Using the Analysis Services Cube

The purpose of this assignment is to help you practice the following skills that are essential to your success in this course and in your professional life beyond school as a business analyst.

- Apply knowledge that you have previously acquired in class to query a relational database for aggregate information.
- Use the SQL Server Analysis cube that was built in Assignment 9 and acquire knowledge and apply it while creating queries using Data Analysis Expressions (DAX).
- Use the SQL Server Analysis cube that was built in Assignment 9 and acquire knowledge and apply it while creating queries using Multidimensional Expressions (MDX).
- Gain an understanding of the high-level differences between the two

While another purpose of most assignments is for you to possibly struggle and feel confused while you apply what you have learned, this assignment should not be like that as it is primarily a tutorial. IF your cube was created correctly and if you type the provided queries correctly, they should work. There are checkpoints with each set of queries. If your answers don't match when they should something is wrong. If the query is correct, there is probably a problem with your cube. If this happens, let me know and I will help you try to figure out the problem. If you have other questions, please ask as well. I am here to help you learn.

This assignment will use the Analysis Services cube which was built in Assignment 9. You must have successfully completed Assignment 9 before you can successfully complete this assignment. You will also use AdventureWorks2017 in the Database Engine and AdventureWorksDW2017 in the Database Engine.

We are using Microsoft SQL Server Analysis Services (SSAS) in **Tabular** mode. That decision was made when we installed SSAS. The other choice is **Multidimensional** mode. Tabular mode is newer and has some advantages over Multidimensional mode. Although there are currently more multidimensional models in existence and there are a few reasons to pick Multidimensional over Tabular. I'm not going to get into specifics of pros and cons right now. Most sources seem to think that Tabular is simpler to learn than Multidimensional and that is the primary reason we installed Tabular.

Tabular mode in SSAS uses DAX (Data Analysis Expressions) for creating formulas for measures and KPI when you are building a model. Tabular mode also uses DAX for queries in SSMS. Tabular mode can also use MDX (Multidimensional Expressions.) Multidimensional mode can ONLY use MDX therefore it is the more common Query Language for Data Warehouses.

DAX is also used in Microsoft PowerPivot and Power BI Desktop.

While DAX is a Microsoft proprietary language, MDX is a standards-based query language used to retrieve data from SQL Server Analysis Services (SSAS) databases. MDX is part of the OLE DB for OLAP specification sponsored by Microsoft. Many other OLAP providers support MDX, including SAS's Enterprise BI Server.

When one refers to either DAX or MDX they might be referring either to the DAX/MDX query language or to DAX/MDX expressions. Even though the query language of either has similar syntax as that of SQL, they are also significantly different.

This assignment will have you create SQL queries using the **relational** database **AdventureWorks** and the relational database **AdventureWorksDW**. Then you will create a DAX query using the tabular data warehouse model **Adventure Works Internet Sales**. Then you will create an MDX query using the tabular data warehouse model **Adventure Works Internet Sales.** All queries will be created in SSMS.

Using SSMS, connect to the **Database Engine** and open a new query window for **AdventureWorks2017**.

1. (10 points) Create and execute a query that will report the year of the order and the total of online sales (**InternetSales)** (Use LineTotal) for 2012 and 2013. Use the tables Sales.SalesOrderHeader and Sales.SalesOrderDetail. Sort by the year.

Paste your entire query into Word.
Use the Snipping Tool to capture the Results area of your query. Paste it into your Word document as Answer #1.

2. (15 points) Create and execute a SQL query that will report the name of the country, the category of the product, and the total of InternetSales (Use LineTotal) for 2012 and 2013. The country is based on the shipping address. The total should be for each country/category combination. Sort the results by the year, country and the category. Your result will have 24 rows. Use the following tables in this order:

   Sales.SalesOrderHeader
   Person.Address
   Person.StateProvince
   Person.CountryRegion
   Sales.SalesOrderDetail
   Production.Product
   Production.ProductSubcategory
   Production.ProductCategory

   Paste your entire query into Word.
   Use the Snipping Tool to capture the Results area of your query. Paste it into your Word document as Answer #2.

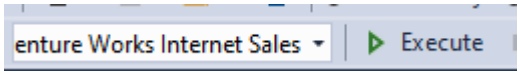
3. (10 points) In the **Database Engine** and using **AdventureWorksDW2017**, create and execute a query that will report the year of the order and the total of InternetSales (Use SalesAmount) for 2012 and 2013. Use the table FactInternetSales. Sort by the year. These totals will NOT match Answer 1.

   Paste your entire query into Word.
   Use the Snipping Tool to capture the Results area of your query. Paste it into your Word document as Answer #3.

4. (15 points) In the **Database Engine** and using **AdventureWorksDW2017**, create and execute a query that will report the name of the country, the category of the product, and the total of InternetSales (Use SalesAmount) for 2012 and 2013. The total should be for each country/category combination. Sort the results by the year, country and the category. Your result will have 36 rows. The totals will **NOT** match Answer 2. Use the following tables in this order:

   FactInternetSales
   DimProduct
   DimProductSubcategory
   DimProductCategory
   DimCustomer
   DimGeography

   Paste your entire query into Word.
   Use the Snipping Tool to capture the Results area of your query. Paste it into your Word document as Answer #4.

5. Using SSMS, connect to **Analysis Services.** Make sure that the Database in the Available Databases box 

is Adventure Works Internet Sales (It will be different if you named yours something different in Assignment 9. Next to the New Query button, you will notice new icons. 

Click on the last icon which is for a DAX query. The blank query window will look the same as a SQL query but notice the temporary name for this window is MSDAXQueryX.msdax instead of SQLQueryX.sql.

6. (10 points) In the query, type the following and execute. These numbers should match Answer 3. (Make sure that you are in Analysis Services, in the correct data warehouse and running a DAX query.)

```
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    TREATAS({2012, 2013}, DimDate[CalendarYear]),
    "TotalSales", [InternetTotalSales]
)
ORDER BY [CalendarYear]
```

Use the Snipping Tool to capture the Results area of your query. Paste it into your Word document as Answer #5.

7. (10 points) In a DAX query, type the following and execute. These numbers should match Answer 4. Notice that there are no row numbers.

```
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    DimGeography[EnglishCountryRegionName],
    DimProductCategory[EnglishProductCategoryName],
    TREATAS({2012, 2013}, DimDate[CalendarYear]),
    "TotalSales", [InternetTotalSales]
)
ORDER BY [CalendarYear], [EnglishCountryRegionName],
[EnglishProductCategoryName]
```

Use the Snipping Tool to capture the Results area of your query. Paste it into your Word document as Answer #6.

8. (10 points) In a DAX query, type the following and execute. This will be the same as Answer 6 with the addition of a column named **CombinedYearsTotalSales**.

```
EVALUATE
SUMMARIZECOLUMNS
(
    DimDate[CalendarYear],
    DimGeography[EnglishCountryRegionName],
    DimProductCategory[EnglishProductCategoryName],
    TREATAS({2012, 2013}, DimDate[CalendarYear]),
    "TotalSales", [InternetTotalSales],
    "CombinedYearsTotalSales", CALCULATE([InternetTotalSales],
ALLSELECTED(DimDate[CalendarYear]))
)
ORDER BY [CalendarYear], [EnglishCountryRegionName],
[EnglishProductCategoryName]
```

Look at the row for 2012, Germany and Accessories. The CombinedYearsTotalSales is 59748.56. Exactly what makes up this number? Answer in your document and label it Answer #7.

9. Using SSMS, connect to **Analysis Services** and open a new MDX Query window for your Analysis database. Notice that there is a new pane.

   Both Relational Databases and Multidimensional Data Warehouses use Set Theory from mathematics; however, it seems to be a little more obvious in MDX. First let's look at some basic definitions.

   A well-defined collection of unordered objects is defined as a **Set**.

   A **Tuple** is a finite ordered list of elements. A row from a table in a relational database is sometimes referred to as a tuple.

   A multidimensional database is typically referred to as a **cube**. The cube is the foundation of a multidimensional database, and each cube typically contains more than two dimensions.

   A set can contain zero, one, or more tuples. A set with zero tuples is referred to as an **empty set**.

   The syntax for an MDX query is as follows:

```
[ WITH <SELECT WITH clause> [ , <SELECT WITH clause> ... ] ]
SELECT [ * | ( <SELECT query axis clause>
    [ , <SELECT query axis clause> ... ] ) ]
FROM <SELECT subcube clause>
[ <SELECT slicer axis clause> ]
[ <SELECT cell property list clause> ]
```

The keywords WITH, SELECT, FROM, and WHERE along with the expressions following them are referred to as a clauses. In the preceding MDX query template, anything specified within square brackets means it is optional; that is, that section of the query is not mandatory in an MDX query. We can see that the WITH and WHERE clauses are optional because they are enclosed within square brackets.

The WITH clause is typically used for custom calculations and operations.

The SELECT Statement and Axis Specification

The MDX SELECT statement is used to retrieve a subset of the multidimensional data in an OLAP cube. In SQL, the SELECT statement allows us to specify which columns will be included in the row data we retrieve, which is viewed as two-dimensional data. If you consider a two-dimensional coordinate system, you have the X and Y axes. The Y axis is used for the COLUMNS and the X axis is used for ROWS. In MDX, the SELECT statement is specified in a way that allows retrieving data with more than just two dimensions. Indeed, MDX provides you with the capability of retrieving data on one, two, or many axes.

The syntax of the SELECT statement is :

```
SELECT [ <axis_expression> , [ <axis_expression> …]]
```

The axis_expressions specified after the SELECT refer to the dimension data we are interested in retrieving. These dimensions are referred to as axis dimensions because the data from these dimensions are projected onto the corresponding axes.

Axis dimensions are used to retrieve multidimensional result sets. The set, a collection of tuples, is defined to form an axis dimension. MDX provides you with the capability of specifying up to 128 axes in the SELECT statement. The first five axes have aliases. They are COLUMNS, ROWS, PAGES, SECTIONS, and CHAPTERS. Axes can also be specified as a number, which allows you to specify more than five dimensions in your SELECT statement.

In the following you will only be using COLUMNS, and ROWS.

10. (10 points) In the query, type the following and execute: (The name in the FROM clause is the name of your cube. You can see it at the top of the new pane.) These numbers should match Answer 3 and Answer 5.

```
SELECT    { InternetTotalSales } ON COLUMNS,
          {[DimDate].[Calendar].[CalendarYear].&[2012],
        [DimDate].[Calendar].[CalendarYear].&[2013]}  ON ROWS
FROM [Adventure Works Internet Sales Model]
```

Use the Snipping Tool to capture the Results area of your query. Paste it into your Word document as Answer #8.

11. (10 points) Type and execute the following. These numbers should match Answer 4 and Answer 6 (although this query will have numbers that the earlier ones do not).

```
SELECT    {InternetTotalSales } ON COLUMNS,
          Non Empty (
          {DimDate.Calendar.CalendarYear.&[2012] *
           DimGeography.EnglishCountryRegionName.Members *
           DimProduct.Category.Category.Members,
           DimDate.Calendar.CalendarYear.&[2013] *
           DimGeography.EnglishCountryRegionName.Members *
           DimProduct.Category.Category.Members
          }
          ) ON ROWS
FROM [Adventure Works Internet Sales Model]
```

Use the Snipping Tool to capture the Results area of your query. Paste it into your Word document as Answer #9.

12. Submit your PDF to Assignment 10.