# BI Course Project – Part 2

*Business Intelligence Spring 2015*
*Due Date – No later than midnight Sunday May 3, 2015*
*This project must be completed in project teams as assigned by the instructor.*

## INTRODUCTION

To goal of this project is to design and implement a data warehouse for the BIke Project Company (hereafter BIke). BIke executives have declared that a new data warehouse is to be the center piece of their *Data Driven Decision Making* strategic initiative and must be capable of answering typical business intelligence (BI) questions

This project is broken into two parts. In part 1, you will use Microsoft's SQL Server Integration Services (SSIS) to construct a repeatable ETL process to move data from a set of relational OLTP table into a dimensional model. In part 2, and in keeping with best practices, you have decided that access to the completed data warehouse tables should be via an OLAP layer implemented using Microsoft's SQL Server Analysis Services (SSAS). As a proof-of-concept, you have been asked to implement a single MOLAP cube with dimensions, measures, and several hierarchies as defined below.

Lastly, using data from your deployed cube, you will be required to answer a few exploratory business intelligence questions using the SSAS connectivity and Pivot Table features of Microsoft Excel.

## PART 2 –OLAP (50%)

The second part of this project requires that construct an OLAP cube based on your work from Part 1. Essentially, you are required to integrate your previous work into a functioning and deployable multidimensional database implemented using SSAS. This requires that all of your DW dimension tables and your fact table are populated according to the Part 1 requirements. Lastly, you will use Excel Pivot Tables to access your OLAP cube to answer several highly exploratory business intelligence questions.

### OLAP Requirements

Your deployed multidimensional database must be named **GroupX** where the **X** is to be replaced with your group number. I recommend that you set the SSAS project properties to fully re-process your cube on each deployment. Here's why. Once you have created the data source and data source views in SSAS and you then subsequently make a change to the underlying data warehouse data (usually because of a mistake in your ETL process) fully re-processing your cube will ensure that the newly corrected data is being used. See the SSAS project properties **Deployment** page to make this change. Under the **Options** group set the **Processing Option** to *Full*, the **Transaction Deployment** to *False* and **Deployment Mode** to *Deploy All*.

## ***Common Requirements***

The list of requirements in this section is to be applied to all OLAP dimensions and the cube that you create. Any specific requirements for a dimension or fact table are spelled out in subsequent sections.

1. All dimension tables and the fact table from your data warehouse should be included in your OLAP design.
2. For the dimensions, retain all attributes from each data warehouse table in your data source view. For the cube attributes, refer to the *BIProjectCube* section below.
3. Change the "friendly name" of each table by removing the "**Dim**" and "**Fact**" prefixes from these tables.
4. For any dimension requiring a user-defined hierarchy, all of the individual attributes involved in the construction of the hierarchy must be "invisible" to client applications.

## ***DimDate***

The Date dimension will consist of the attributes and BI functionality necessary to define a full featured date dimension and must be named **DimDate**. Incorporate the following functionality into your date dimension.

1. When browsed, the **Date Key** attribute should display as ***yyyy mmm, dd***. Rename the **Date Key** attribute to **Date** making any changes to the current list of **DimDate** attributes necessary to accomplish this task.
2. Increase the user-friendliness of some of the attribute names as follows. Attributes beginning with the text *Calendar,* should be renamed to remove the *Calendar* prefix.
3. When browsed, the *MonthName* should appear in temporal and not alphabetical order
4. Create a user-defined hierarchy named **Calendar Hierarchy** consisting of the *Year-Quarter-MonthName-Date* attributes. This drill-down path will permit users to summarize cube measures at different levels of granularity
5. Create the attribute relationships to efficiently support the **Calendar Hierarchy**.
6. Configure time-base dimension business intelligence capabilities for attributes appearing in the **Calendar Hierarchy**. Use *Date > Calendar* SSAS BI attribute types.
7. Configure the time-base dimension business intelligence capabilities to support the **IsHoliday** time based AttributeType. Note, this will require adding a new named calculation to your date dimension that reflects whether a date is any one of the three holidays supported in by your DW date dimension.

## ***DimProduct***

The product dimension will consist of the attributes and BI functionality necessary to define a full featured product dimension and must be named **DimProduct**. Incorporate the following functionality into your product dimension.

1. Create a user-defined hierarchy named **Products Hierarchy** consisting of *Products, ProductSubcategory, and ProductCategory* attributes. Your attribute names may, but should not, be different.
2. Create the attribute relationships to efficiently support the **Products Hierarchy**.

## ***DimCustomer***

The customer dimension will consist of the attributes and BI functionality necessary to define a full featured customer dimension and must be named **DimCustomer**. Incorporate the following functionality into your customer dimension.

1. When browsed, the **CustomerID** attribute should display a customer's full name. This will require adding a new **Customer Name** named calculation. 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 fist initial of the middle name, not the entire name.
2. Rename the **CustomerID** attribute to **Customer Name**.
3. Create a user-defined hierarchy for customers named **Customer Geography**. The hierarchy should be composed of the following attributes – **Country Code** > **State Name** > **City** > **Full Name**.
4. Create the attribute relationships to efficiently support the **Customer Geography** hierarchy.
5. Use SSAS's built-in automatic grouping feature (i.e., discretization) to create 4 "buckets" for the **YearlyIncome** attribute.
6. 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.*

### ***DimSalesPerson***

The sales person dimension will consist of the attributes and BI functionality necessary to define a full featured sales person dimension and must be named **DimSalesPerson**. Incorporate the following functionality into your sales person dimension.

1. When browsed, the **Sales Person ID** attribute should display a sales person's full name. Note, this will require adding a new **Sales Person Name** named calculation. Ex: `Mary Crawley` when there is no middle name and `Mary A.Crawley` when there is a middle name. Note that this only use the fist initial of the middle name, not the entire name.
2. Rename the **Sales Person ID** attribute to **Sales Person Name**.
3. Management would like to have a hierarchy to view sales by store id and then sales person name. This hierarchy is to be named **Sales Person Hierarchy**. Moreover, when viewing this hierarchy management would like to see the **StoreDescription** attribute value displayed and NOT the numeric store **StoreID** surrogate key value. The problem? The **StoreDescription** attribute is in the **DimStore** table and not the **DimSalesPersonTable**. This requires a bit of thinking and applying your skills in novel ways. That said, you should be quite capable of reasoning your way through this challenge.
4. Create the attribute relationships to efficiently support the **Sales Person** hierarchy.

### ***BIProjectCube***

Your cube must be named **BIProjectCube**. Your cube will consist of a single measures group with the following two measures – *Order Quantity* and *Sales Amount* where *Sales Amount* is a named calculation that is the product of *OrderQty* and *UnitPrice*.

1. Management would like to see several time-based BI capabilities incorporated into the cube. Specifically, Growth and Growth % for the Sales Amount measure for each of the time based attributes **Year**, **Quarter** and **Month**. Add the necessary time-based BI capabilities to your both your cube and date dimension to make this happen.

## BI Questions (a.k.a., Proof of Concept)

Using Excel connectivity features, establish an external connection to your deployed OLAP multidimensional database. Name your connection *CubeConn*. For each question that follows, create a new Pivot Table in a separate worksheet that reflects the data in the underlying OLAP cube. For each question, provide a brief written discussion (in a Word document) of your findings. Your answers to all of the questions below must be supported by the **data**! That means your answers must incorporate some amount of data analysis or data description to be considered correct.

1. Construct a pivot table to explore *SalesAmount* by *Year*, *Quarter* and *Store.* Augment your pivot table with a pivot chart to display the *SalesAmount* by *Year* and *Quarter*. Describe any trends you see in the data.
2. Construct a pivot table to explore *SalesAmount* by *ProductSubcategory,* and several (your choice) customer demographic variables like gender, income, number of children. Describe any relationships you see in the data.

## Part 2 Deliverables

1. A functioning SSAS project in ZIP file format. Your folder structure should be as follows:

```
GroupX
        SSIS_ETL_Project
        SSAS_OLAP_Project
        DataSources
        BI Questions Analysis .xlsx
        BI Questions Response.docx
```

   The SSAS_*OLAP_Project* subfolder should contain all your SSAS solution and project files. The *DataSources* subfolder should contain all data required by your project including your data warehouse database files (.MDF, .LDF) constructed in **Part 1** and a backup of your SSAS multidimensional database as described below. Note also that there is only one ZIP file to be submitted for both Parts 1 and 2 of the project. The *SSIS_ETL_Project* folder shown above contains your SSIS project files from **Part 1** of this project.

2. A backup of the SSAS multidimensional database. To create your backup, follow the instructions listed under **Part 2 Helpful Hints / Suggestions** located in the *Course Project* folder. Your backup of the SSAS multidimensional database must be located in the *DataSources* folder.
3. The Excel file used to construct your pivot tables for the BI questions listed above. The file must be named ***BI Questions Analysis.xlsx.*** This file must be located directly beneath the **GroupX** folder as shown above.
4. A Word document containing your written responses for the BI questions listed above. The file must be named ***BI Questions Response.docx.*** This file must be located directly beneath the **GroupX** folder as shown above.