|  |
| --- |
| **National University of Computer and Emerging Sciences** |
| Lab Manual 3  “**OLAP Cube**” |
|  |
| Data Warehousing and Data Mining |
|  |
|  |
|  |

|  |  |
| --- | --- |
| Course Instructor | Ishaq Raza |
| Lab Instructor(s) | Farhan Azhar  Nouman Ali |
| Section | CS |
| Semester | Fall 2019 |

|  |
| --- |
|  |

Department of Computer Science

FAST-NU, Lahore, Pakistan

**Contents**

*Total (160 minutes)*

* + **Introduction to OLAP** *(30 minutes)*
  + **Creating and Deploying OLAP cube**  *(70 minutes)*
  + **Exercise**  *(40 minutes)*
  + **Evaluation** *(20 minutes)*

*The purpose of this lab is to make you familiar with OLAP cube. We will start with the introduction of OLAP cube. Then we will see how to create and deploy OLAP cube SQL server business studio*

*.*

**Understanding Analysis Services**

The basic idea of OLAP is fairly simple. Let’s think about that book ordering data for a moment. Suppose you want to know how many people ordered a particular book during each month of the year. You could write a fairly simple query to get the information you want. The catch is that it might take a long time for SQL Server to churn through that many rows of data.

And what if the data was not all in a single SQL Server table, but scattered around in various databases throughout your organization? The customer info, for example, might be in Oracle database and supplier information in a legacy base database.

SQL Server can handle distributed heterogeneous queries, but they’re slower.

What if, after seeing the monthly numbers, you wanted to drill down to weekly or daily numbers? That would be even more time-consuming and require writing even more queries.

This is where OLAP comes in. The basic idea is to trade off increased storage space now for speed of querying later. OLAP does this by pre calculating and storing aggregates. When you identify the data that you want to store in an OLAP database,

Analysis Services analyzes it in advance and figures out those daily, weekly, and monthly numbers and stores them away (and stores many other aggregations at the same time). This takes up plenty of disk space, but it means that when you want to explore the data you can do so quickly.

Later in the chapter, you’ll see how you can use Analysis Services to extract summary information from your data. First, though, you need to familiarize yourself with a new vocabulary. The basic concepts of OLAP include:

• Cube

• Dimension table

• Dimension

• Hierarchy

• Level

• Fact table

• Measure

• Schema

**Cube**

The basic unit of storage and analysis in Analysis Services is the cube. A cube is a collection of data that’s been aggregated to allow queries to return data quickly. For example, a cube of order data might be aggregated by time period and by title, making the cube fast when you ask questions concerning orders by week or orders by title.

Cubes are ordered into dimensions and measures. The data for a cube comes from a set of staging tables, sometimes called a star-schema database. Dimensions in the cube come from dimension tables in the staging database, while measures come from fact tables in the staging database.

**Dimension table**

A dimension table lives in the staging database and contains data that you’d like to use to group the values you are summarizing. Dimension tables contain a primary key and any other attributes that describe the entities stored in the table. Examples would be a Customers table that contains city, state and postal code information to be able to analyze sales geographically, or a Products table that contains categories and product lines to break down sales figures.

**Dimension**

Each cube has one or more dimensions, each based on one or more dimension tables.

A dimension represents a category for analyzing business data: country or product line in the examples above. Frequently, a dimension has a natural hierarchy so that lower results can be “rolled up” into higher results. For example, in a geographical level you might have city totals aggregated into state totals, and state totals into country totals.

**Hierarchy**

A hierarchy can be best visualized as a node tree. A company’s organizational chart is an example of a hierarchy. Each dimension can contain multiple hierarchies; some of them are natural hierarchies (the parent-child relationship between attribute values occur naturally in the data), others are navigational hierarchies (the parent-child relationship is established by developers.)

**Level**

Each layer in a hierarchy is called a level. For example, you can speak of a week level or a month level in a fiscal time hierarchy, and a city level or a country level in a geography hierarchy.

**Fact table**

A fact table lives in the staging database and contains the basic information that you wish to summarize. This might be order detail information, payroll records, drug effectiveness information, or anything else that’s amenable to summing and averaging. Any table that you’ve used with a Sum or Avg function in a totals query is a good bet to be a fact table. The fact tables contain fields for the individual facts as well as foreign key fields relating the facts to the dimension tables.

**Measure**

Every cube will contain one or more measures, each based on a column in a fact table that you’d like to analyze. In the cube of book order information, for example, the measures would be things such as unit sales and profit.

**Schema**

Fact tables and dimension tables are related, which is hardly surprising, given that you use the dimension tables to group information from the fact table. The relations within a cube form a schema. There are two basic OLAP schemas: star and snowflake. In a star schema, every dimension table is related directly to the fact table.

In a snowflake schema, some dimension tables are related indirectly to the fact table.

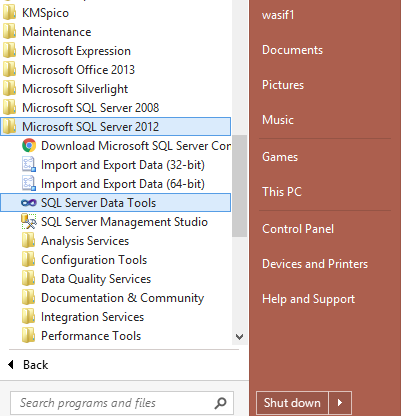
For example, if your cube includes OrderDetails as a fact table, with Customers and Orders as dimension tables and Customers is related to Orders, which in turn is related to OrderDetails, then you’re dealing with a snowflake schema.

Developing an OLAP Cube

For creation of OLAP Cube in Microsoft BIDS Environment, follow the 10 easy steps given below.

**Step 1: Start BIDS Environment**

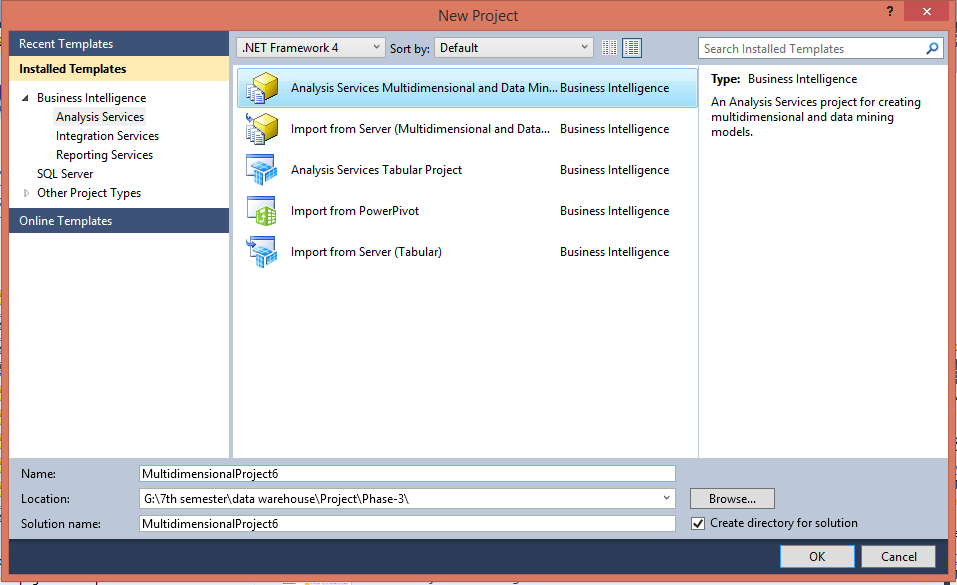
Click on **Start Menu** -> Microsoft **SQL Server 2012** -> Click **SQL Server Data Tools**



**Step 2: Start Analysis Services Project**

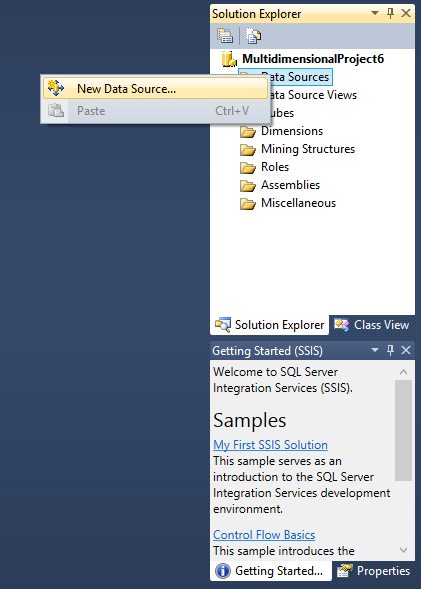
Click **File** -> **New** -> **Project**

Select **Analysis services** in **Business Intelligence** component in the left panel of window. Give project a name and click ok.

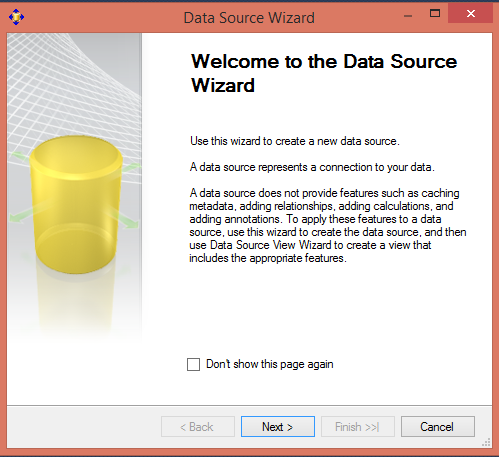


**Step 3: Creating New Data Source**

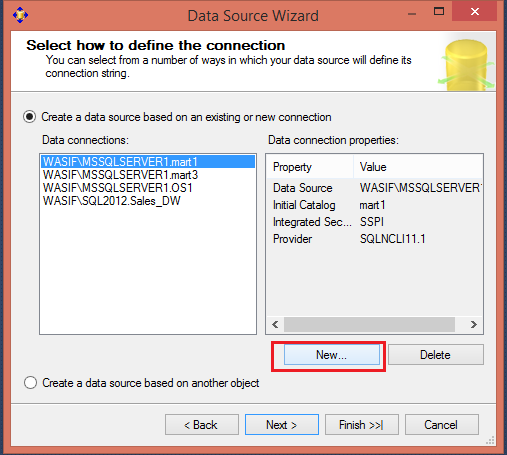
3.1 In Solution Explorer, Right click on **Data Source** -> Click **New Data Source**



3.2 Click on **Next**

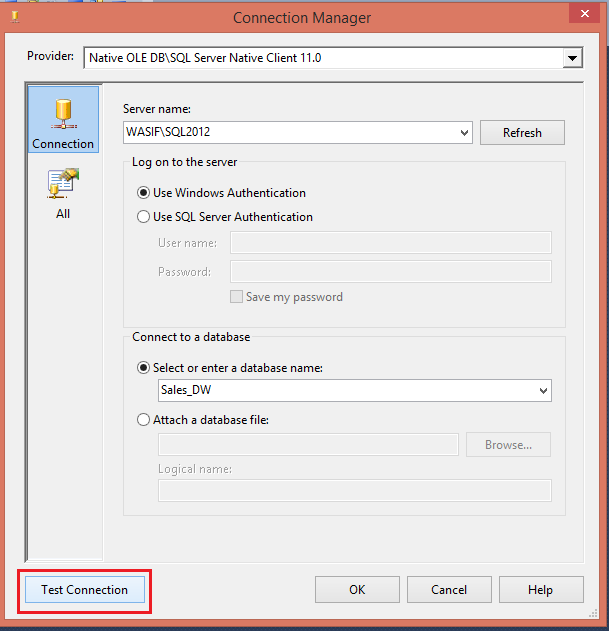


3.3 Click on New **Button**

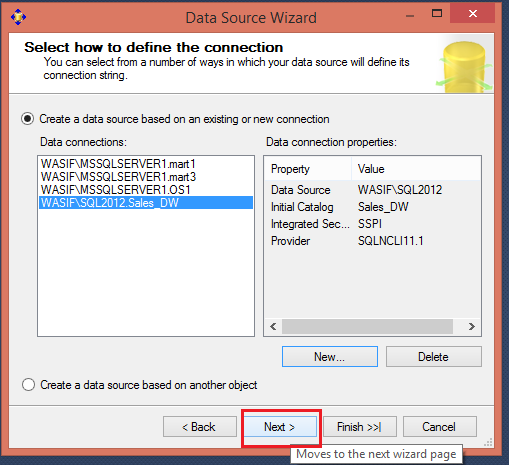


3.4 Creating **New connection**

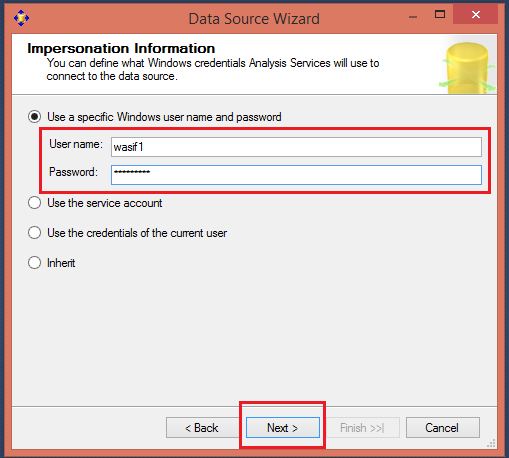
1. Specify Your SQL **Server Name** where your Data Warehouse was created
2. Select Radio Button according to your **SQL Server Authentication** mode
3. Specify your **Credentials** using which you can connect to your SQL Server
4. Select database Sales\_DW.
5. Click on **Test Connection** and verify for its success
6. Click **OK**.



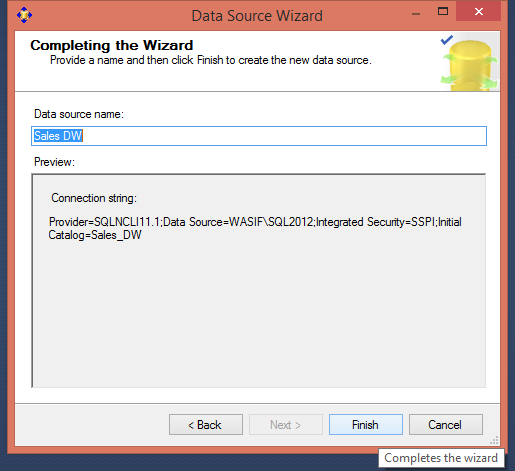
3.5 Select Connection created in **Data Connections**-> Click **Next**



3.6 Select Option **Windows username and password** and **enter username** and **password** of your windows.

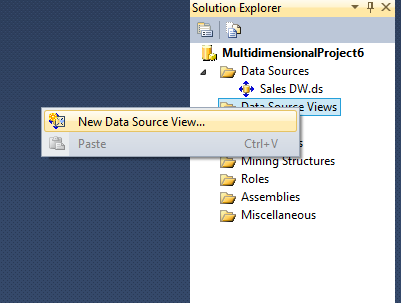


3.7 Assign Data Source **Name** -> Click **Finish**



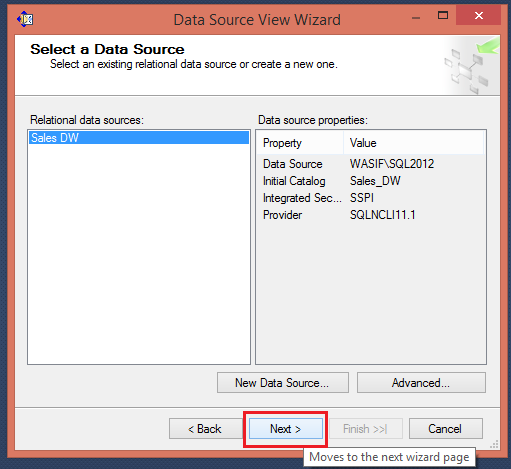
**Step 4: Creating New Data Source View**

4.1 In the Solution Explorer, Right Click on **Data Source View** -> Click on **New Data Source View**

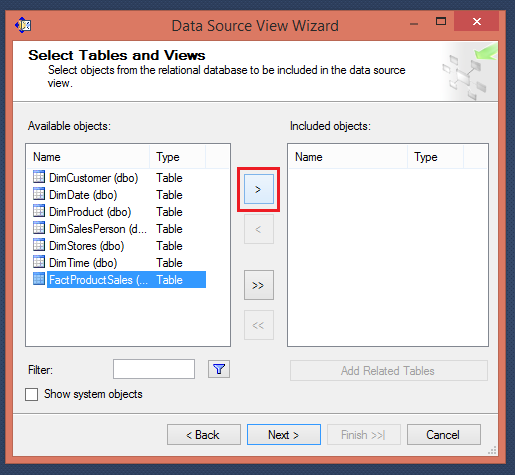


4.2 Click **Next**

4.3 Select**Relational Data Source**we have created previously (Sales\_DW)-> Click **Next**



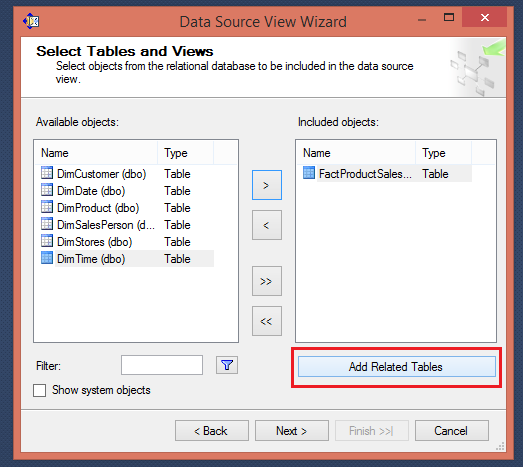
4.4 First move your **Fact Table** to the right side to include in object list.



SelectFactProductSales Table -> Click on Arrow Button to move the selected object to Right Pane.

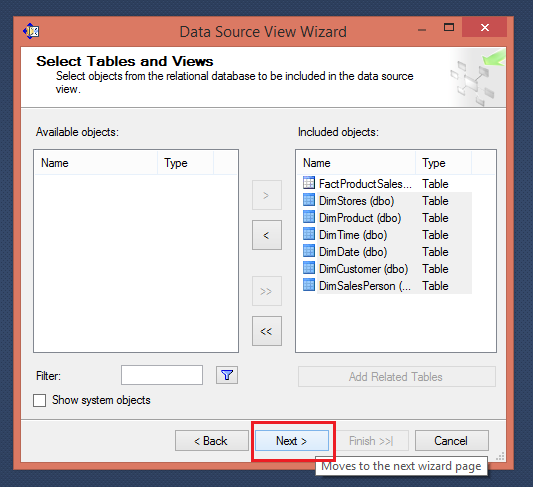
4.5 Now to **add dimensions** which are **related** to your **Fact Table**, follow the given steps:

Select **Fact Table** in Right Pane (Fact product Sales) -> Click On**Add Related Tables**

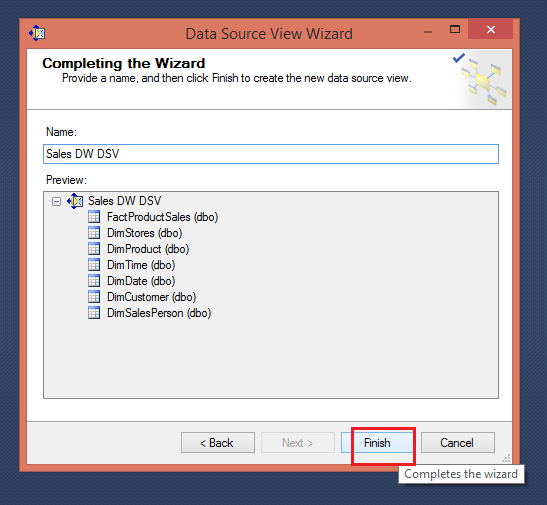


4.6 It will add all associated dimensions to your Fact table as per relationship specified in your SQL DW (Sales\_DW).

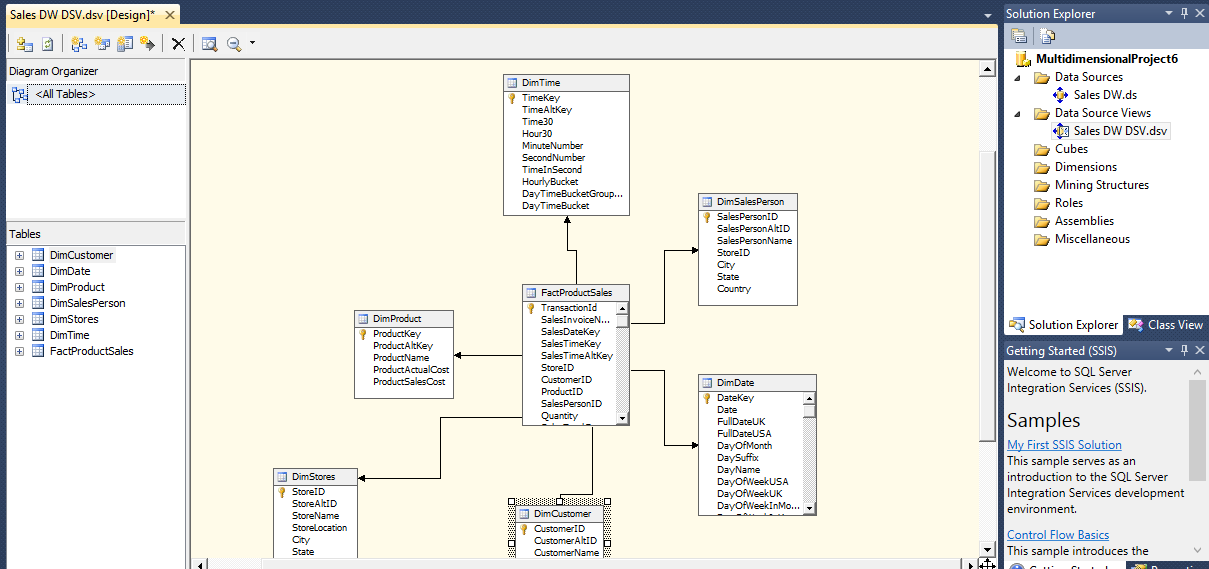
Click **Next**.



4.7 Assign**Name (SalesDW DSV)->**Click**Finish**

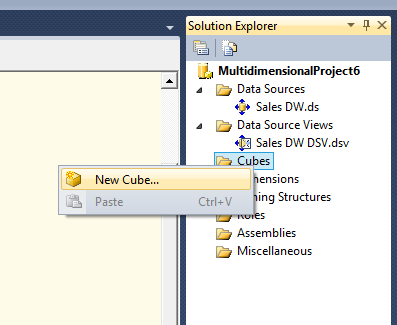


4.8 **Now Data Source View is ready to use.**



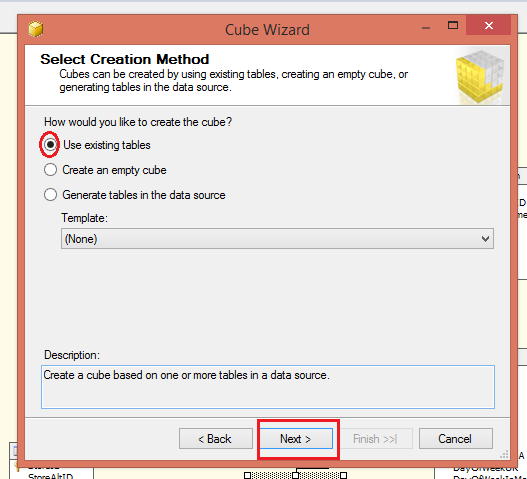
**Step 5: Creating New Cube**

5.1 In Solution Explorer -> Right Click on **Cube->**Click**New Cube**

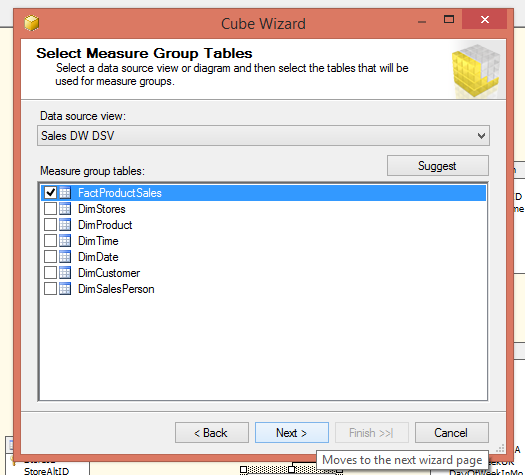


5.2 Click**Next**

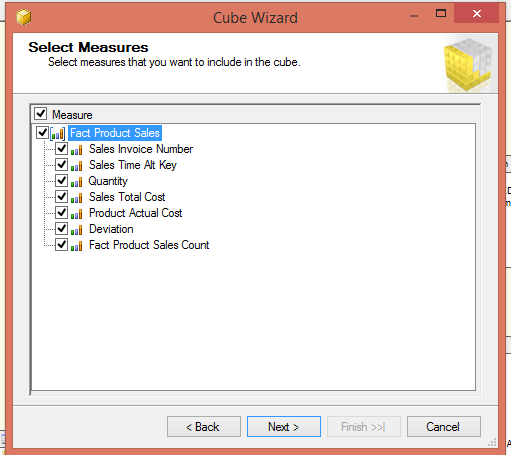
5.3 Select Option**Use existing Tables ->**Click**Next**



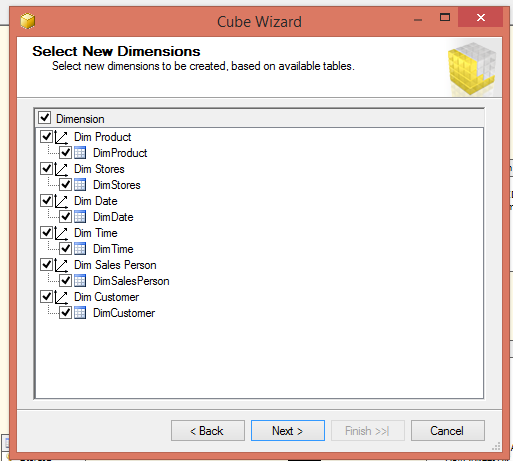
5.4 Select Fact Table Name from**Measure Group Tables (FactProductSales) ->**Click**Next**



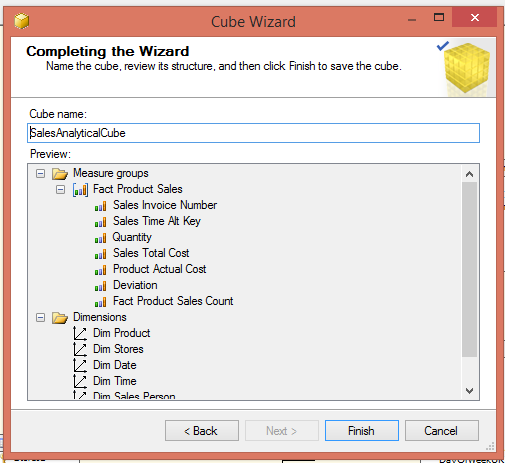
5.5 Choose**Measures**from the List which you want to place in your Cube --> Click **Next**



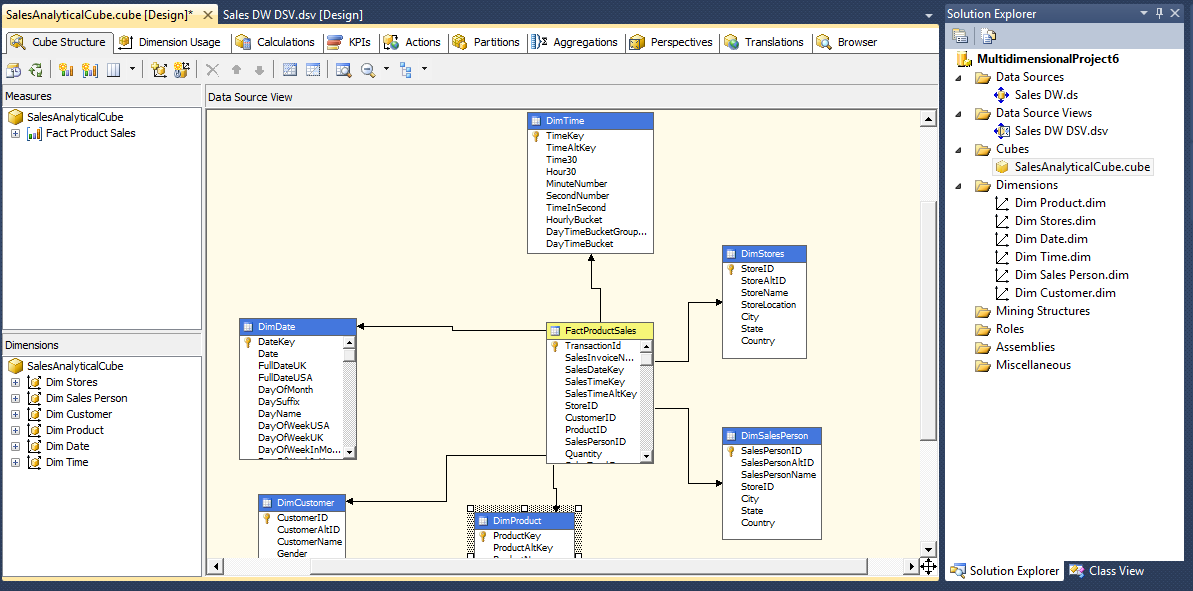
5.6 Select All **Dimensions** here which are associated with your Fact Table-> Click **Next**



5.7 Assign **Cube Name** (SalesAnalyticalCube) -> Click **Finish**

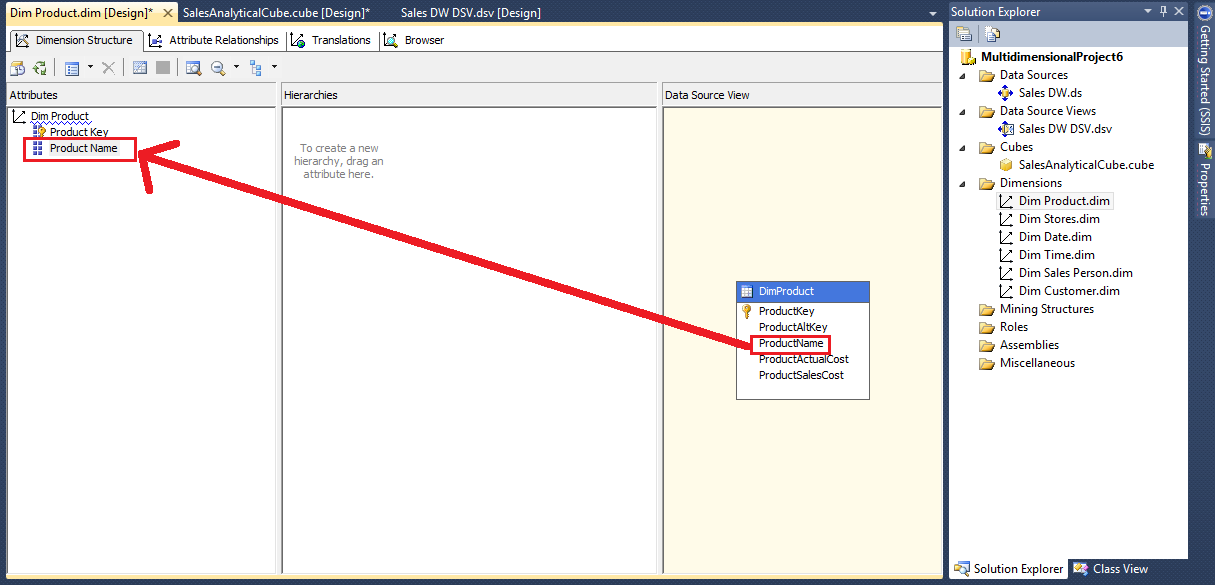


5.8 Now your Cube is ready, you can see the newly created cube and dimensions added in your solution explorer.



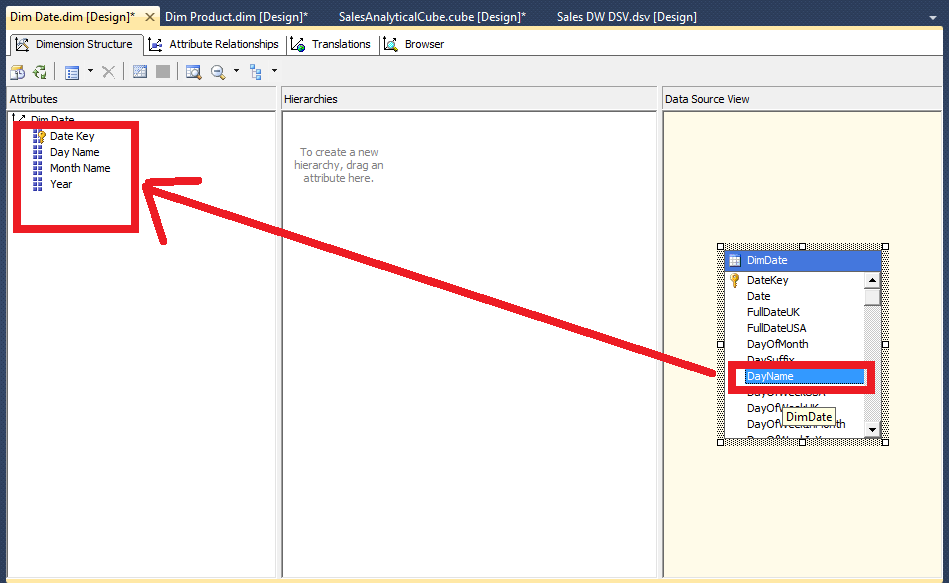
**Step 6: Dimension Modification**

In Solution Explorer, double click on dimension **Dim Product ->**Drag and Drop Product Name from Table in Data Source View and Add in Attribute Pane at left side.



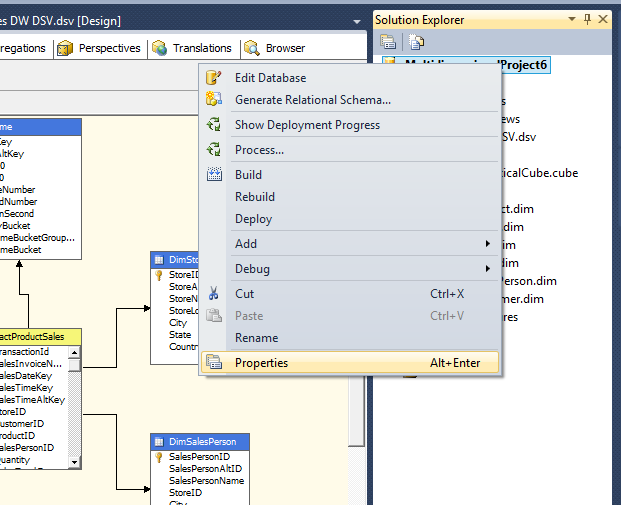
**Step 7: Modify Other Dimensions**

Double click on Dim date. Drag drop Day Name, Month Name and Year.



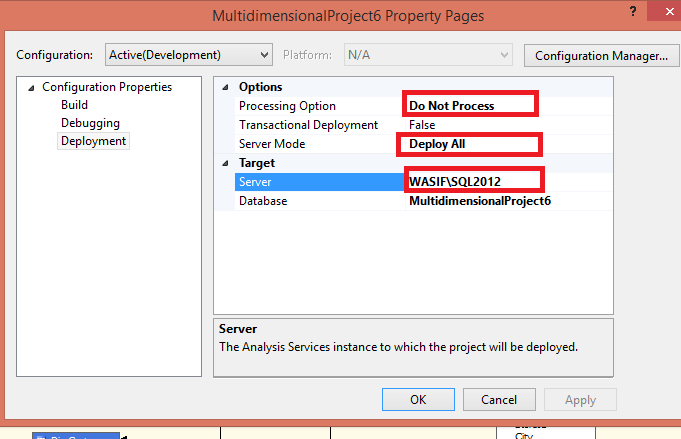
**Step 8: Deploy the Cube**

8.1 In Solution Explorer, right click on Project Name (Multidimensionalproject6) -- > Click **Properties**

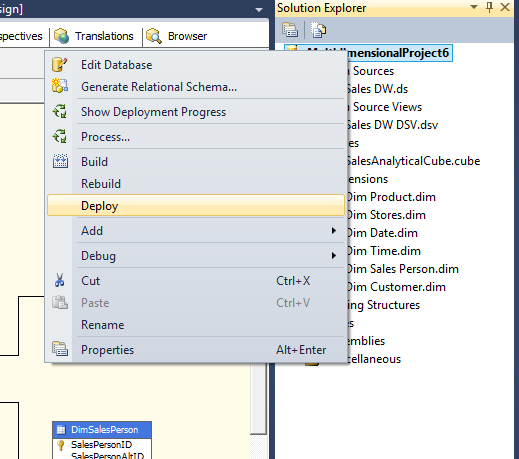


8.2 Set **Deployment Properties** First

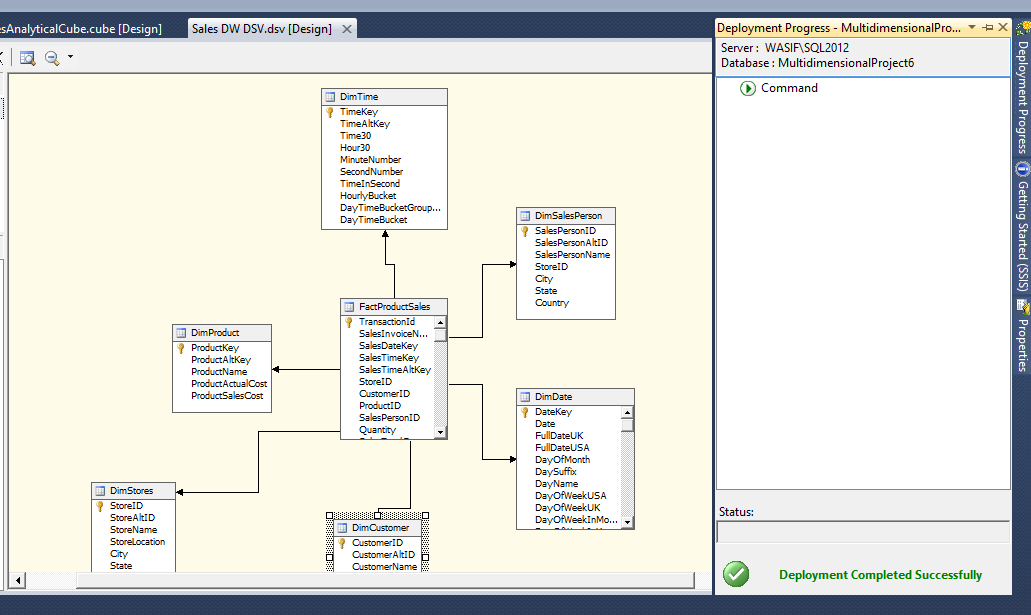
In **Configuration Properties** select **Deployment**. Enter the **server** name, select **Processing Option->Do not Process**, **Server Mode->Deploy All** and click **OK.**



8.3 In Solution Explorer, right click on **Project Name** (Multidimensionalproject6) -- > Click **Deploy**

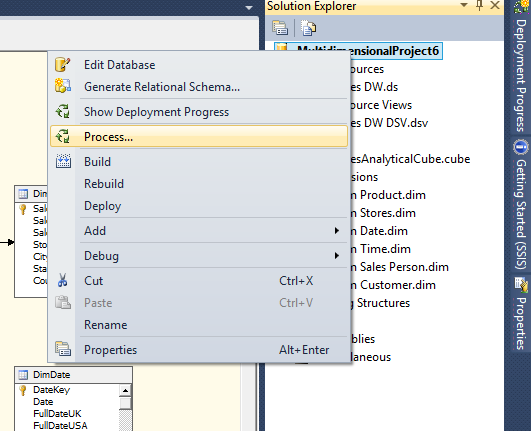


8.4 Once Deployment will finish, you can see the message**Deployment Completed**in deployment Properties.

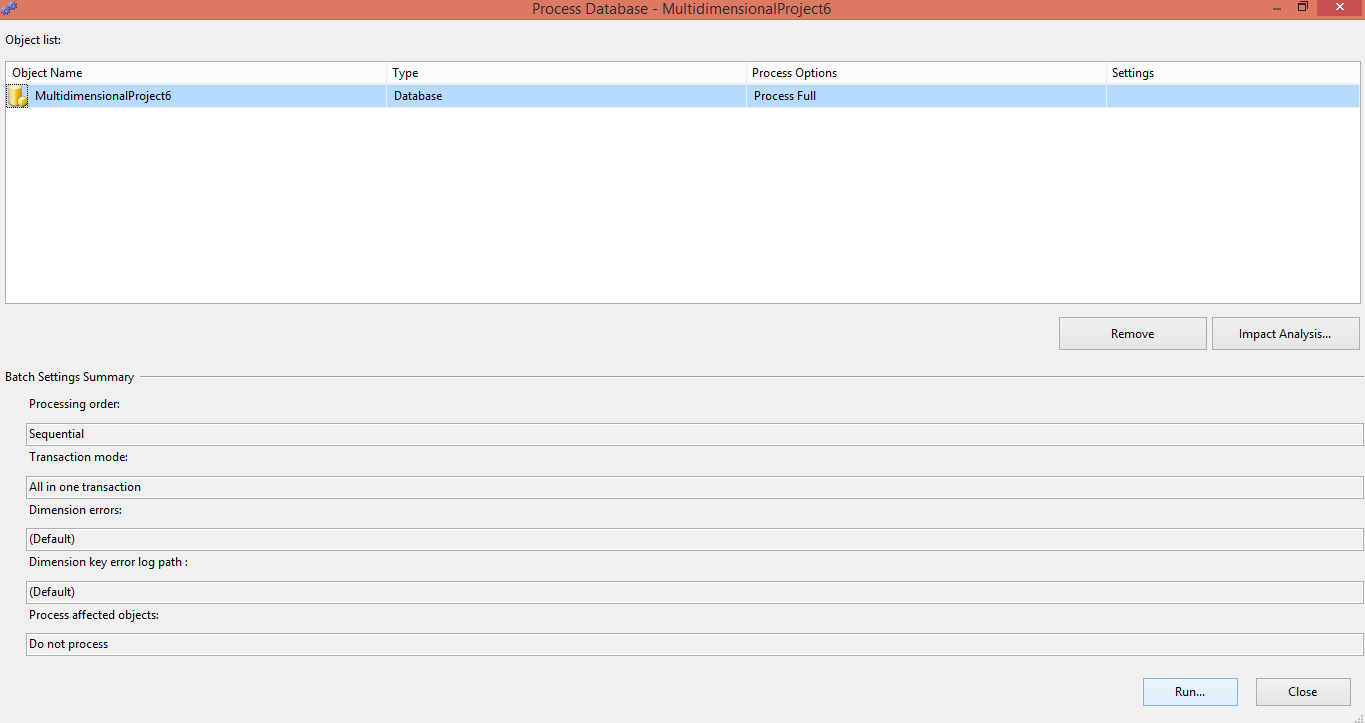


**Step 9: Process the Cube**

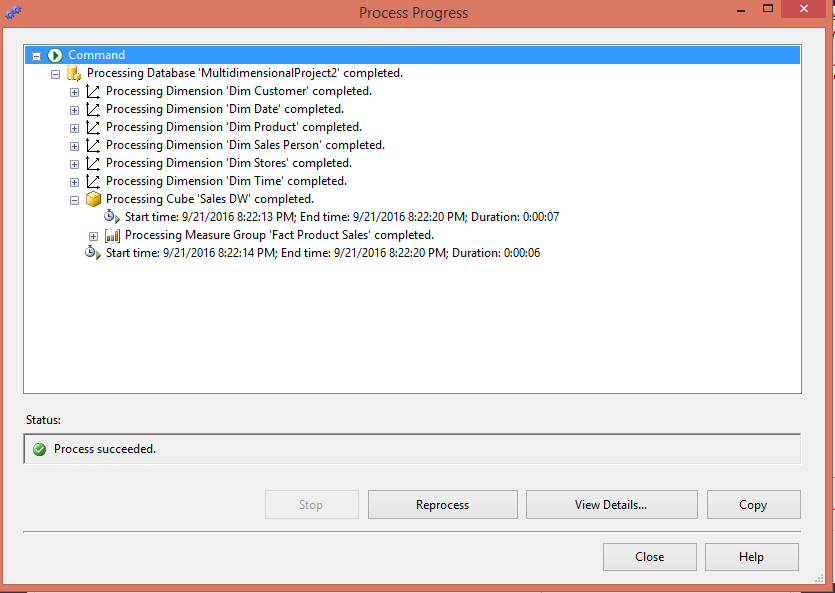
9.1 In Solution Explorer, right click on Project Name (Multidimensionalproject6) -- > Click **Process**



9.2 Click on **Run** button to process the Cube

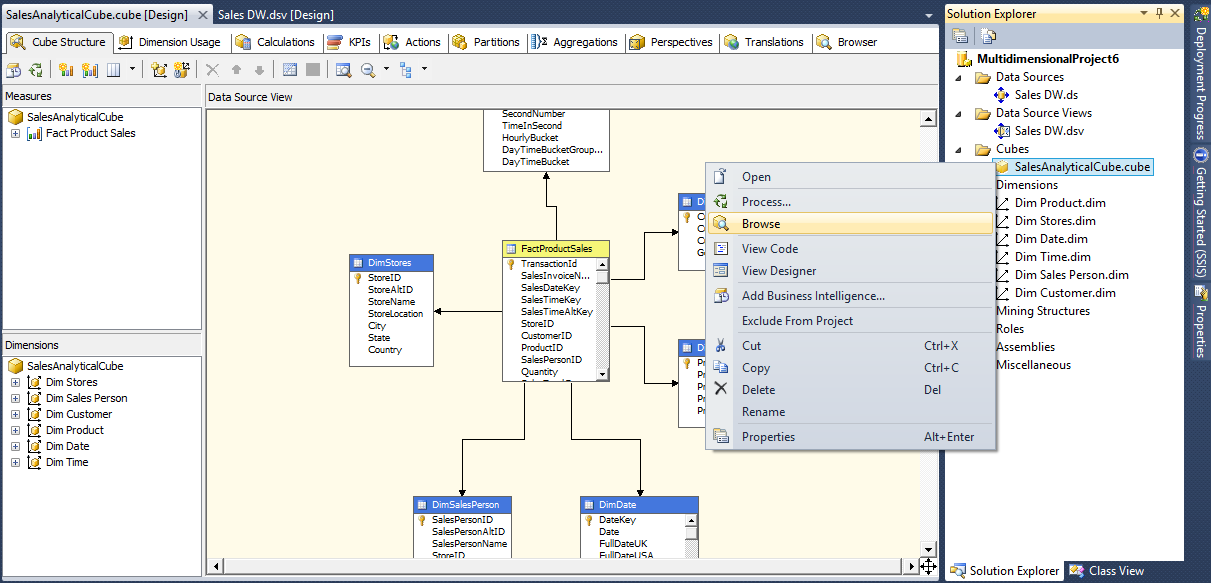


9.3 Once processing is complete, you can see **Status** as **Process Succeeded**-->Click **Close**to close both the open windows for processing one after the other.



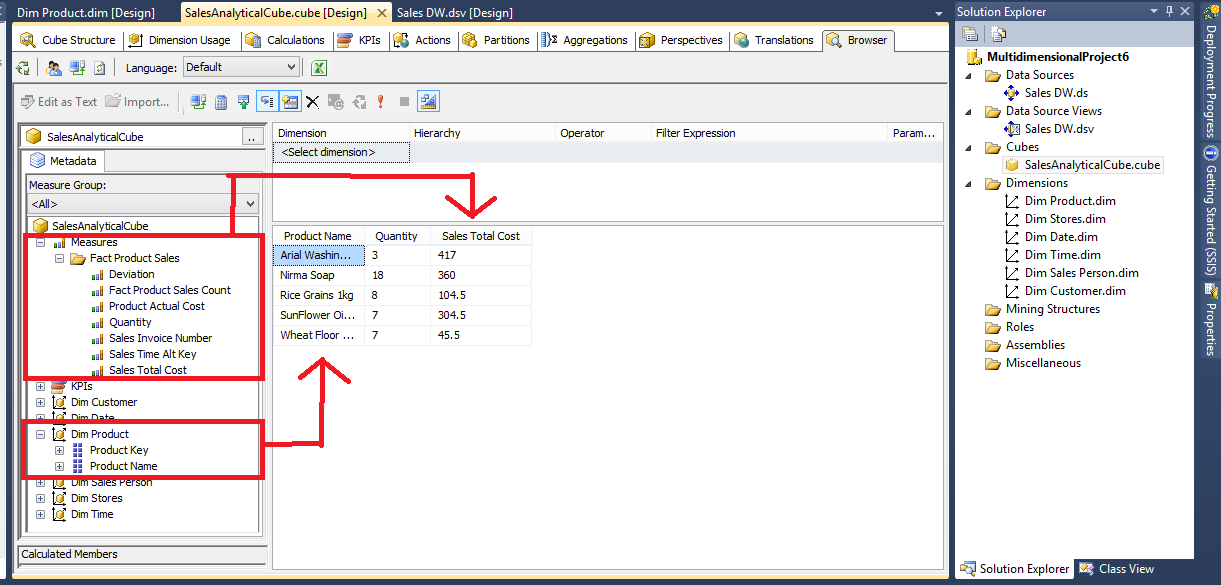
**Step 10: Browse the Cube for Analysis**

10.1 In Solution Explorer, right click on Cube Name (SalesDataAnalysisCube) -- > Click **Browse**



Now to **Browse Our Cube**

1. Drag drop **Quantity** and **Sales Total Cost** from **Measures.**
2. Drag drop **Product Name** from **Dim Product** dimension.



**Exercise:**

1. What can you interpret from the cube developed above? What insights can be found using the cube?
2. Create a cube to browse the aggregated product actual cost associated with each sales person.

**Additional Material:**

* Data WareHouse SQLScript (Script to create the Database and fill the columns)