From here, you can launch the Table Import Wizard, view and edit existing connections, refresh workspace data, browse your model in Microsoft Excel with the Analyze in Excel feature, create perspectives and roles, select the model view, and set calculation options.

Click on the **Table** menu. Here, you can create and manage relationships between tables, create and manage, specify date table settings, create partitions, and edit table properties.

Click on the **Column** menu. Here, you can add and delete columns in a table, freeze columns, and specify sort order. You can also use the AutoSum feature to create a standard aggregation measure for a selected column. Other toolbar buttons provide quick access to frequently used features and commands.

Explore some of the dialogs and locations for various features specific to authoring tabular models. While some items will not yet be active, you can get a good idea of the tabular model authoring environment.

Next Steps

To continue this tutorial, go to the next lesson: Lesson 2: Add Data.

Lesson 2: Add Data

In this lesson, you will use the Table Import Wizard in SQL Server Data Tools to connect to the AdventureWorksDW2012 SQL database, select data, preview, and filter the data, and then import the data into your model workspace.

By using the Table Import Wizard, you can import data from a variety of relational sources: Access, SQL, Oracle, Sybase, Informix, DB2, Teradata, and more. The steps for importing data from each of these relational sources are very similar to what is described below. Additionally, data can be selected using a stored procedure.