

**EECS 495**  
**Introduction to Database Systems**  
**Fall 2018**  
**Instructor: Mas-ud Hussain**  
**Project No. 4**  
**Due: Friday, December 14, 2018**

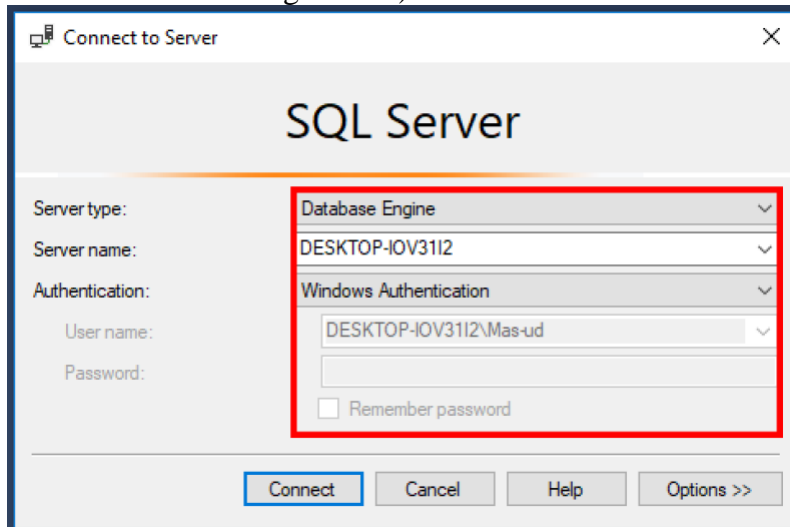
**Guidelines:**

- **This project is supposed to be done in groups of 2 or at most 3 people.**
- The project is due by the midnight on the day mentioned on the top. Please submit project electronically in Canvas by uploading your files. If you have more than 1 file to submit, please zip the files into a single file and upload the zip file. *Only one submission per group is required.*
- **Late policy:** As the due date is already very close to the end of the quarter – there is no late submission allowed for this project. **We won't grade the projects submitted late.**

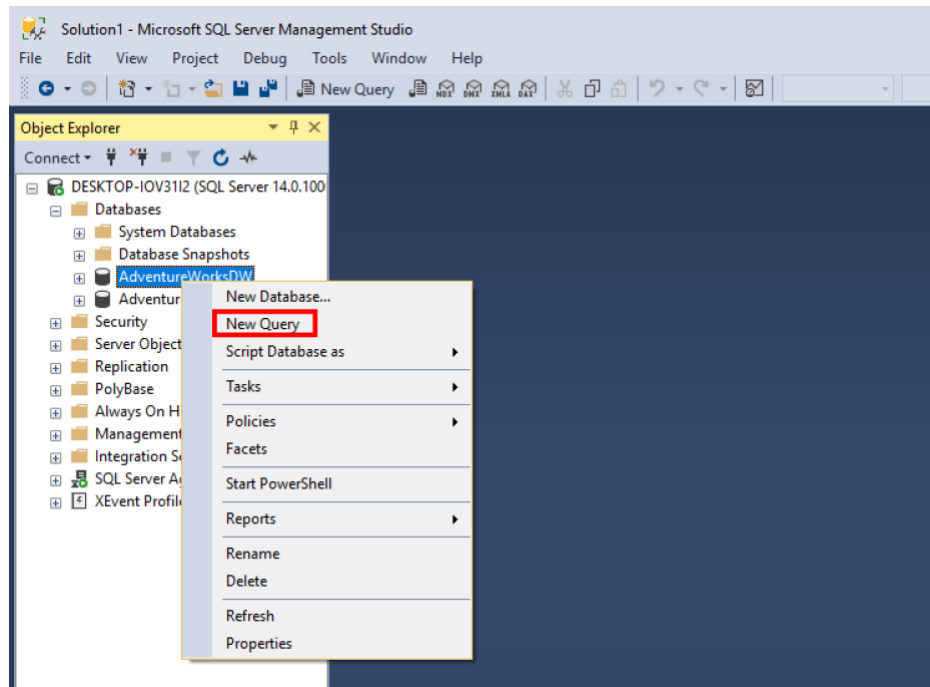
**Browsing Cube and Running Queries:**

As a first step, go through the tutorial (s) attached with the project (do all the steps necessary). Look carefully what is shown in the last class for MDX demo. Once you are done practicing, download the “Adventure Works Enterprise DB Project.zip” file attached with the project. After that, unzip the file and within, you will find a .sln file (the prepared multi-dimensional model for the project). Open the .sln file in Visual Studio, or SQL Server Data Tools. After that, you right click on the project and run the process (as shown in the tutorial attached). After completing the process, you will see “process successful” and “deployed successful” notifications.

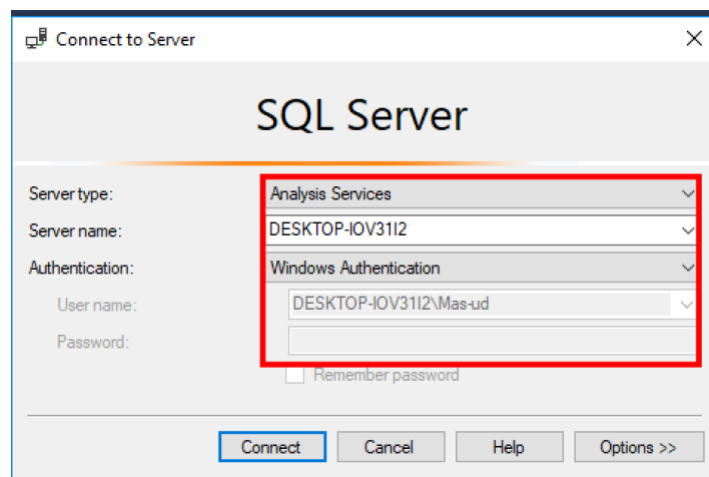
Plain SQL query on the original Relational Database: (Use SQL Server Management Studio and connect to the database as Database Engine first)



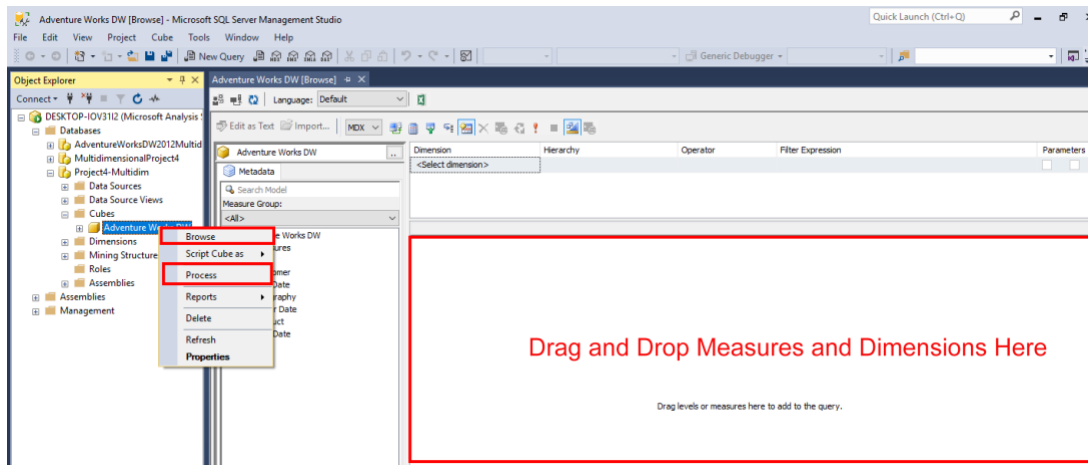
Then, you can select the “AdventureWorksDW” database from the databases and run new queries accordingly:



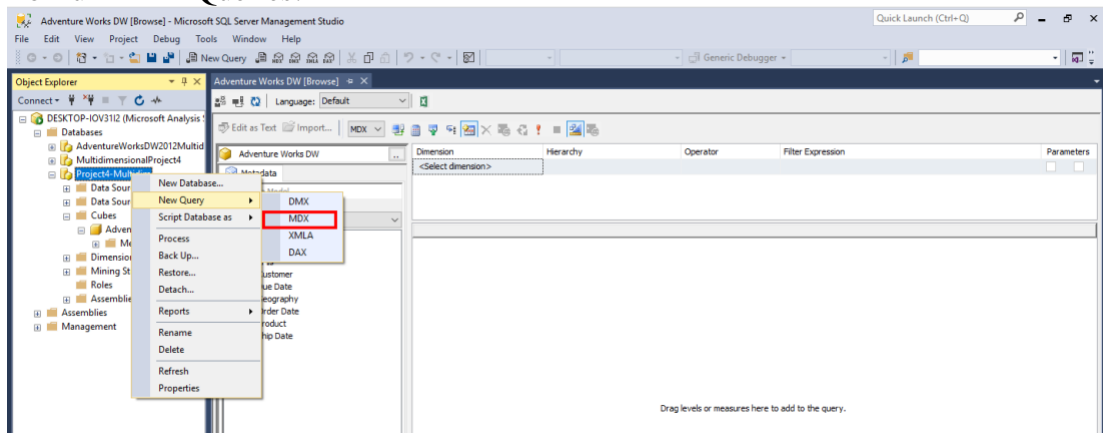
Open the SQL Server Management Studio (SSMS) again and connect to the “Analysis Service” to browse and process cubes, and run MDX queries.



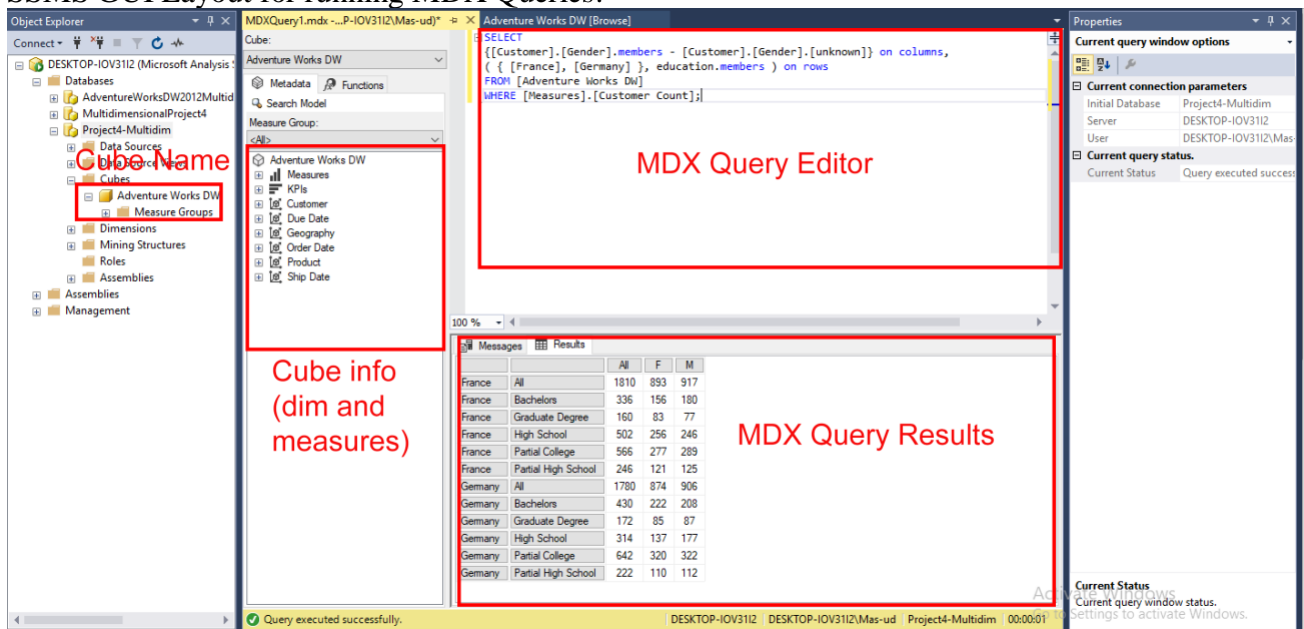
To browse and process cubes:



## To Run MDX Queries:



## SSMS GUI Layout for running MDX Queries:



## Exercises (8 x 12.5)

1. A) Write an SQL query (run against the SQLServer AdventureWorks database) that returns the aggregates needed for the following cross tabulation:

		All	F	M
France	All	1810	893	917
France	Bachelors	336	156	180
France	Graduate Degree	160	83	77
France	High School	502	256	246
France	Partial College	566	277	289
France	Partial High School	246	121	125
Germany	All	1780	874	906
Germany	Bachelors	430	222	208
Germany	Graduate Degree	172	85	87
Germany	High School	314	137	177
Germany	Partial College	642	320	322
Germany	Partial High School	222	110	112

Use group by cube/rollup as you feel fit. The table was generated with the following MDX expression: (Your given Cube name is “Adventure Works DW”)

**SELECT**

{[Customer].[Gender].members - [Customer].[Gender].[unknown]} on columns,  
( { [France], [Germany] }, education.members ) on rows

**FROM** [Adventure Works DW]

**WHERE** [Measures].[Customer Count];

- B) Create now the cross-tabulation using the cube browser. (by dragging and dropping, and then setting dimension filters). The final output should look like this:

Country	Education	Gender	Customer Count
France	Bachelors	F	156
France	Bachelors	M	180
France	Graduat...	F	83
France	Graduat...	M	77
France	High School	F	256
France	High School	M	246
France	Partial C...	F	277
France	Partial C...	M	289
France	Partial Hi...	F	121
France	Partial Hi...	M	125
Germany	Bachelors	F	222
Germany	Bachelors	M	208
Germany	Graduat...	F	85
Germany	Graduat...	M	87
Germany	High School	F	137
Germany	High School	M	177
Germany	Partial C...	F	320
Germany	Partial C...	M	322
Germany	Partial Hi...	F	110
Germany	Partial Hi...	M	112

2. Rewrite the MDX-query of question 1, such that the answer becomes:

		F	M
France	All	893	917
France	Bachelors	156	180
France	Graduate Degree	83	77
France	High School	256	246
France	Partial College	277	289
France	Partial High School	121	125
Germany	All	874	906
Germany	Bachelors	222	208
Germany	Graduate Degree	85	87
Germany	High School	137	177
Germany	Partial College	320	322
Germany	Partial High School	110	112

3. Write a MDX query to create cross-tabulation between countries and education level that only includes count of Males (i.e., Gender values M). The output should be in following

format:

	All	Australia	Canada	France	Germany	United Kingdom
All	9351	1814	804	917	906	979
Bachelors	2728	843	187	180	208	292
Graduate Degree	1578	152	171	77	87	131
High School	1699	302	160	246	177	187

4. Create a measure that counts the percentage of males in the customer count. Use this measure to make an overview of the percentage of males in the customer counts per country and year. The output should be in following format:

	2005	2006	2007	2008	2009	2010
Australia	50.52%	50.52%	50.52%	50.52%	50.52%	50.52%
Canada	51.18%	51.18%	51.18%	51.18%	51.18%	51.18%
France	50.66%	50.66%	50.66%	50.66%	50.66%	50.66%
Germany	50.90%	50.90%	50.90%	50.90%	50.90%	50.90%

5. Generate a list of the internet “sales amount” in all cities of France and Germany. Omit the empty cells. The output should be in following format:

		Sales Amount
France	Bobigny	90204.4535000001
France	Boulogne-Billancourt	14289.2439
France	Boulogne-sur-Mer	11342.9225
France	Cergy	46755.9003
France	Chatou	89830.1988000001
France	Colombes	90268.5149000002
France	Colomiers	54641.7231999999

6. Create a set “North America”, which includes USA and Canada. Then, use the set to create view of how the “Order Quantity” number varies by Education for North American customer. The output should be in the following format:

	Bachelors	Graduate Degree	High School	Partial College
United States	5795	5190	3169	6171
Canada	1701	1872	1328	2011

7. Create a set storing the “model name” of top 5 products by number of orders placed for them. Then, use the defined set to list sales amount of those products for each month. The output should be in the following format:

	Sport-100	Water Bottle	Mountain-200	Patch kit
April	15640.5299999999	1531.93	484272.0069999998	602.2700000000
August	19209.51	2020.95	774682.8889999998	641.2
December	23583.26000000002	2160.67	995906.0383999997	638.91
February	14590.8299999999	1367.26	376678.0075999999	533.5700000000
January	14170.9499999999	1377.24	438780.9535999999	538.1500000000
July	17984.86	1771.45	673309.4005999998	613.72
June	21063.98000000001	1971.05	763832.5777999998	638.91
March	16725.2199999999	1591.81	472974.0203999999	545.0200000000

8. Show quarterly sales by customer countries compared to those of the previous year (same quarter). You have to define a new measure for this query using Parallel Period concept. Your output should be in the following format:

		Sales Amount	Prev Year
Australia	4	20909.78	(null)
Australia	1	588674.9166000001	(null)
Australia	2	699462.5012000001	(null)
Australia	3	631289.0758000001	(null)
Australia	4	644305.7557000001	20909.78
Australia	1	396321.817	588674.92
Australia	2	485812.0225	699462.50
Australia	3	606714.2643000001	631289.08

### **Hints/Tips:**

**For all the queries above, you have to avoid showing “Unknown” column in your results, using the method shown in the example MDX query for question 1.**



To learn about how to install and set-up MS SQL Server appropriately, see the attached Documents with the project files. There is detailed information on how to install, setup, create cubes, dimensions, and run MDX queries in there.



To learn more about the syntax and rules of MDX queries, see the attached document: **MDXTutorial.pdf**.



More information on MSDN website (MDX queries):<https://msdn.microsoft.com/en-us/library/ms145514.aspx>.



More information on MSDN website (Analysis Services):<https://msdn.microsoft.com/en-us/library/hh231701.aspx>.

## **Submission Guidelines:**

Attach screen shots with the results of your queries and upload your solution on Canvas.

## **Collaboration**

Lastly, and most importantly, a note on what is allowed and what is not allowed in terms of collaboration. It is OK to speak in generalities about how one might approach these problems with others. In other words, you can bounce ideas off of one another. However, it is NOT OK TO DISCUSS SQL ANSWERS & STATEMENTS AT ALL, as it relates to these problems, with anyone else in class. It is NOT OK to look at anyone else's answers, or to specifically discuss answers to the problems with anyone else. In short, two sets of answers to the given problems that are turned in with the same SQL statements will be viewed with extreme suspicion (and will be penalized accordingly if cheating is proved).