|  |
| --- |
| Department of Software Engineering  Mehran University of Engineering and Technology, Jamshoro |

|  |  |  |  |
| --- | --- | --- | --- |
| Course: SWE324 - Data Warehousing and Data Mining | | | |
| Instructor | Rabeea Jaffari | **Practical/Lab No.** | 06,07,08 |
| Date | 07 May 2019 | **CLOs** | CLO-4: P3 & P4 |
| Signature |  | **Assessment Score** | 1 Marks |

|  |  |
| --- | --- |
| Topic | To become familiar with DW implementation, query execution and result analysis |
| Objectives | * To learn physical DW Design * To learn executing queries in DW * To learn analyzing results of DW queries |

|  |
| --- |
| Lab Discussion: Theoretical concepts and Procedural steps |

**DW Physical design:** Physical design is the actual implementation of the DW created in logical design with M/ERD into the system memory with the help of any appropriate product. The physical design of a DW can be either:

* Relational
* Multi-dimensional

**DW PRODUCT:** Any suitable DW product can be used to implement the DW system. Available choices are:

* SQLSERVER (Relational DW)
* TERADATA (Relational DW)

**SQL SERVER:** Microsoft SQL Server is a relational database management system, or RDBMS, that supports a wide variety of transaction processing, business intelligence and analytics applications in corporate IT environments. It's one of the three market-leading database technologies, along with Oracle Database and IBM's DB2.

**Components:** The Microsoft SQL Server product is made up of four primary components as follows:

1. **Database Engine:** This part of SQL Server actually creates and drives relational databases.
2. **SQL Server Analysis Services (SSAS):** SSAS is the data-analysis component of SQL Server. It can create OLAP (OnLine Analytical Processing) cubes — sophisticated programming objects for organizing data inside a relational database — and do data mining (pulling relevant data out of a database in response to an ad-hoc question).
3. **SQL Server Reporting Services (SSRS):** SSRS is a component of SQL Server that provides reporting regardless of a database’s operating system.
4. **SQL Server Integration Services (SSIS):** SSIS is a component of SQL Server that does the Extract, Transform, and Load (ETL) process that cleans up and formats raw data from source systems for inclusion in the database as ready-to-use information.

This lab would be covering an introduction of both the DW implementation approaches using SQL Server Analysis services and SQL Server Data tools.

**SQL Server Analysis Services:** Analysis Services is an analytical data engine used in decision support and business analytics. It provides enterprise-grade semantic data models for business reports and client applications such as Power BI, Excel, Reporting Services reports, and other data visualization tools.

A typical workflow includes creating a tabular or multidimensional data model project in SQL Server data tools (Visual studio shell), deploying the model as a database to a server instance, setting up recurring data processing, and assigning permissions to allow data access by end-users. When it's ready to go, your semantic data model can be accessed by client applications supporting Analysis Services as a data source. It should be noted that SQL SERVER only allows relational DW modeling but SQL SERVER Analysis services allows such data to be visualized either in tabular format (Star schema) as well as multidimensional format (cubes with hierarchies).  
  
**Note:** Install SQL Server 2012 or above along with SQL server management studio, SQL server analysis services and SQL server Data tools to take advantage of the business intelligence services for DWs.

**Tutorial Scenario:**

This tutorial is based on Adventure Works Cycles, a fictitious company. Adventure Works is a large, multinational manufacturing company that produces bicycles, parts, and accessories for commercial markets in North America, Europe, and Asia. With headquarters in Bothell, Washington, the company employs 500 workers. Additionally, Adventure Works employs several regional sales teams throughout its market base.

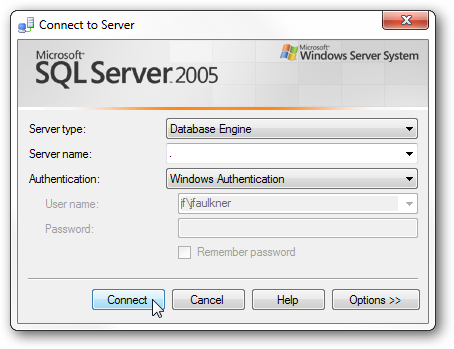
To better support the data analysis needs of sales and marketing teams and of senior management, you are tasked with creating a tabular model or a multidimensional model for users to analyze Internet sales data in the AdventureWorksDW sample data warehouse. Appropriate version of the data warehouse according to your SQL SERVER version can be downloaded from the site below:  
  
 <https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks>

This data sample contains data from about 9 years.

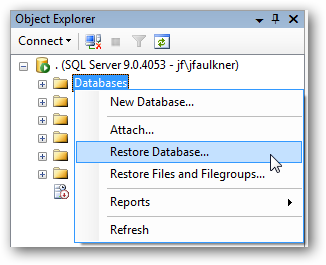
**Step 1: Importing the AdventureWorksDW in SQL SERVER Management Studio:**

After downloading the appropriate version of the DW from the above site you can import it into SQL SERVER Management studio by following the steps below:

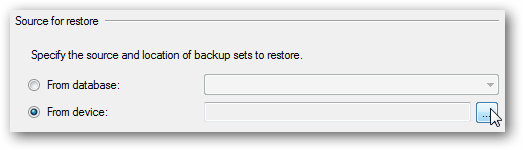
1. Open SQL Server management studio and connect to the Database server using either “Windows Authentication” mode or “SQL SERVER Administrator” mode as per your feasibility.



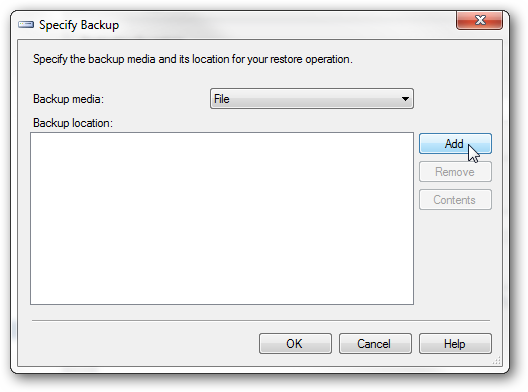
1. Once logged in, right click on the Databases folder and select ‘Restore Database’.



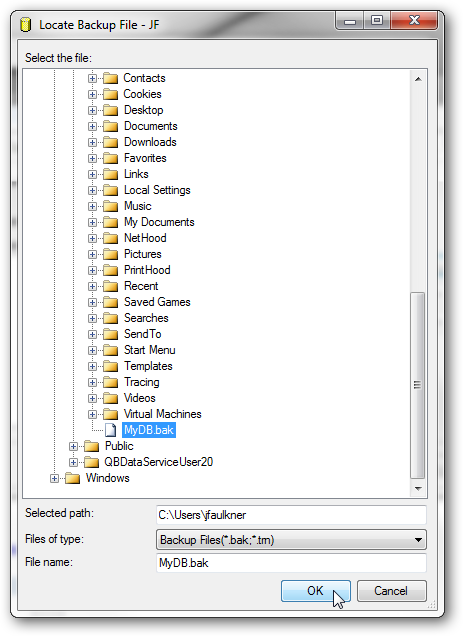
1. Click the ellipses button next to ‘From device’ under the ‘Source for restore’ section.

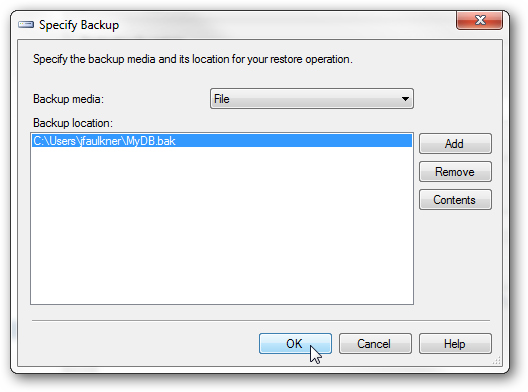


1. Set ‘File’ as the backup media and then click ‘Add’.

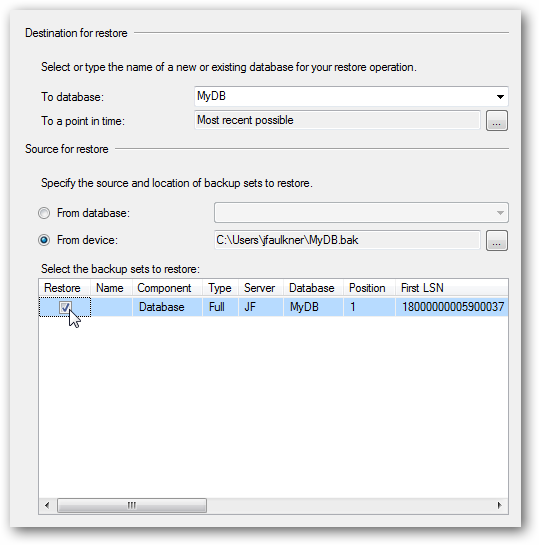


1. Browse to the SQL backup (BAK) file of the AdventureWorks DWH that you downloaded, you want to restore.





1. In the Restore Database dialog, type or select the name of the database (e.g.: AdventureWorksDW) you want this backup restored to. Next, select the restore point you want to use. Since a SQL backup file can hold multiple backups you may see more than one restore point listed.



1. Finally your DWH has been imported into a database name specified by you above (for e.g.: “AdventureWorksDW”).
2. To view the data in it, expand the databases option in the left pane, locate “AdventureWorksDW”, expand it, go to the “Tables” option within it and expand it to view the tables in this DWH.
3. The tables starting with the prefix “Dim” represent the dimensions while those starting with “Fact” represent the facts.

**Step 2:** After the DWH data has been loaded into SQL SERVER, you can visualize it either using a tabular format (Star schema i.e. one central fact table connected to all the dimension tables) or multidimensional format (i.e. cubes representing facts with measures and dimensions that assist in calculating those measures along with any hierarchies if applicable.).

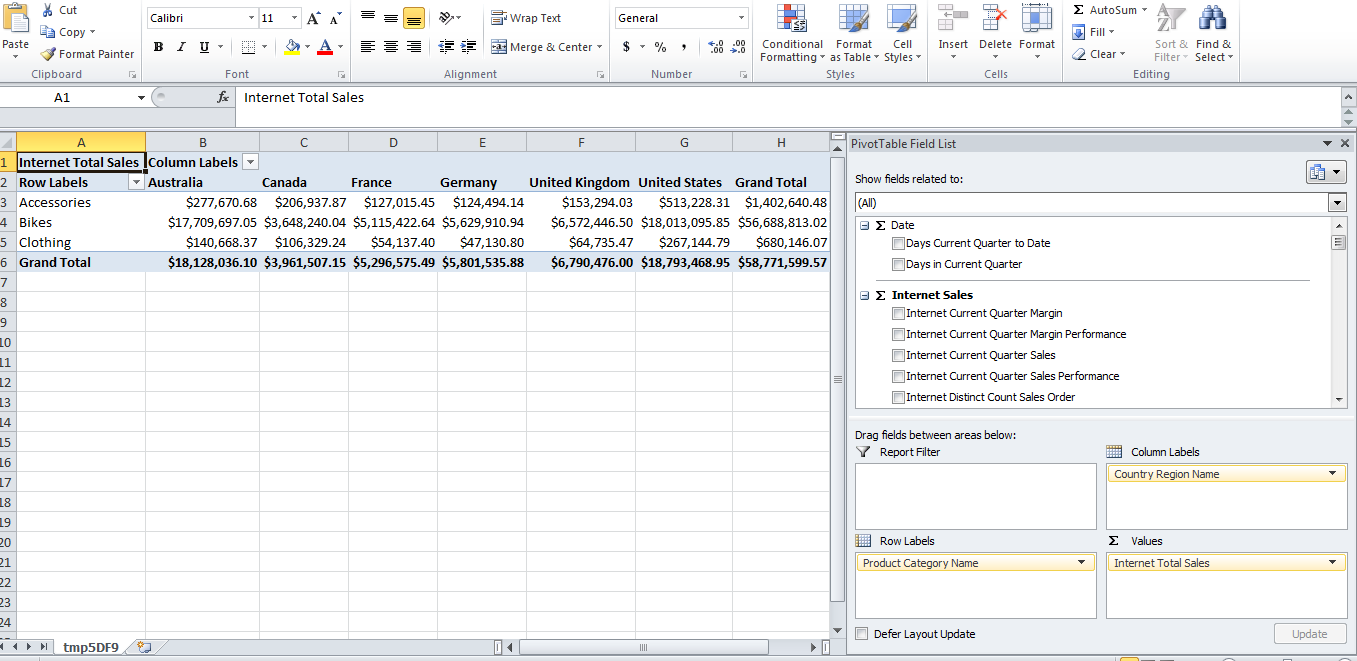
**Option 1: Tabular modeling:**

To implement DW in tabular format using SQL Server Data tools and to analyze it using SQL Server analysis services. Follow the tutorial below:

<https://docs.microsoft.com/en-us/sql/analysis-services/tutorial-tabular-1400/as-adventure-works-tutorial?view=sql-analysis-services-2017>

**Option 2: Multidimensional modeling:**

To implement DW in tabular format using SQL Server Data tools and to analyze it using SQL Server analysis services. Follow the tutorial below:  
  
<https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-modeling-adventure-works-tutorial?view=sql-analysis-services-2017>   
  
(Note: perform till lesson 3 of the above tutorial. After lesson 3 data will be in a format to be analyzed using Excel).

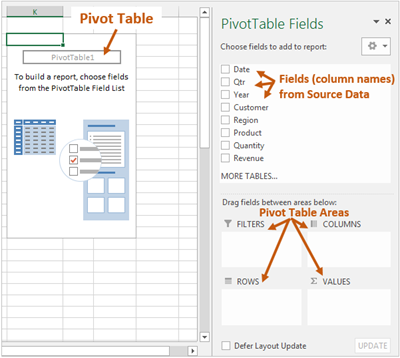


The output from both the modeling approaches (either tabular or multidimensional) results in the data which can be analyzed using Excel Pivot table as shown above. For example, the pivot report above shows the total sales of product categories with respect to sales regions. Pivot charts can be created from such data to better visualize it.

**EXCEL PIVOT TABLE (EXPLAINED):**

Pivot Tables are one of the most powerful and useful tools in Excel.  They will save you a lot of timeby allowing you to quickly summarize large amounts of data into a meaningful report.

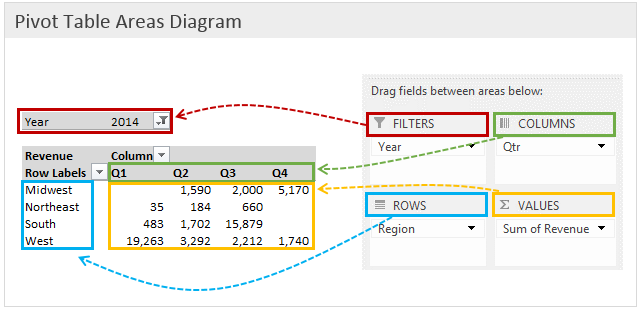
Let’s suppose we have entire cube’s Internet sales data as an example for the source of the pivot table.  The Pivot table opens by default when you finish the modeling tutorial.



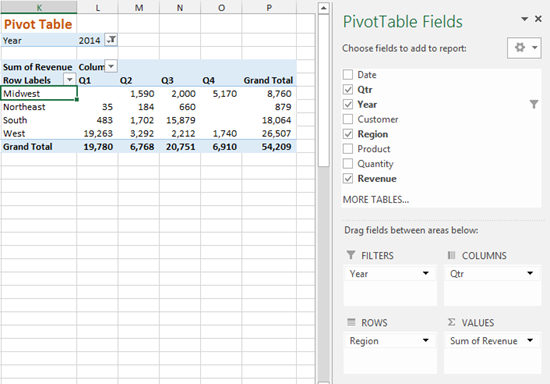
The pivot table contains **four areas** that you can drag the fields into to create a report.

1. Filters area
2. Columns area
3. Rows area
4. Values area

The following diagram shows where each area of the pivot table is located on the report.



We are going to create this quarterly sales report by region as an example shown below.



Such report can be created as follows:

**Rows area:** The Rows area of the pivot table is where we typically start when building the outline of our report. When you drag a field (dimension attribute or hierarchy in cube) into the Rows area of the pivot table, all the unique values in that field will be displayed in the first column of the pivot.

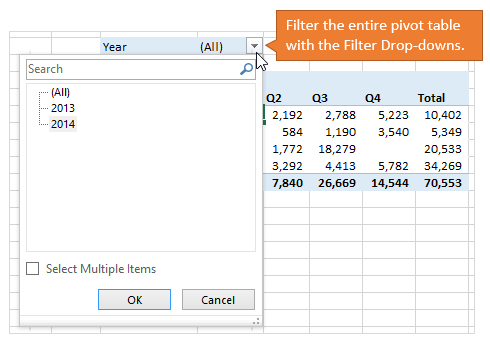
When the Region field is placed in the rows area, each region is only listed one time in the first column of the pivot table.

**Values area:** The Values area displays the data (values) that we want to summarize in our pivot table report. Usually contains the measures of the cube. Just drag and drop any desired cube measure in this area to create a report.

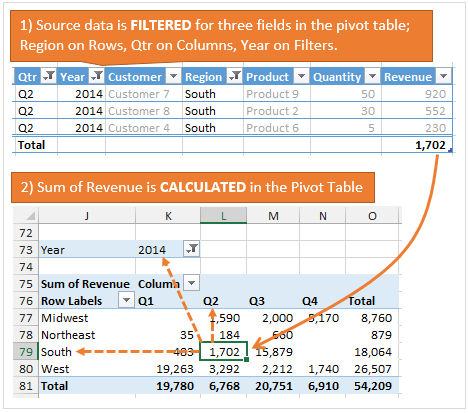
**Columns area:** The Columns area works just like the Rows area.  It lists the unique values of any field (another dimension attribute or hierarchy w.r.t which you want to analyze the data in the **Rows area**) in the pivot table.  The only difference is that it lists the values across the top row of the pivot table.

**Filters area:** The Filters area applies a filter to the entire pivot table.

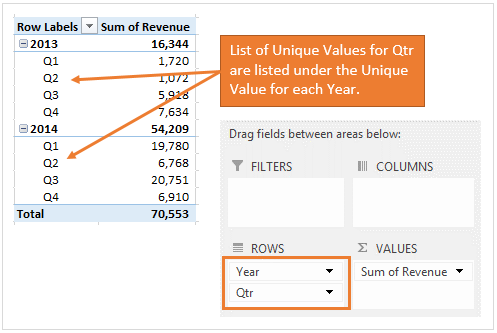
In the example the Year field is placed in the Filters area.  This does NOT make any immediate changes to the pivot table.  Instead, it gives the option to apply a filter to the entire pivot table.



The fields in the Filters area contain drop-down menus that allow you to apply a filter to the entire pivot table. When 2014 is selected from the Year filter, the entire pivot table is also filtered for only the rows in the source data that contain “2014” in the Year column.



**Tip:** For multiple fields in any area, the pivot table works in the same basic way level wise as shown below.



Apart from querying the data in Excel Pivot table as explained above, the data can also be queried directly from the SQL Analysis Server using MDX (multi-dimensional expressions) query for multidimensional modeling or DAX (data analysis expressions) query for tabular modeling format. The MDX format for querying multidimensional cube is as follows:

**MDX:** Multidimensional Expressions (MDX) is a query language for online analytical processing (OLAP) using a database management system. Much like SQL, it is a query language for OLAP cubes. It provides a specialized syntax for querying and manipulating the multidimensional data stored in OLAP cubes. While it is possible to translate some of these into traditional SQL, it would frequently require the synthesis of clumsy and complex SQL expressions even for very simple MDX expressions. MDX has been embraced by a wide majority of OLAP vendors and has become the standard for OLAP systems.

The basic syntax of an MDX query is as follows:

**SELECT {measures} ON COLUMNS , {Dimension attribute} ON ROWS**

**FROM [insert\_your\_cube\_name\_here]**

**WHERE --slicer**

Where

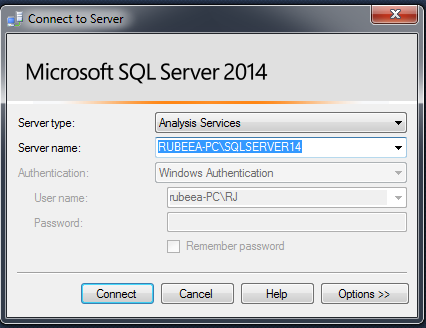
1. **SELECT** clause determines the query axes of an MDX SELECT statement. This contains the number of axes that you want the result set to contain (**ON COLUMNS**). You can specify up to 128 axes in an MDX query specified via the **measures** option above. Measures are separated by a comma just like columns are separated by a comma in an SQL statement.

SELECT also contains the set of members or tuples (**ON ROWS**) to include on each axis of the MDX query specified via the **Dimension attribute** option above. The dimensional attributes are separated by an asterisk (\*).

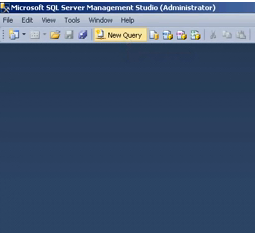
1. **FROM** clause determines which cube will be queried.
2. An optional **WHERE** clause determines which members or tuples to use on the slicer axis to restrict the data returned.

To execute an MDX query on your deployed cube, follow the steps below:

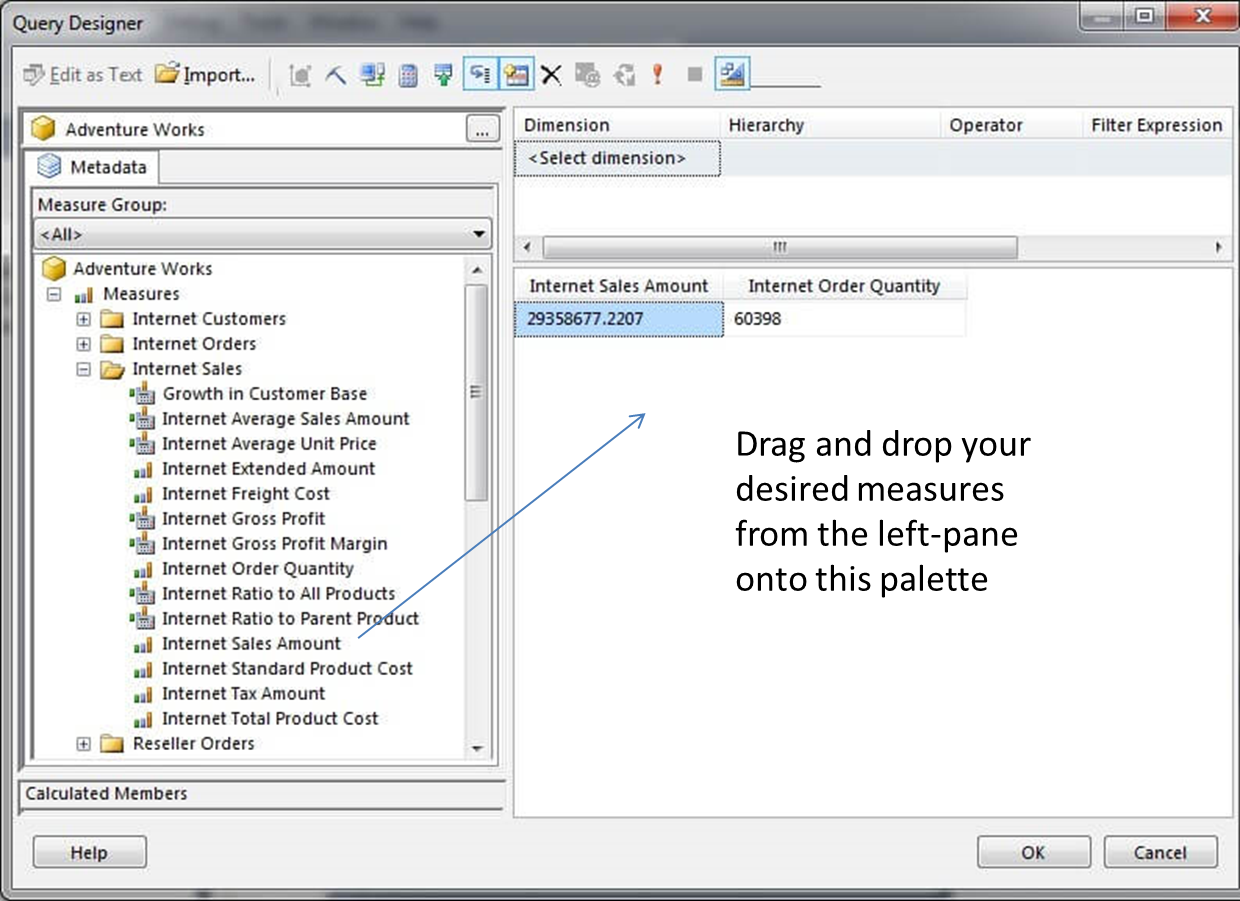
1. Open SQL Server management studio and connect to the Analysis server using either “Windows Authentication” mode or “SQL SERVER Administrator” mode as per your feasibility.



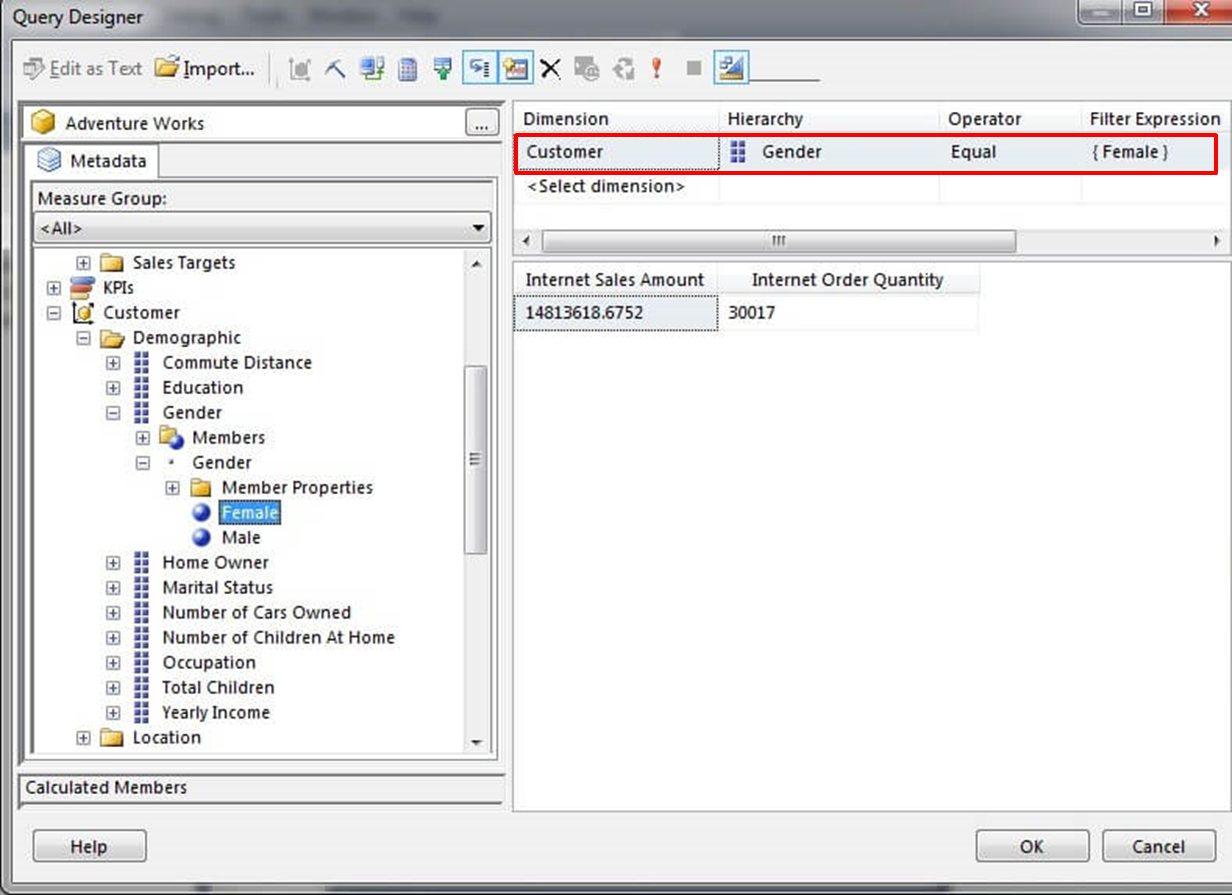
1. Once logged in, expand the Databases folder (by clicking the **+** sign), expand your “**AdventureWorksDW2014**” database and then expand the “**Cubes**” option within it and confirm your deployed cube is present or not.
2. After this, click on the **“New Query”** option in the taskbar to open the query builder for your deployed cube.



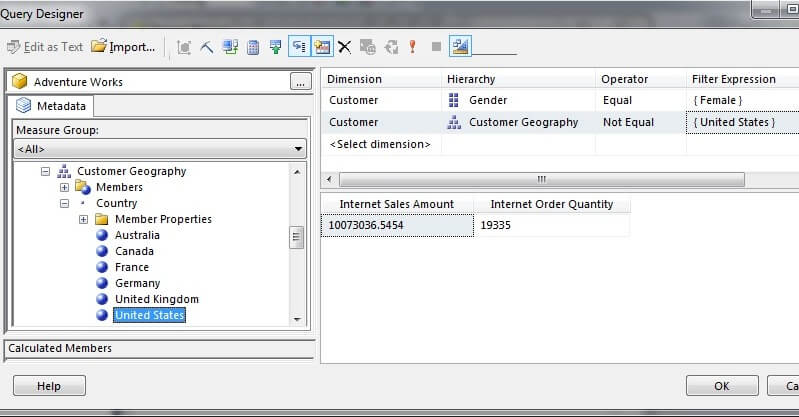
1. You can either write an MDX query according to the syntax above inserting appropriate values according to your requirements or use the drag-and-drop functionality in the Query Designer, and then switch to the text editor to see the auto-generated MDX query.
2. Using the drag and drop functionality, drag the **measures Internet Sales Amount** and **Internet Order Quantity** from the Measures section onto the palette. (This portion specifies the **{measures}** section of the MDX query).



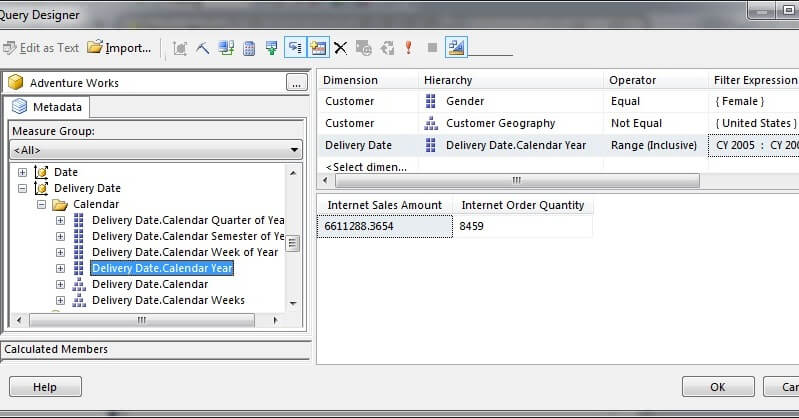
1. Secondly, choose the **Gender** attribute of the **Customer** dimension and filter where the gender equals **Female**. Notice how the values for Internet Sales Amount and Internet Order Quantity have been reduced.



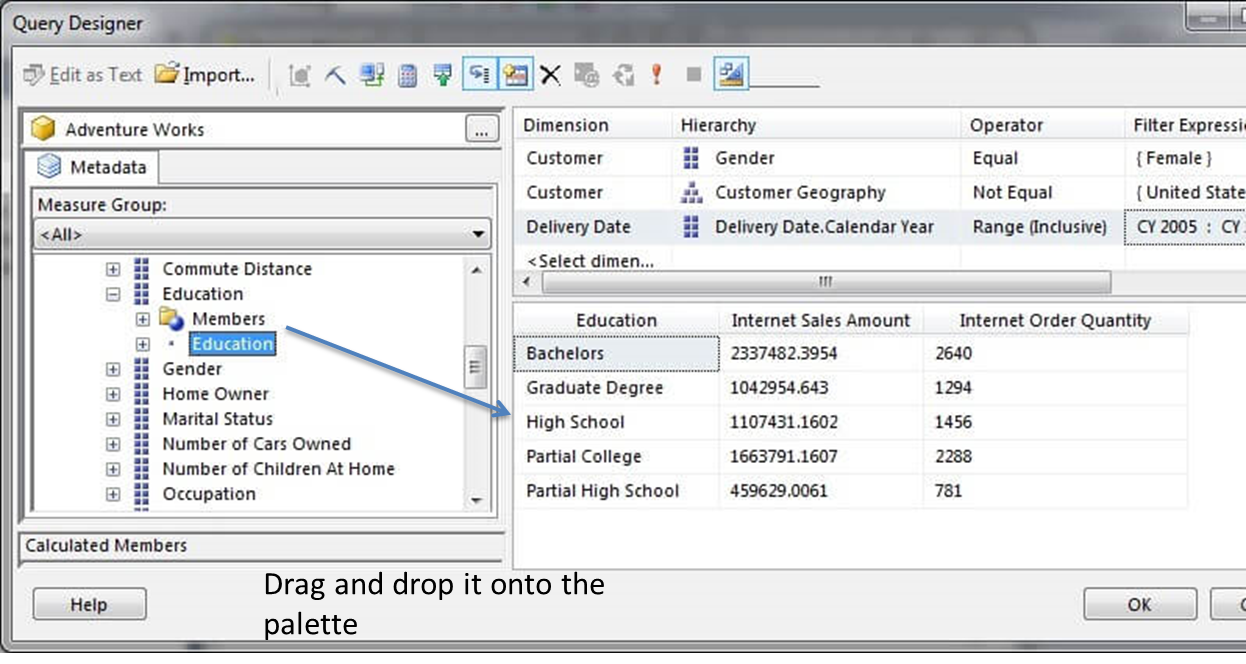
1. Similarly, choose the **Country** attribute of the **Geography** dimension and filter where the country does not equal the United States.



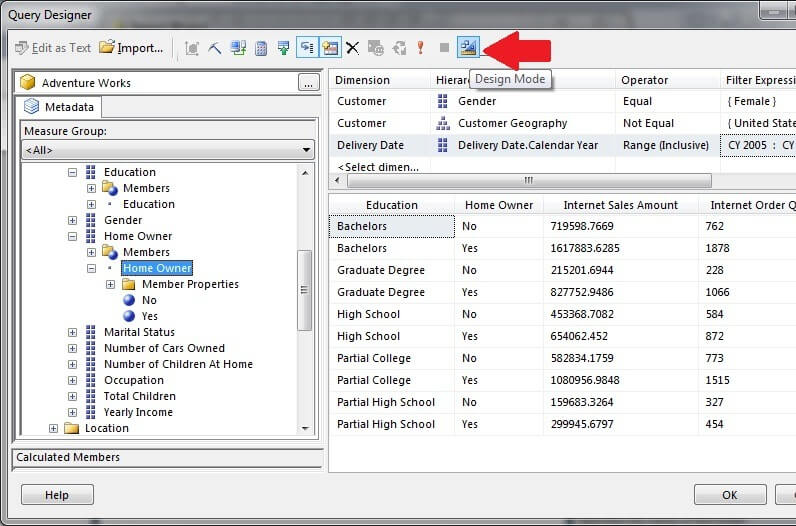
1. Then, choose the **Calendar Year** attribute of the **Delivery Date** role-playing dimension and filter where the range is the calendar years 2005 through 2007 inclusive.



1. Next, drag the **Education** attribute of the **Customer** dimension onto the palette to slice the measures by the education level.

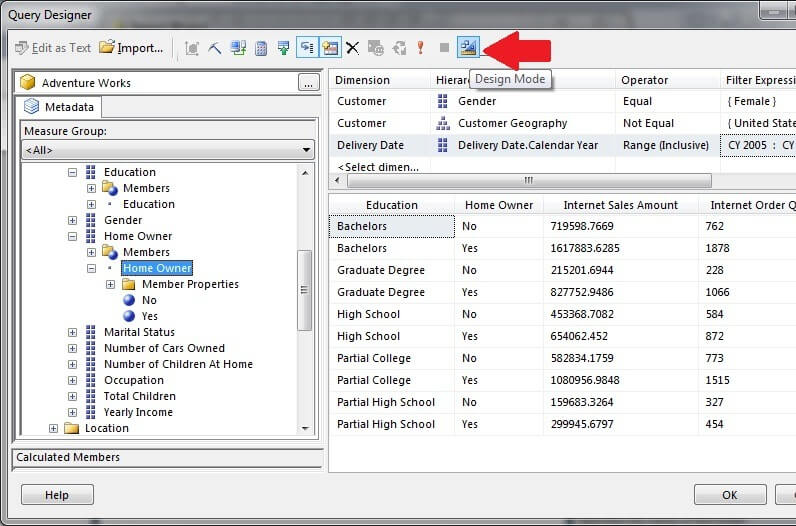


1. Similarly, drag the **Home Owner** attribute of the **Customer** dimension onto the palette to slice the measures by the education level and the home owner attributes.

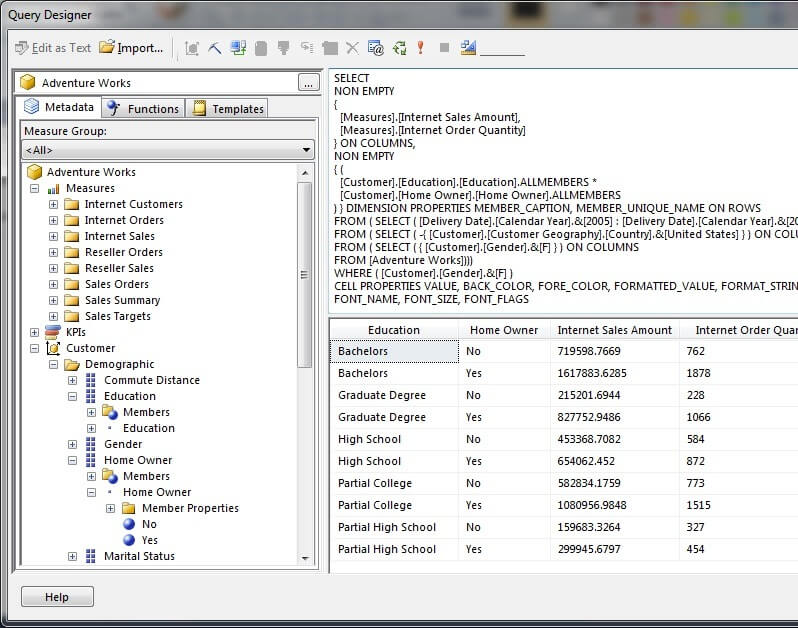


(Steps 9 and 10 specify the **{Dimensional attribute}** section of the MDX query while Steps 6,7,8 are more like slicers to be placed in WHERE clause of the query but since WHERE clause contains only one slicer, the query builder has formatted the MDX using a sort of sub-query options for multiple slicers or filters).

1. Finally, click on the Design Mode icon indicated by the arrow in the image below. This will display the MDX query in a textbox for editing.



Please note that the query in the textbox is not formatted for easy viewing. With a bit of formatting, the query can be viewed more easily.



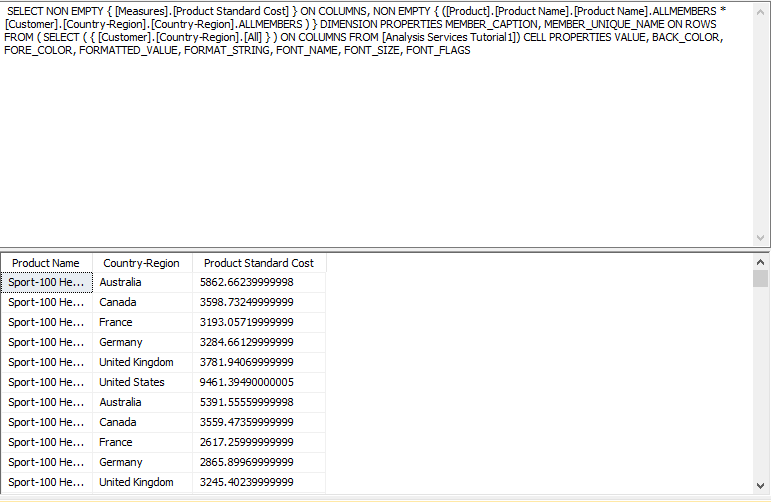
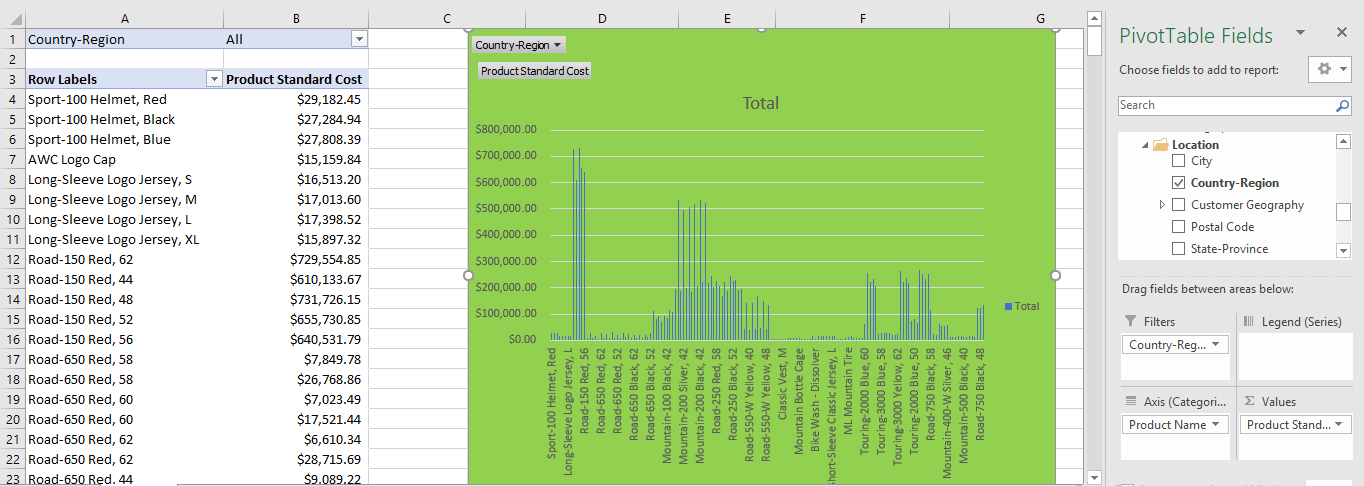
|  |
| --- |
| Lab Tasks |
| Submission Date: |

Generate reports using both MDX and Excel Pivot table for the following scenarios and Generate pivot charts for any 3 scenarios mentioned below to better visualize the data:

1. Display the standard cost of all the products along with their categories according to the regions in which they were sold. Which region is the most and the least expensive according to the result retrieved?

**Answer:**

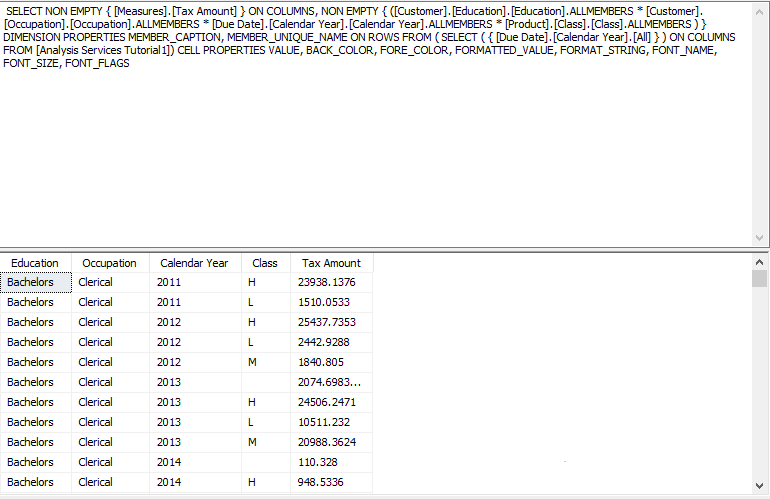
According to the report United States is the most expensive region and Franceis the least expensive region**.**

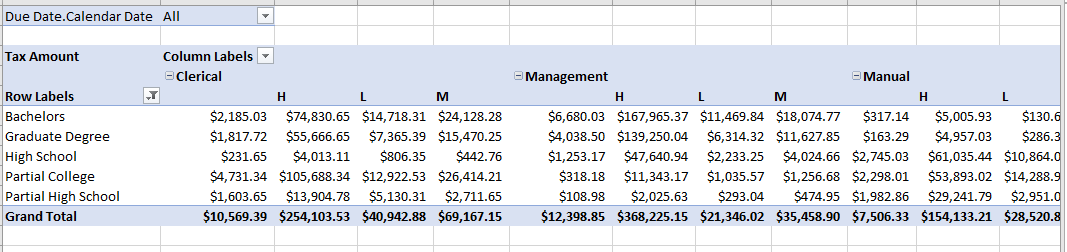


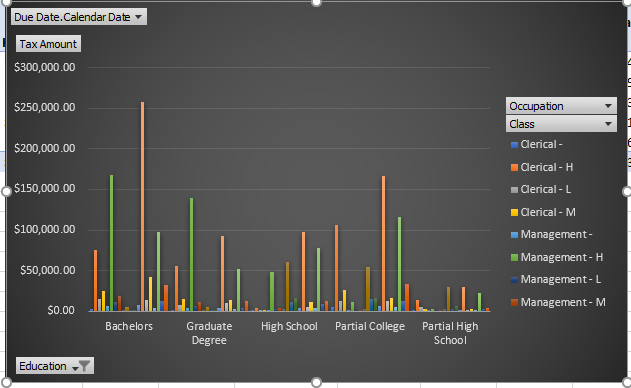
1. Display the Tax amount for customers according to their educations and occupations over calendar year. Do these factors impact the tax amount paid by a customer? Which class of customers pays the highest tax amount?

**Answer:**

Yes, according to the report, the above-mentioned factors impact the tax amount paid by a customer. High class of customers pays the highest tax amount.



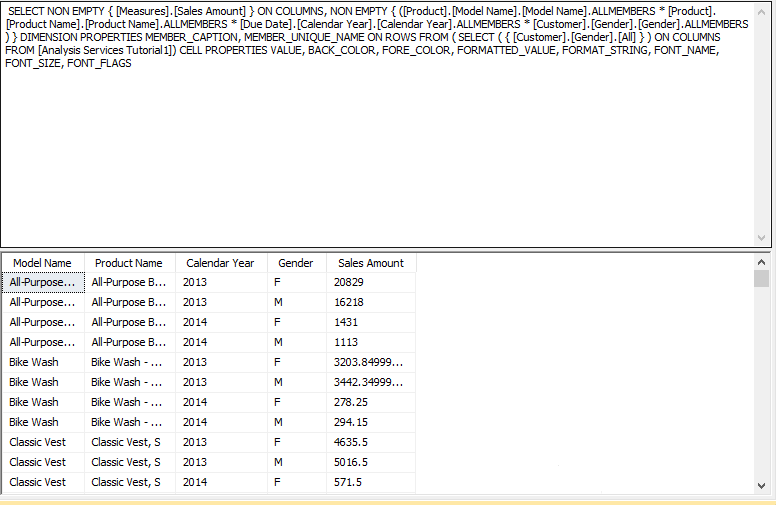


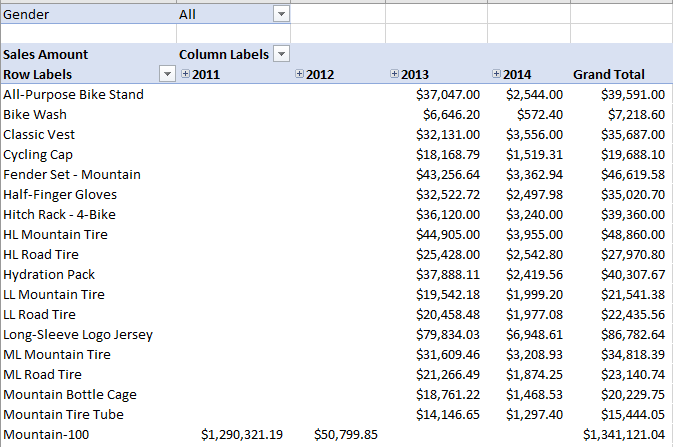


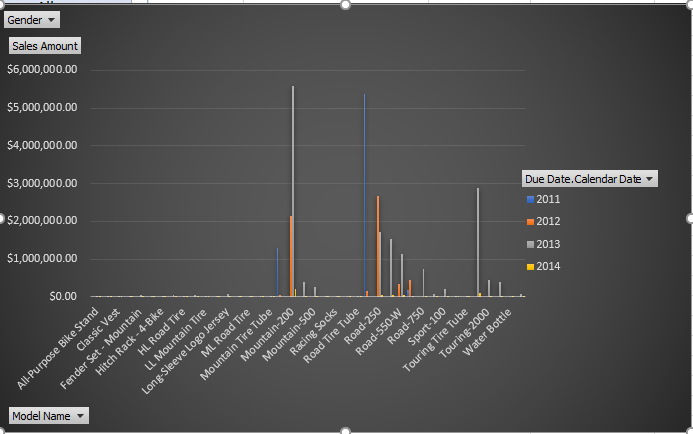
1. Display the sales amount for all products (use product hierarchy) purchased according to the calendar year and filter the results according to the customers’ genders. What type of customers bought the most Biking products?

**Answer:**

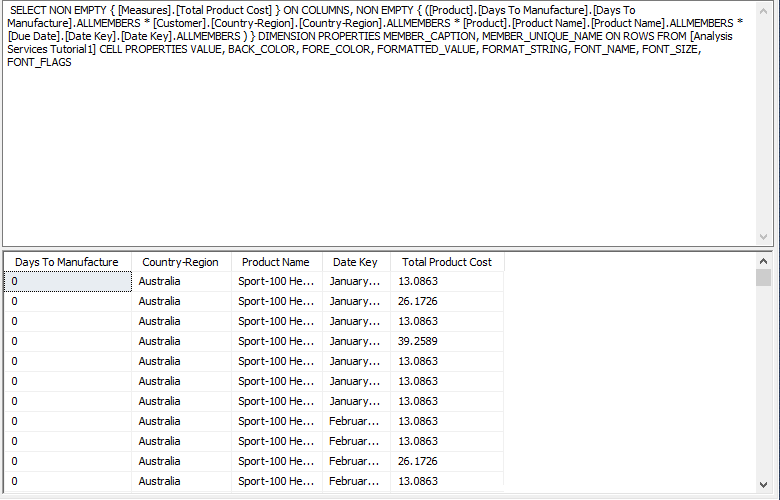
According to the report male customers bought the most Biking products.

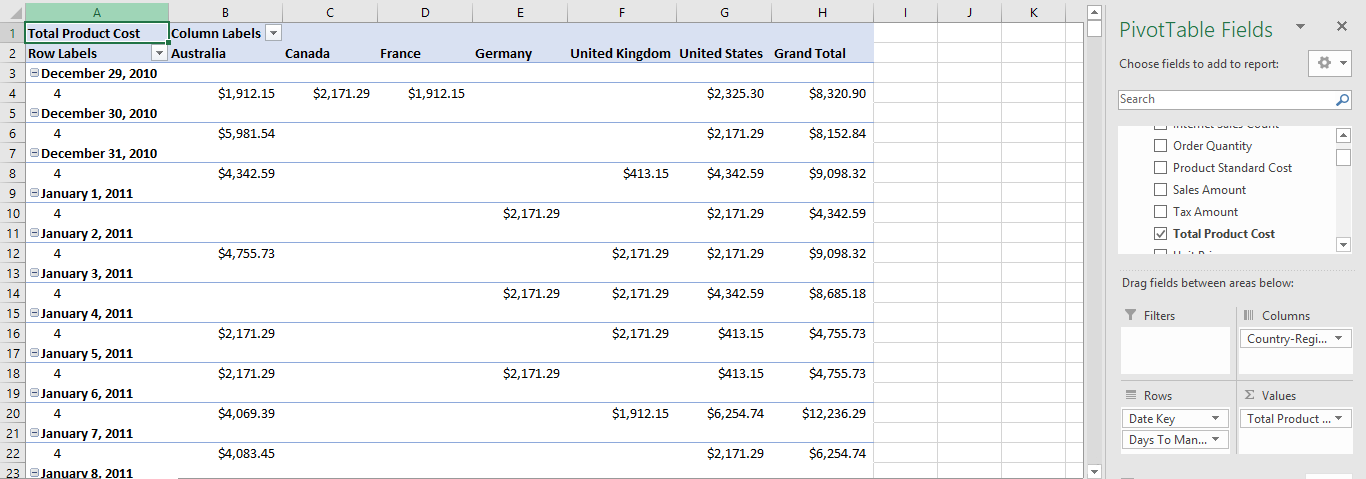






1. Display the total days required to manufacture the products (use product hierarchy) according to region and time (use date hierarchy).





1. Display the order quantity for all the products ordered by customers according to their marital statuses and genders. Filter the results for customers having yearly income greater than or equal to 50,000.

