

SQL Server Analysis Services Assignment

Group Submissions: This assignment is to be a joint submission from your BI course project team. Only one submission per team.

Assignment Overview

In this assignment, you will build and deploy a fully functional MDB using SSAS. Unlike the SSIS assignment, all problems in this assignment should be familiar as they require you to perform several of the same tasks and activities performed in both SSAS labs. Because of that, there is very little “recipe” about this assignment. Each problem describes an outcome or goal which you are to implement using the knowledge gained from the labs.

All of the data required by this assignment has been provide in the assignment ZIP file. Extract the ZIP file to a folder and rename the *GroupX* folder from the restored ZIP file **replacing the X with the number of your group – do this first!**

To complete this assignment, the system you use must have SQL Server with the **SSAS Assignment_DW** database attached along with the SSDT development environment. To attach the required database start **SQL Server Management Studio** and attach the **SSAS Assignment_DW** database located in the *GroupX\DataSources* folder restored from the ZIP file. There is also a database schema diagram located in the same folder.

Deliverables

Once you have completed the assignment, save all files and exit SSDT. Detach the **SSAS_Assignment_DW** database from SQL Server.

A few of the problem steps require some discussion write-up. Any required written responses are to be included in you submission ZIP file in a single Word document. Your document must be located in the *GroupX* folder (now renamed with your group number). Example: *GroupX\OurResponses.docx*.

Create a ZIP file of the *GroupX* folder and upload that single ZIP file to the Blackboard **SSAS Assignment** item.

Problem 1: Startup Activities (6 pts)

1. Restore the ZIP file and rename the **GroupX** folder.
2. Using SSMS, attach the **SSAS_Assignment_DW** database located in the *DataSources* folder.
3. Begin your assignment solution by using the *SSAS Assignment.sln* SSDT solution file provided. Yes, the solution *is* empty.

Problem 2: Attaching Data (4 pts)

1. Add a new **Data Source** for the **SSAS_Assignment_DW** database. Your data source must be named **SSAS Assignment DW**.

2. Add a new **Data Source View (DSV)** using the **SSAS Assignment DW** data source. Your data source view must be named **SSAS Assignment DW**. Select all of the tables except for **sysdiagrams (dbo)** from the list of tables that are available from the selected data source.

Problem 3: Refining the Data Source View (10 pts)

1. Add a named calculation called **TotalSales** to the FactSales table that represents the product of order quantity and unit price. Recall that named calculations use SQL functions and syntax.
2. Add a named calculation called **DateLabel** to the **DimDate** table that will show the date in the following format – *July 1, 2009*.
3. Add a named calculation called **FullName** to the **DimCustomer** table that will show the customer name in the following format. Ex: *Joe Blow* when there is no middle name and *Joe B. Blow* when there is a middle name. Note that this only use the first initial of the middle name, not the entire name.
4. Change the “friendly name” of each table by removing the "**Dim**" and "**Fact**" prefixes from these tables.

Problem 4: Defining Dimensions (10 pts)

1. Create new SSAS dimension named **DimDate** using the Date table as the source. Retain all attributes from the **Date** table for the new dimension.
2. For each of the three remaining dimension tables in the DSV, create a new SSAS dimension. Retain all attributes from each table for the new dimension. Save the new dimensions as **DimProduct**, **DimCustomer** and **DimSalesPerson** respectively. Note there no spaces in the names.

Problem 5: Defining a Cube (10 pts)

1. Create a single cube named **SSAS Assignment.cube**. Include all dimensions and all measures except **Fact Sales Count** in your cube.
2. Format the **Unit Price** and **Total Sales** measures as currency.
3. Format the **Order Quantity** measure as a number with no decimal places and a thousands separator.

Problem 6: Modifying DimDate (10 pts)

1. Delete the **Date Label** attribute from the Attribute pane.
2. Use the **DateLabel** named calculation created previously as the **NameColumn** property of the **Date Key** attribute. Rename the **Date Key** attribute to **Date**.
3. Save and deploy your project.
4. Once successfully deployed, browse **DimDate** using the designer’s browser tab. Select the **Month Name** attribute in the Hierarchy: pull down menu. Notice that the month names are sorted alphabetically and not temporally. This is very disconcerting to users. Fix this sorting problem so that month names sort temporally.
5. Add time-based dimension business intelligence to the **DimDate** attributes – **FullDateAlternateKey**, **MonthName**, **CalendarQuarter**, and **CalendarYear**.

Problem 7: Modifying DimProduct (5 pts)

1. Use SSAS's built-in automatic grouping feature (i.e., discretization) to create 3 "buckets" for **List Price** attribute.

Problem 8: Modifying DimCustomer (25 pts)

1. Delete the **Full Name** attribute from **DimCustomer** attribute list.
2. Instead of seeing a number for the **CustomerID** attribute users would prefer to see a customer's full name (i.e., the **FullName** named calculation created previously). Also, while customers must be uniquely identified via the **CustomerID** attribute, the **CustomerID** attribute name should be changed to reflect the data being displayed. I.e. it should be changed to **Customer Name**.
3. The **State ID** and **Customer ID OLTP** attributes have too much data to be maintained as hierarchies. Make sure they are not.
4. Add each of the following attributes to an attribute hierarchy display folder called **Demographic – Commute Distance, Education, Gender, House Owner Flag, Marital Status, Number Cars Owned, Number Children At Home, Occupation, Total Children, Yearly Income**.
5. Create a user-defined hierarchy for customers named **Customer Geography**. The hierarchy should be composed of the following attributes – **Country Code > State Name > City > Customer Name**.
6. Set all of the individual attributes involved in the hierarchy to be "invisible" to all client applications. Now, save and deploy your project. Resolve any errors that may have caused the deployment to fail.
7. On the Attribute Relationships tab, create the relationships needed to fully support the **Customer Geography** hierarchy. Now, deploy your project. This time, the deploy will fail, the question is why? Explore the Error List window (*View > Error List*) to see why the deployment failed. Obviously, it has something to do with the last step. To dig deeper, run the following query in SQL Server Management Studio and look for the offending city name that you found in the *Output*.

```
SELECT [City],[StateName]
FROM [SSAS_Assignment_DW].[dbo].[DimCustomer]
GROUP BY City, StateName
ORDER BY 1
```

- a) Describe (in a Word document that you will upload) exactly what you see as the problem.
- b) Fix the problem. Hint: In SSAS Lab2 we performed a very similar set of activities for the customer dimension. This time however, some important steps pertaining to some properties of the attributes involved in the hierarchy and the accompanying relationships were omitted. To solve this problem you must perform those steps. Once completed (correctly), your cube will deploy.

Problem 9: Modifying DimSalesPerson (10 pts)

1. Attributes used in the hierarchy should not be visible to cube clients as individual dimension attributes.
2. When used by client applications, the **Sales Person ID** attribute name should display the **FullName**. As a result, the name of this attribute should be changed to **Sales Person**.
3. Management would like to have a hierarchy to view sales by store id and then sales person. This hierarchy is to be named **Sales Person Hierarchy**. Create this hierarchy.

Problem 10: Modifying the Cube (10 pts)

1. Management would like to see two time-based BI capabilities incorporated into the cube – *Year Over Year Growth* and *Year Over Year Growth %* for the Total Sales measure. Using the **Calendar Year** attribute hierarchy, add the time-based BI capabilities to your cube to make this happen.
2. What was the *Year Over Year Growth* and *Year Over Year Growth %* for the **Total Sales** measure for 2012? Record your answer in the Word document created for Problem 8.