KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Politechnika Wrocławska

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

ROZWÓJ POTENCJAŁU I OFERTY DYDAKTYCZNEJ POLITECHNIKI WROCŁAWSKIEJ

Wrocław University of Technology

Computer Engineering

Przemysław Kazienko, Piotr Bródka

DATA WAREHOUSES

Wrocław 2011

Projekt współfinansowany ze środków Unii Europejskiej w ramach
Europejskiego Funduszu Społecznego

Wrocław University of Technology

Computer Engineering

Przemysław Kazienko, Piotr Bródka

DATA WAREHOUSES

Wrocław 2011

Reviewer: Zygmunt Mazur

ISBN 978-83-62098-97-2

Published by PRINTPAP Łódź, www.printpap.pl

Content

Introduction 5

Part I. Microsoft 6

Prepare Your Environment. 6

SQL Server Integration Services and Sample ETL Process 6

SQL Server Analysis Services and Sample Cube 15

SQL Server Reporting Services and Sample Report 23

Additional Examples 33

References and Additional Information 33

Task List 1. SQL Server Integration Services, ETL - Extraction, Transformation, Loading 34

Task List 2. Introduction to SQL Server Analysis Services – build your first cube 36

Task List 3. SQL Server Analysis services – build your own cube 38

Task List 4. Processing and accessing your cube –MDX, Excel, SQL Server Management Studio, SQL Server Reporting Services 40

Part II. SAS 42

References and Additional Information 57

Task List 5. Introduction to SAS and 4GL 57

Task List 6. Analysing data using SAS tools 60

Task List 7. The idea of multidimensional databases – cubes in SAS 63

Task List 8. Metadata, OLAP reports and charts in SAS 66

Part III. Project 68

Task List 9. Choose your Tool with justification and prepare data set for analysis 68

Task List 10. ETL process – transfer data to database, derived variables 70

Task List 11. Cubes, measures, dimensions..... 72

Task List 12. Reports and Charts – selection and adjustment, cubes efficiency issues..... 73

Task List 13. Processing and accessing your cube 75

Introduction

The general goal of the laboratory classes is to provide suitable knowledge about various tools used for data warehousing as well as their applications. The classes consist of a series of task lists (assignments) solved by students themselves and reported to the supervisor. Two different software tools are used during the lab: SQL Server (Analysis Services) by Microsoft and SAS Institute tools.

There are three main parts of the course:

1. Basic features of data warehouse management by means of SQL Server, four first task lists.

2. Data warehouses in SAS tools and basics of SAS 4GL language, task lists no. 5-8.

3. A project, task lists no. 9-13.

The task lists usually contain some optional tasks, which, if solved, increase the individual grade for the task list. However, it is not necessary to solve these tasks to pass the course.

- Basic literature:

1. Inmon W.H.: Building the Data Warehouse, Wiley, 2005
2. Poe V., Klauer P., Brobst S.: Building A Data Warehouse for Decision Support, Prentice Hall PTR, 1997.
3. Giovinazzo W.: Object-Oriented Data Warehouse Design: Building a Star Schema. Prentice Hall, 2000.
4. Barquin R.C., Edelstein H.A. (eds): Planning and Designing the Data Warehouse. Prentice Hall, 1997.
5. Bischoff J., Alexander T.: Data Warehouse: Practical Advice from the Experts. Prentice Hall, 1997.

- Additional literature:

1. Mundy J.: The Microsoft Data Warehouse Toolkit: With SQL Server 2005 and the Microsoft Business Intelligence Toolset, Wiley, 2006
2. Wang J.: Encyclopedia of Data Warehousing and Mining, Idea Group Publishing, 2005
3. Todman C.: Designing a Data Warehouse: Supporting Customer Relationship Management, Prentice Hall PTR, 2000

Part I. Microsoft

First four task lists will be solved using tools from SQL Server by Microsoft, in particular *SQL Server Analysis Services* and *Integration Services*.

Prepare Your Environment.

1. Install *SQL Server 2008 R2* (or a newer version). If you have any problems during installation you can find help on http://technet.microsoft.com/en-us/library/ms143219.aspx. During installation of *SQL Server 2008 R2* select the following components on the *Components to install page* of the *Installation Wizard*:

   a) *Integration Services* - http://msdn.microsoft.com/en-us/library/bb522532.aspx,

   b) *Analysis Services*. - http://msdn.microsoft.com/en-us/library/bb522612.aspx,

   c) *Reporting Services* - http://msdn.microsoft.com/en-us/library/bb522676.aspx

2. Install *Visual Studio 2010* (or a newer version). If you have any problems during installation you can find help on http://msdn.microsoft.com/en-us/library/e2h7fzkw.aspx

SQL Server Integration Services and Sample ETL Process

"Microsoft Integration Services is a platform for building enterprise-level data integration and data transformations solutions. You use Integration Services to solve complex business problems by copying or downloading files, sending e-mail messages in response to events, updating data warehouses, cleaning and mining data, and managing SQL Server objects and data. The packages can work alone or in concert with other packages to address complex business needs. Integration Services can extract and transform data from a wide variety of sources such as

XML data files, flat files, and relational data sources, and then load the data into one or more destinations.

Integration Services includes a rich set of built-in tasks and transformations; tools for constructing packages; and the Integration Services service for running and managing packages. You can use the graphical Integration Services tools to create solutions without writing a single line of code; or you can program the extensive Integration Services object model to create packages programmatically and code custom tasks and other package objects."[1]

To create a sample ETL process:

1. Start *Microsoft Visual Studio 2010* or *SQL Server Business Intelligence Development Studio* (this is a free version of *Visual Studio* which allows to create *BI* processes but it may have limited functionality).

2. *Select File->New->Project*... or press *Ctrl+Shift+N*

3. In *Project types* select *Business Intelligence Project***,** in *Templates* select *Integration Services Project***,** choose *Name*, project *Location* and press *OK*

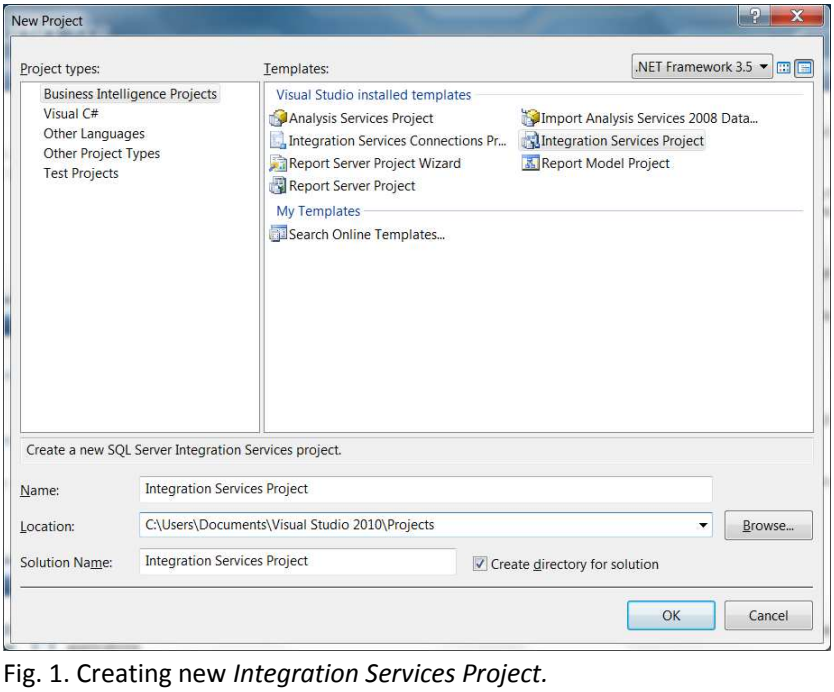


Fig. 1. Creating new *Integration Services Project.*

[1] http://msdn.microsoft.com/en-us/library/ms141026.aspx

4. From the *Control Flow Items* select *Data Flow Task*. Drag and drop it in the *Control Flow* tab

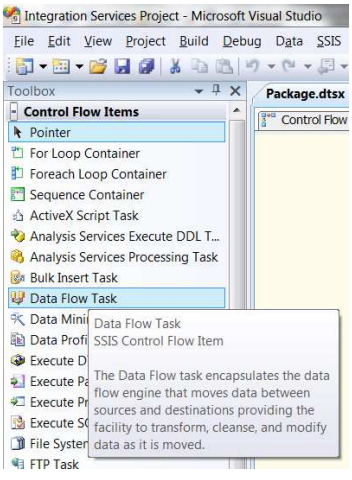


Fig. 2. *Data Flow Task* selection.

5. Double click on newly created *Data Flow Task***.** You should be moved to the *Data Flow* tab.

6. From *Data Flow Sources* select an appropriate data source. During classes in most cases you will be using *Flat File Source* so in the example we will choose it as well.

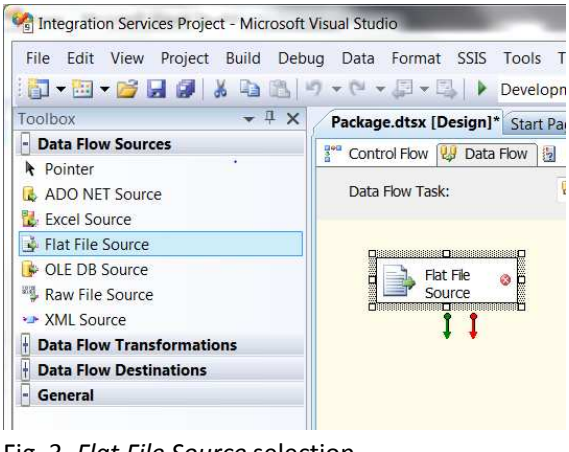


Fig. 3. *Flat File Source* selection.

7. Double click on your newly created *Flat File Source* to open the *Flat File Source Editor*. First we have to create *New Flat File Connection Manager*. To do so click the *New...* button and open *Flat File Connection Manager Editor*. Type the name of your connection, select file location, text qualifier, Headers

row delimiter and number of header rows to skip if you have some headers rows. In this example we will be processing flat file containing *Thurman Social Network*.

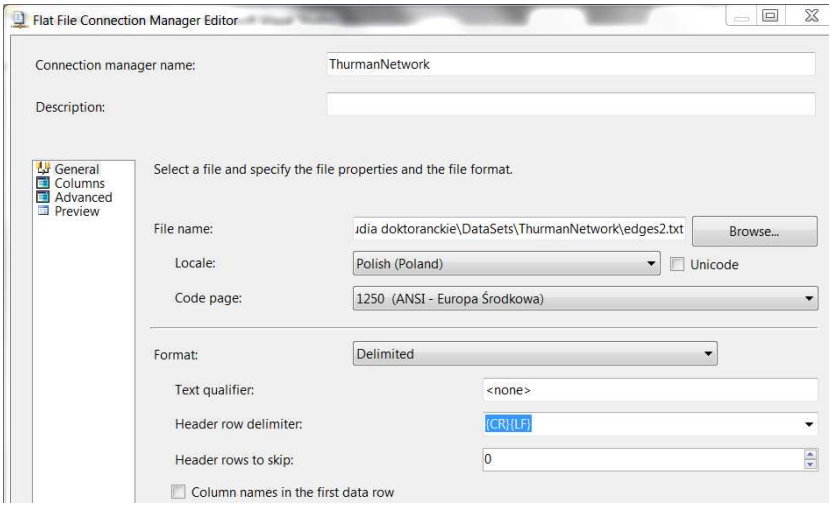


Fig. 4. *Flat File Connection Manager Editor.*

8. Next go to the *Columns* tab and choose Row and Column delimiters. Using *Preview* you can see if you have achieved the desired result.

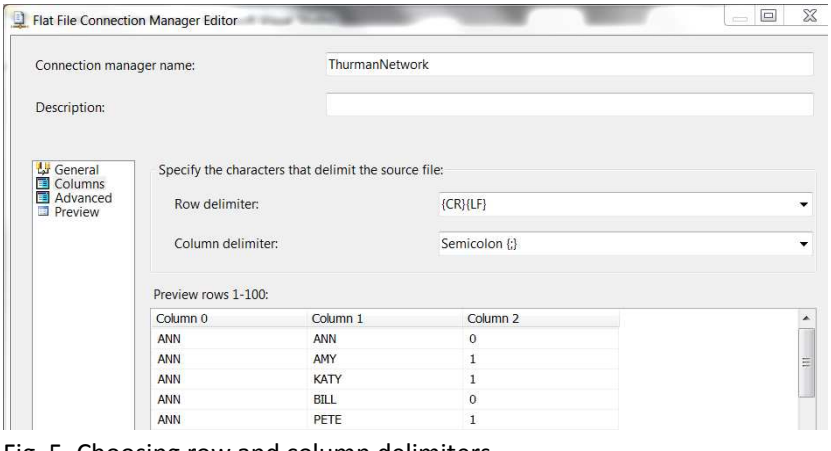


Fig. 5. Choosing row and column delimiters.

9. In the *Advanced* tab you have to name all your columns and set appropriate data types.

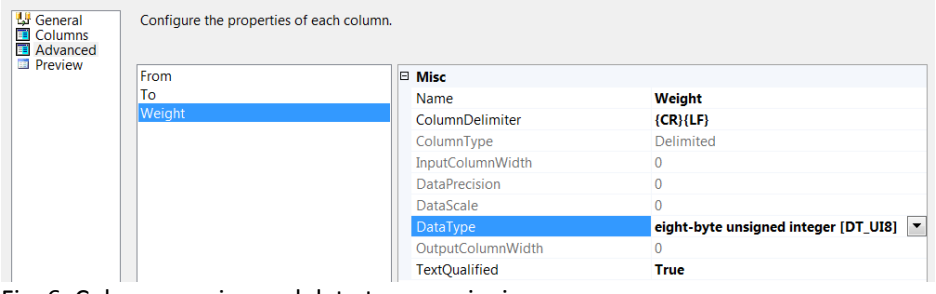


Fig. 6. Column naming and data types assigning.

10. In the *Preview* tab you can check if everything is set correctly. If yes, click the *OK* button.

11. In the *Flat File Source Editor* you can rename columns if you need to and choose columns which you want to extract from your *Flat File*. In this example we are selecting all columns and choosing the *OK* button to complete editing *Flat File Source*.

12. After extraction we can transform our data. We have many built-in transformations. In this example we will try to derive new columns. To do so we drag and drop *Derived Column* transformation and attach it to *Flat File Source* by connecting these two elements with a green arrow.

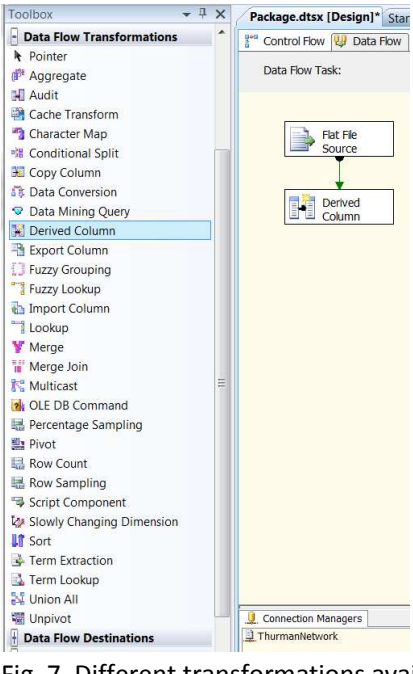


Fig. 7. Different transformations available in *Integration Services*.

13. Double click on the *Derived Column* transformation to open *Derived Column Transformation Editor*. Here we have many built-in Functions which can process our data. We will create 4 new columns. The first one will create an email address from the *From* column, the second one will create an email address from the *To* column, the third column will change our *Weight* and the last new column will be a *Description* column which is a concatenation of all extracted columns. When our new columns are created we can click the *OK* button.

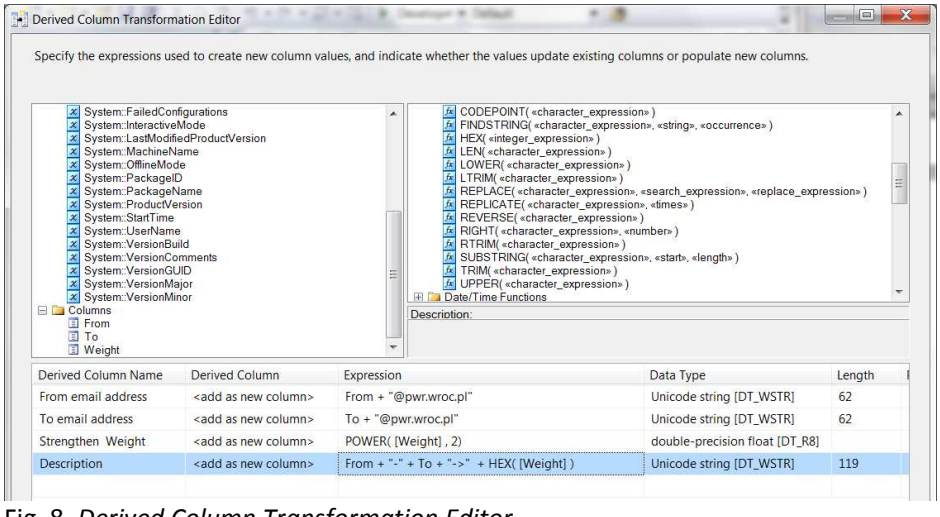


Fig. 8. *Derived Column Transformation Editor*.

14. Now we would like to load our transformed data to a new destination. To do so we have to choose new *Data Flow Destination*. In our example we would like to load transformed data to the *SQL Server 2008*

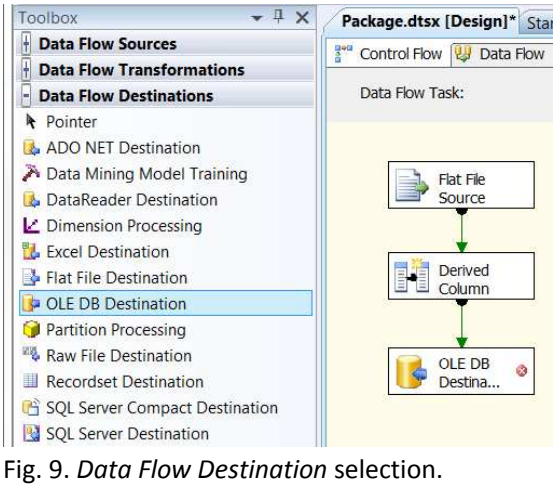


Fig. 9. *Data Flow Destination* selection.

15. Before we can do this we have to prepare the appropriate database and tables in *SQL Server 2008* using *SQL Server Management Studio* (As relational databases are not in the scope of this course we assume that each student know how to do this).

16. Once the appropriate database is created, drag and drop *OLE DB Destination*, connect it to *Derived Colum* and double-click it to open *OLE DB Destination Editor*.

17. First we have to create a new connection to the database. Click *New*... to go to *Configure OLE DB Connection Manager* and once again select *New*... to open *Connection Manager*. *Select SQL Server Native Client* as a provider. Next, provide connection credentials (server name, database name, and user and password in the case of *SQL Server Authentication*), click *Test Connection* to check if everything is ok and, if so, click *OK*.

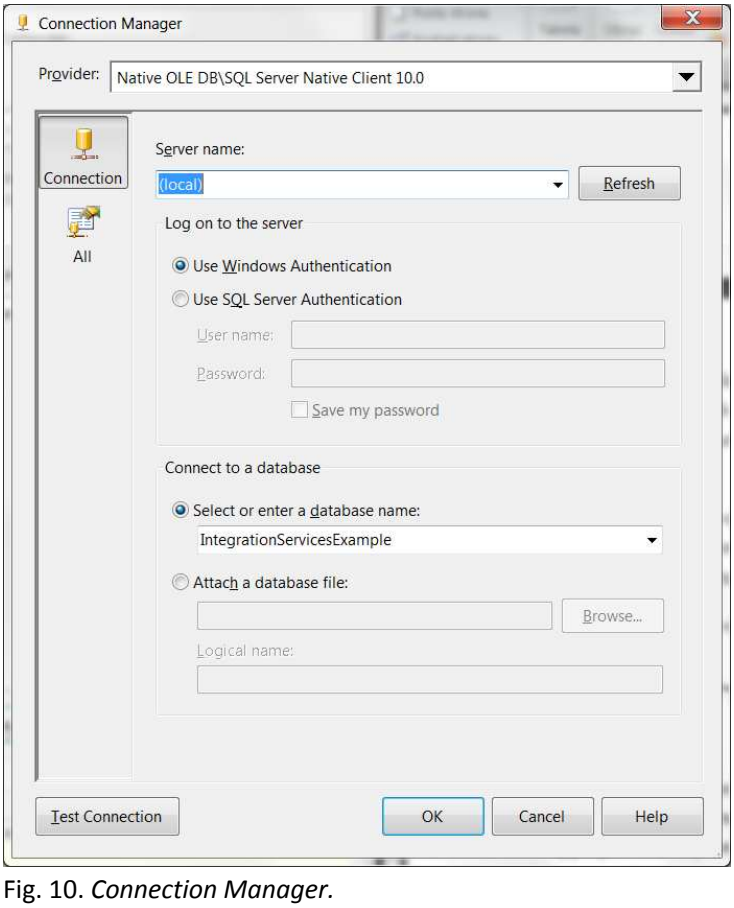


Fig. 10. *Connection Manager.*

18. Select the newly created connection in *Configure OLE DB Connection Manager* and click *OK*.

19. In *OLE DB Destination Editor* you can either select the existing table in the database or create a new table (in this case the application will propose an editable *SQL query* which will create a new table).

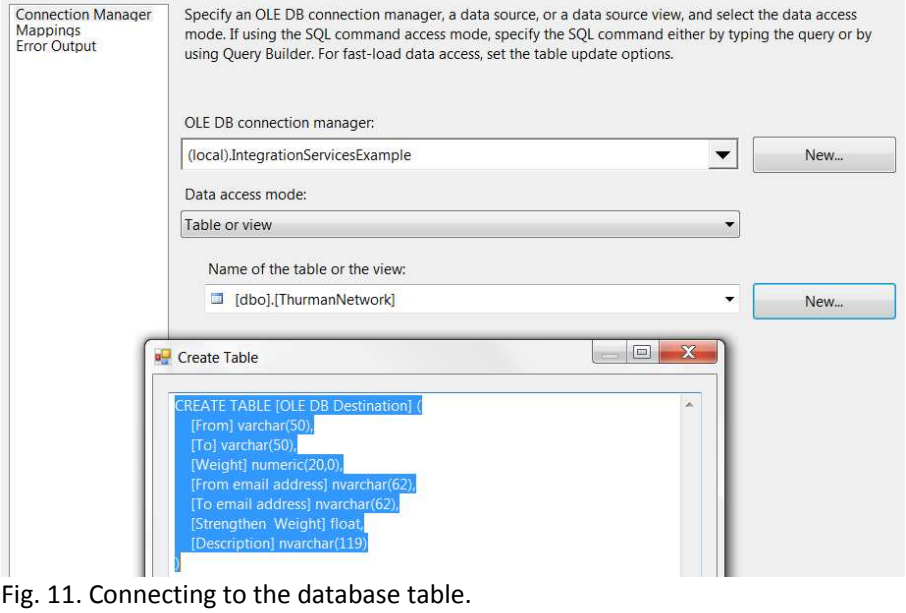


Fig. 11. Connecting to the database table.

20. Next go to the *Mappings* tab and map your transformed columns into database table columns and click *OK*.

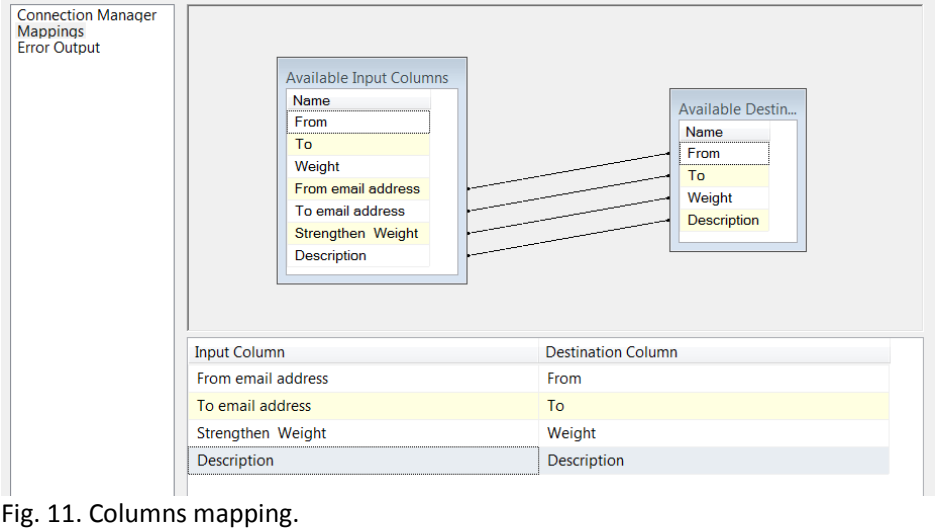


Fig. 11. Columns mapping.

21. The final step is to start our *ETL Process*. To do so press *F5*. If you have done everything correctly all parts of our process change their colour to green; if something went wrong the colour will change to red and the whole process will stop.

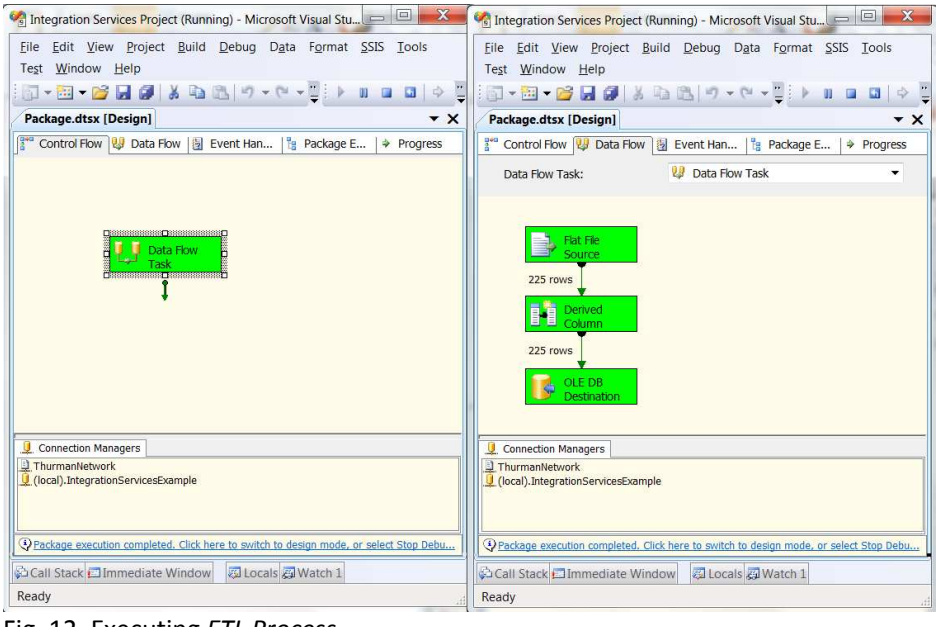


Fig. 12. Executing *ETL Process*.

SQL Server Analysis Services and Sample Cube

"Microsoft SQL Server Analysis Services—Multidimensional Data allows you to design, create, and manage multidimensional structures that contain detail and aggregated data from multiple data sources, such as relational databases, in a single unified logical model supported by built-in calculations.

Analysis Services—Multidimensional Data provides fast, intuitive, top-down analysis of large quantities of data built on this unified data model, which can be delivered to users in multiple languages and currencies.

Analysis Services—Multidimensional Data works with data warehouses, data marts, production databases and operational data stores, supporting analysis of both historical and real time data."[2]

To create a sample cube:

1. Start *Microsoft Visual Studio 2010* or *SQL Server Business Intelligence Development Studio* (this is a free version of *Visual Studio* which allows to create BI processes but it may have limited functionality).
2. *Select File->New->Project*... or press *Ctrl+Shift+N*
3. In *Project types* select *Business Intelligence Project*, in *Templates* select *Analysis Services Project*, choose *Name*, project Location and press *OK*

[2] http://msdn.microsoft.com/en-us/library/bb522607.aspx

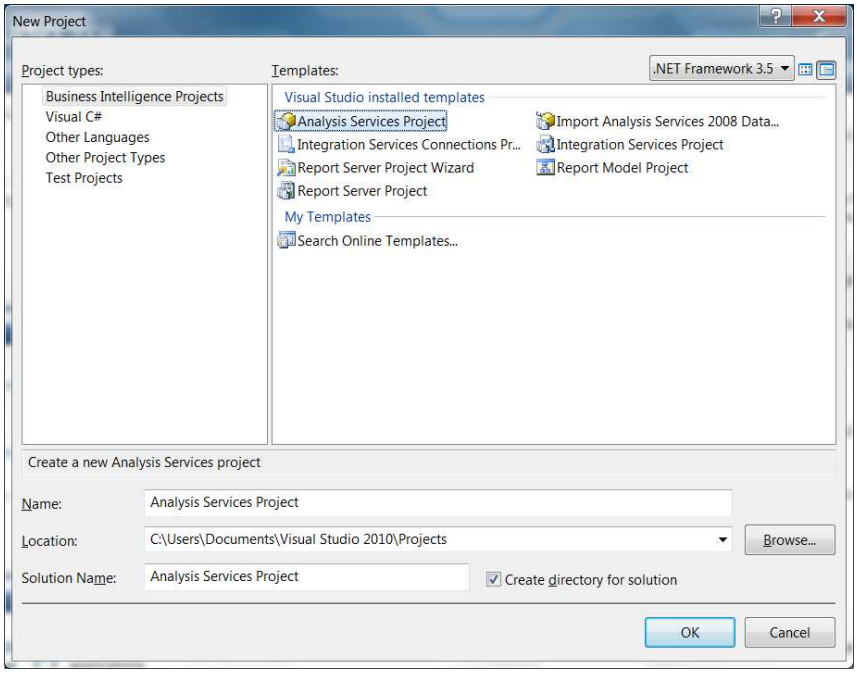


Fig. 13. Creating new *Analysis Services Project*

4. First we have to create *New Data Source* by right-clicking *Data Source* and selecting *New Data Source*... from the context menu. The *Data Source Wizard* should launch.

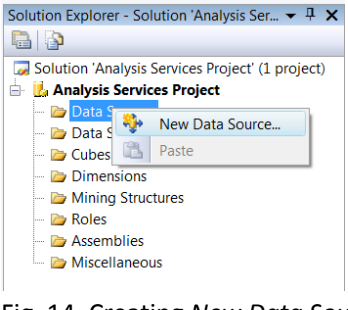


Fig. 14. Creating *New Data Source*

5. Select *Next* > on the welcoming screen. Create a new connection (Press *New*...) in the same way like in step 18 of the *Integration Services* example but connect to the *AdventureWorksDW2008* database (Sample Microsoft Databases - http://msftdbprodsamples.codeplex.com). When you have created the connection, select *Next>* and provide *Analysis Services* login information (if you have the default *SQL Server* installation you can select *Use the service account*). Then click *Finish*.

6. Now we have to create a fact table and dimension tables. In order to do so right-click on *Data Source View* and select *New Data Source View*.... Once again Microsoft has prepared a *Wizard* for us. Pick *Next* on the welcoming screen and choose our previously created *Data Source* (*Adventure Works DW2008*) and click *Next*.

7. Now we have to select from our data source the tables we would like to analyze. We choose *InternetSales* as a fact table and *Customer*, *Geography*, *Product* and *Date* as dimensions (as you can see Microsoft designed this database for *Analysis Services* so the name of a table already provides us with a hint whether it should be a fact or dimension table). When we have selected our tables we should click *Next* and then *Finish*

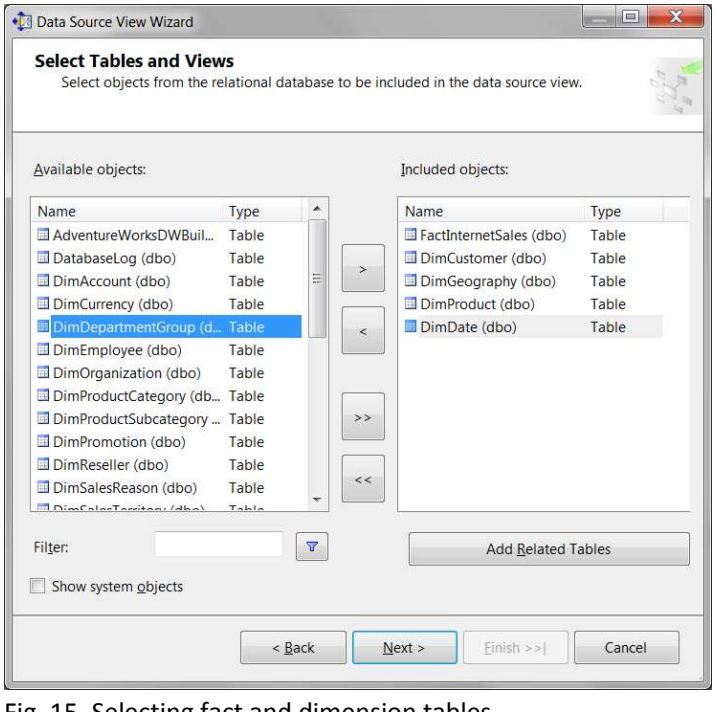


Fig. 15. Selecting fact and dimension tables.

8. Now our *Data Source View* should look like this.

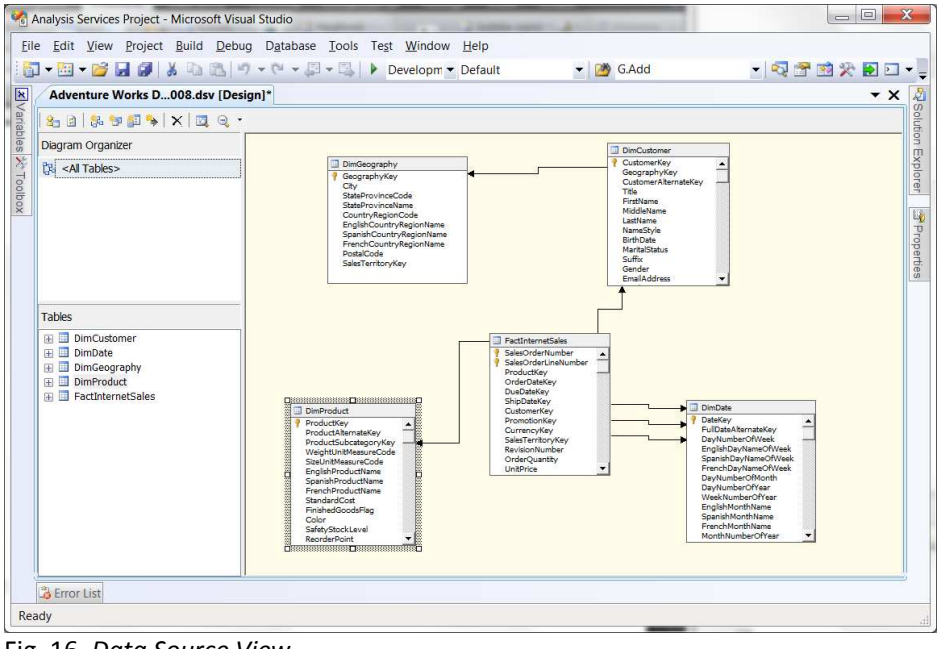


Fig. 16. *Data Source View*.

9. Now we can create our cube. We do so by right-clicking *Cubes* and selecting *New Cube*.... Skip the wizard welcoming screen and select the creation method by checking *Use existing tables* option and clicking *Next*.

10. Select our fact table – *InternetSales* – as a *Measure Group Table* and click *Next*. In future in more complex cases you can use the *Suggest* button (but be aware that in some cases it may select wrong tables).

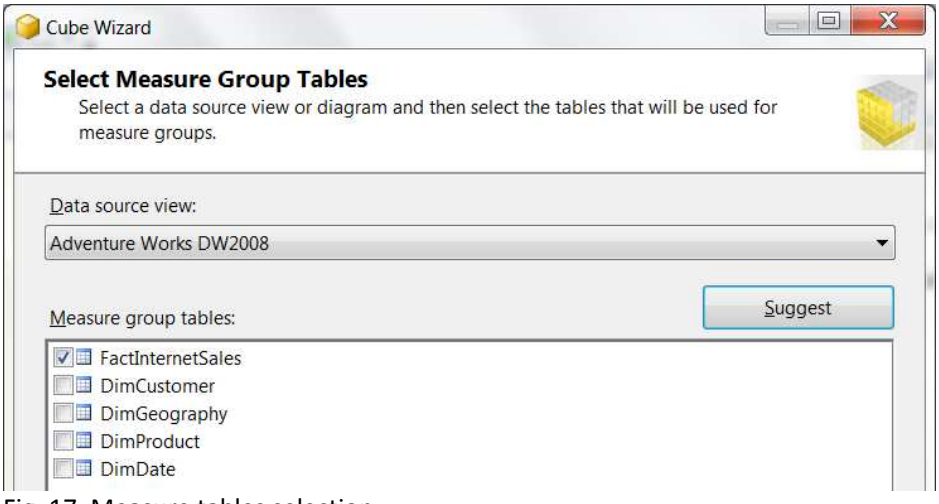


Fig. 17. Measure tables selection.

11. After that we have to select measures we would like to investigate. In this example we will choose all of them and press *Next*.

12. After we have chosen measures for our cube we have to choose the dimensions. The wizard will automatically propose some of them based on our *Data Source View*. We select all of them and click *Next*. The next screen is the summary screen where we can check what measures and dimensions we have created. To complete our cube we have to press *Finish*.

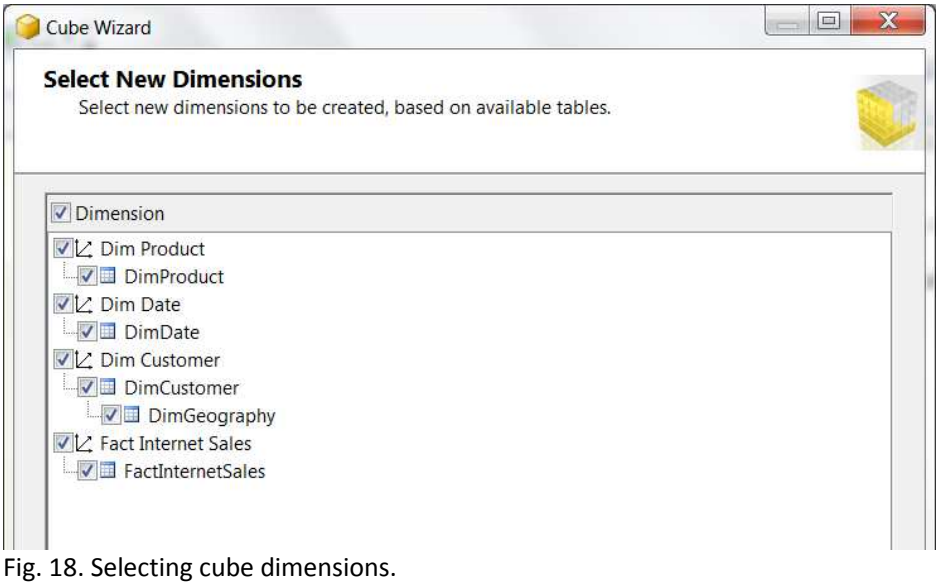


Fig. 18. Selecting cube dimensions.

13. And finally we have created our first cube but... have not finished it yet ☺.

14. When you have created the cube you can add a number of things (calculated measures, partitions, aggregations and so on) to it in order to increase its functionality. In this example you will learn how to add a hierarchy to your dimensions. In the *Dimension* folder you can add new dimensions and edit existing ones. To edit the existing dimension double-click it.

15. We will try to edit the *Date dimension*. Double click *Dim Date*. We have a dimension only with the key field and no hierarchies, so we have to build one. First, we select from *Data Source View* tables columns in which we are interested in. In this case it will be *Calendar Year*, *Semester*, *Quarter* and *English Month Name*. We drag them from *Data Source View* and drop in *Attributes*. When we have chosen attributes we can build a hierarchy from them. *Date* is the most natural kind of hierarchy there could be. *Year-Semester-Quarter-Month-Day-Hour-Minute-Second* etc. We start to build a hierarchy by dragging the top element (in this case *Year*) to the *Hierarchies* window. After that we add next levels of the hierarchy – *Semester, Quarter, Month*. When finished our dimension should look like this:

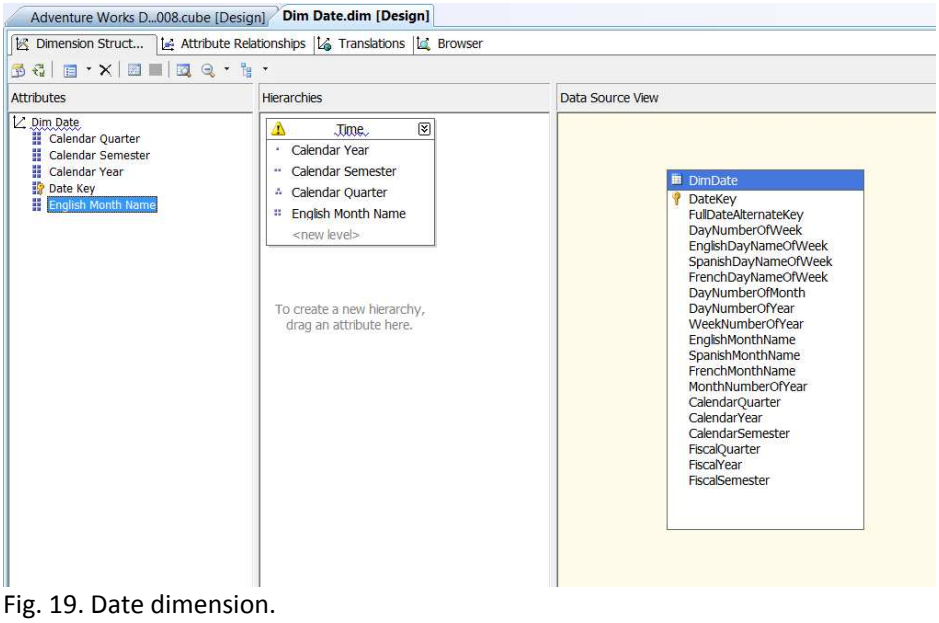


Fig. 19. Date dimension.

16. In the same way we can edit the rest of our dimensions.

17. No we can go to the cube view (double click the cube), deploy it to the database (*F5*) and process it (*Cube->Process*...). In the *Process Cube* window click *Run*... and wait until your cube is processed successfully. Then press *Close* twice.

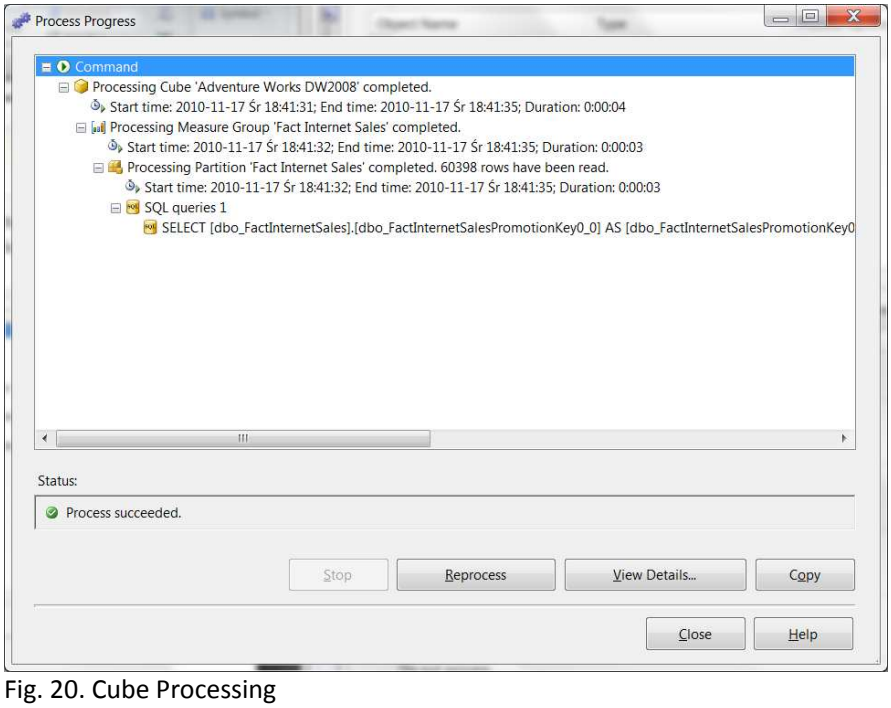


Fig. 20. Cube Processing

18. Now the cube is finished and we can browse data in it using either the *Browser tab* in the cube window or by connecting to the *Analysis Services* database using *SQL Server Management Studio*. We investigate data by dragging and dropping dimensions on rows, columns and filters and measures as table content. Finally we can analyze our sales in individual years, semesters, quarters and months.

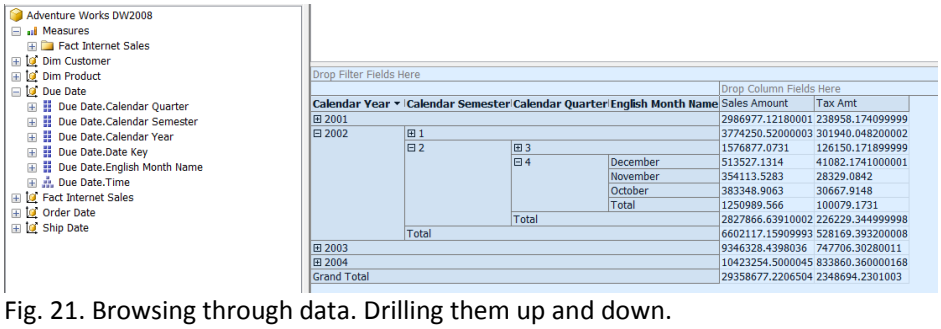


Fig. 21. Browsing through data. Drilling them up and down.

19. But probably you are thinking now: “That’s nice but if I ask my boss to learn how to operate SQL Server Management Studio I will be fired. Is not there a better way to analyze the data?” And the answer is “Yes, there is: SQL Server Reporting Services.”

SQL Server Reporting Services and Sample Report

"SQL Server Reporting Services provides a full range of ready-to-use tools and services to help you create, deploy, and manage reports for your organization, as well as programming features that enable you to extend and customize your reporting functionality.

Reporting Services is a server-based reporting platform that provides comprehensive reporting functionality for a variety of data sources. Reporting Services includes a complete set of tools for you to create, manage, and deliver reports, and APIs that enable developers to integrate or extend data and report processing in custom applications. Reporting Services tools work within the Microsoft Visual Studio environment and are fully integrated with SQL Server tools and components.

With Reporting Services, you can create interactive, tabular, graphical, or free-form reports from relational, multidimensional, or XML-based data sources. You can publish reports, schedule report processing, or access reports on-demand. Reporting Services also enables you to create ad hoc reports based on predefined models, and to interactively explore data within the model. You can select from a variety of viewing formats, export reports to other applications, and subscribe to published reports. The reports that you create can be viewed over a Web-based connection or as part of a Microsoft Windows application or SharePoint site. Reporting Services provides the key to your business data."[3]

To create a sample report:

1. Start *Microsoft Visual Studio 2010* or *SQL Server Business Intelligence Development Studio* (a free version of *Visual Studio* which allows to create BI processes but may have limited functionality).
2. Select *File->New->Project*... or press *Ctrl+Shift+N*
3. In *Project types* select *Business Intelligence Project*, in *Templates* select *Report Server Project*, choose Name, project Location and press *OK*.

[3] http://msdn.microsoft.com/en-us/library/ms159106.aspx

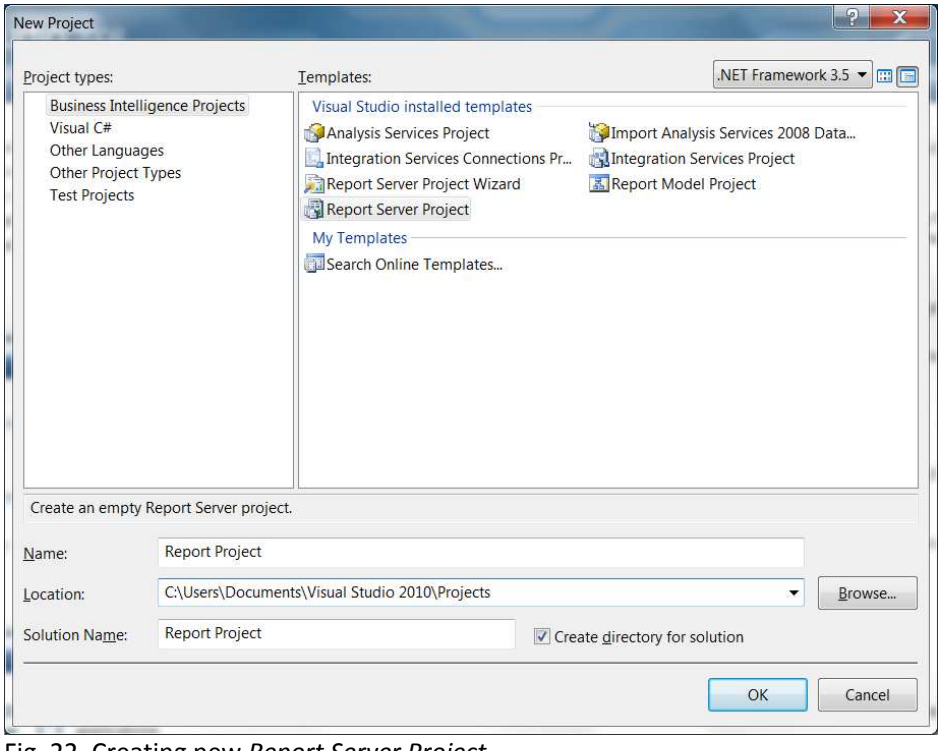


Fig. 22. Creating new *Report Server Project.*

4. To create a report and present data from our cube we need to connect to it. So right-click on *Shared Data Sources* and select *Add New Data Source*. Enter the *Data Source* name and *Type*. In our case it is *Microsoft SQL Server Analysis Services*. In the *Connection* string we have to enter the name of the server and database to which we have deployed our cube, we can do it manually or by clicking the *Edit*... button and filling the form.

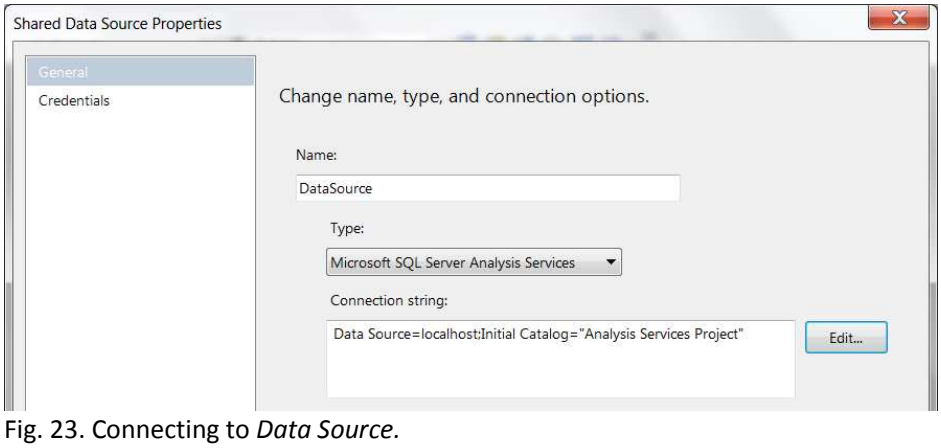


Fig. 23. Connecting to *Data Source.*

5. When we have created *New Data Source* we can start with the report. Right-click on *Reports* and select *Add New Report*...

6. Skip the wizard welcoming screen and select the previously created *Data Source*. Click *Next*.

7. Now we have to prepare a query which will extract data to our report. We can do it manually by typing the query string or use *Query Builder*. We will use *Query Builder*.

8. In *Query Builder* we can create a query in a similar way like in the *cube browser*. We just have to drag and drop measures (*Sales Amount* and *Tax Amount*) and dimensions (previously created the *Time* hierarchy from the *Date* dimension). We can also add dimensions as filters/parameters. To do so we have to click *Select dimension*>, select the dimension (e.g. *Product*), hierarchy or a comparison operator and check the *Parameters* button. We can also narrow our report only to specific products by selecting them in *Filter Expression* (instead of *{All}*). When we have finished designing the query we can click *OK* and then *Next*.

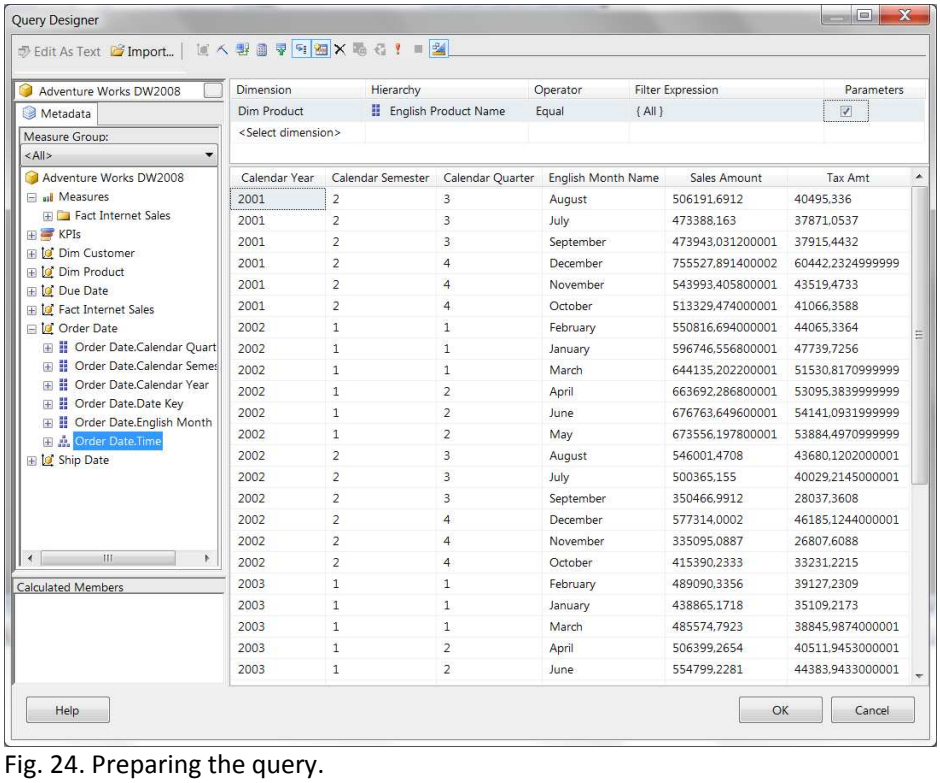


Fig. 24. Preparing the query.

9. After that we have to choose if we need to present our data as a table or matrix. In this example we have only one hierarchy/attribute so we choose the *Tabular* button.

10. After choosing the table we have to design it by dragging and dropping *Available items* (measures and dimensions) to appropriate windows. *Items* in the *Details* window will be presented as data for investigation so that's why we placed our measures there. Items in the *Group* window are used for grouping (similar to *group by* in *TSQL*) so we placed there our dimensions. Items in the *Page* window will also be used for grouping but each group will be presented on a different report page. To finish designing the table click *Next*.

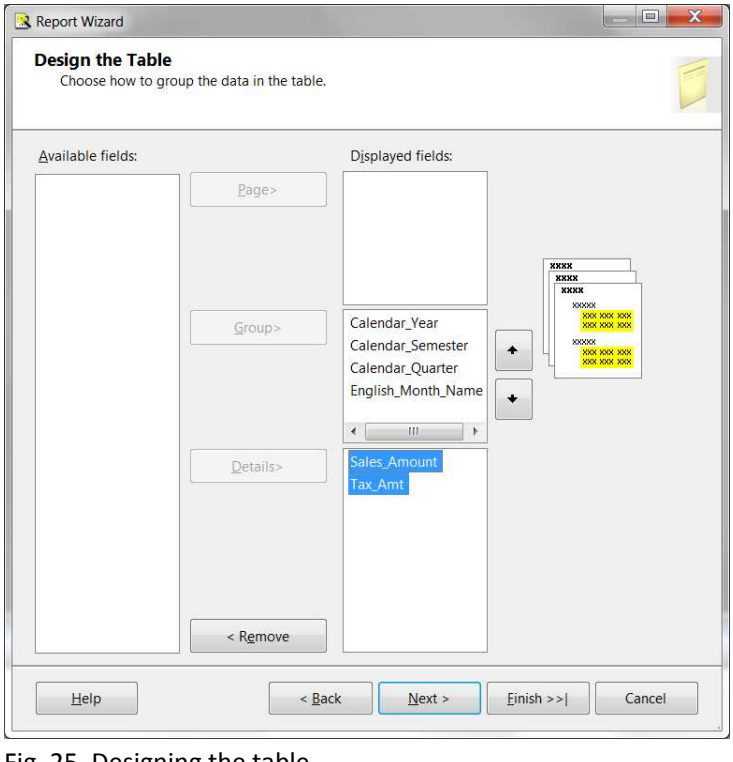


Fig. 25. Designing the table.

11. Next we have to choose the *Table Layout*. Because we have a hierarchy we can select *Enable drilldown*. Additionally we can select *Include subtotals* to calculate sums for each hierarchy level.

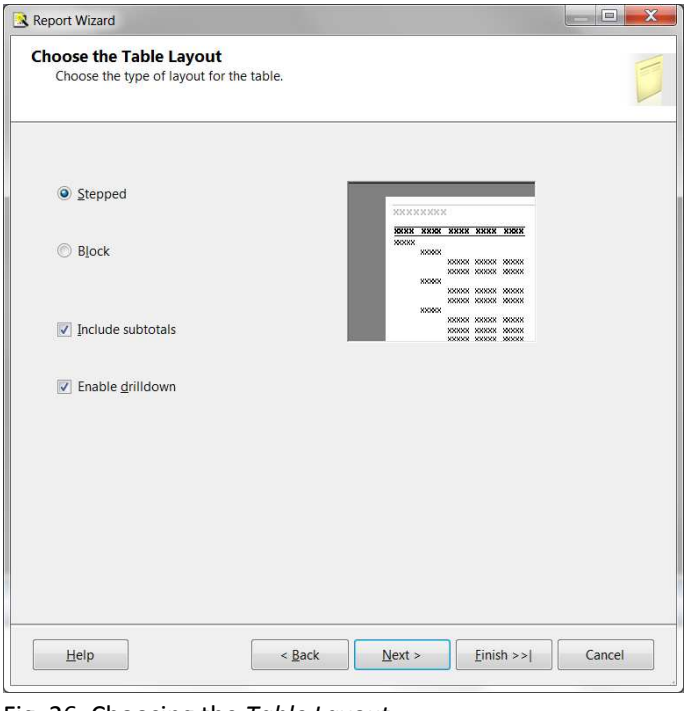
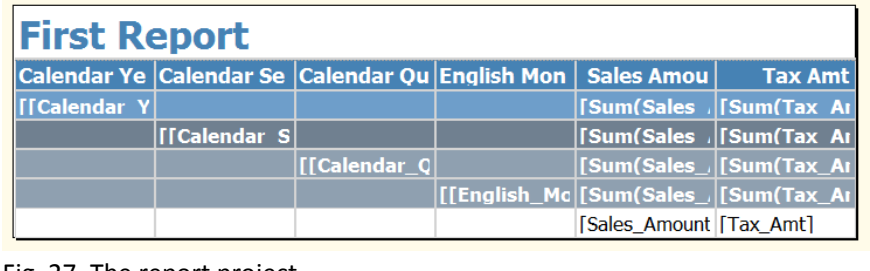


Fig. 26. Choosing the *Table Layout*

12. At the end we can select table colours layout from a few available templates. After that select *Next*, enter the report name and click *Finish*. And we have our first report.



First Report

Calendar Ye	Calendar Se	Calendar Qu	English Mon	Sales Amou	Tax Amt
[[Calendar_Y				[Sum(Sales_	[Sum(Tax_Ar
	[[Calendar_S			[Sum(Sales_	[Sum(Tax_Ar
		[[Calendar_Q		[Sum(Sales_	[Sum(Tax_Ar
			[[English_Mc	[Sum(Sales_	[Sum(Tax_Ar
				[Sales_Amount	[Tax_Amt]

Fig. 27. The report project

13. Now we can change column layouts, aggregation method for measures, add new columns and rows; in other words, we can remodel our report as we want or need. Using the Preview tab we can check if the report is ok. As we can see below we can drill down our measures and filter our report just as we want.

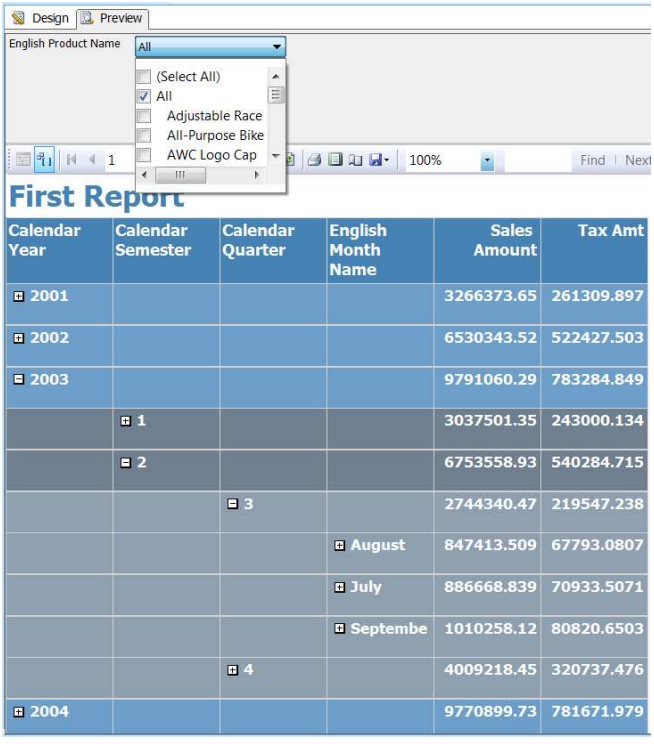


Fig. 28. Preview of the report.

14. When we have designed the report we have to deploy it to *Report Server*. First make sure that you set your report as *StartItem* in *Report Project Properties* page. Also, add the *Report Server URL* (if you have used the default settings during *SQL Server 2008* installation it should be http://localhost/ReportServer)

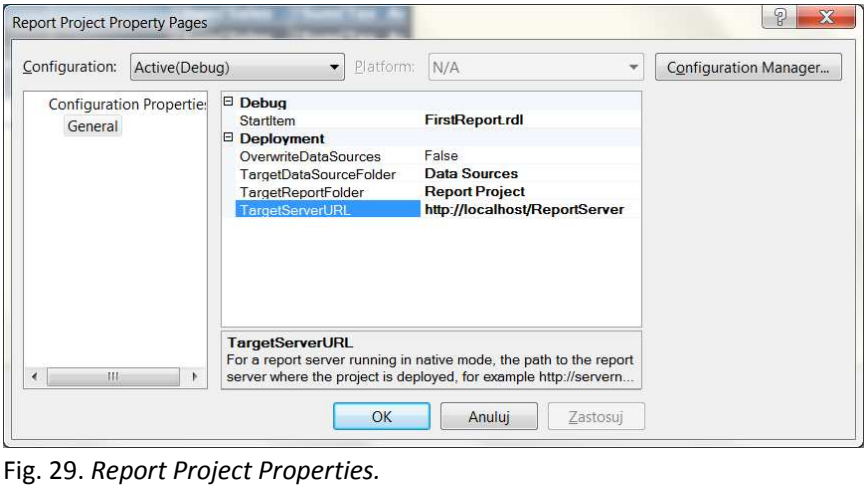


Fig. 29. *Report Project Properties.*

15. Next click *F5* to compile and debug the report in order to check if the report is correct.

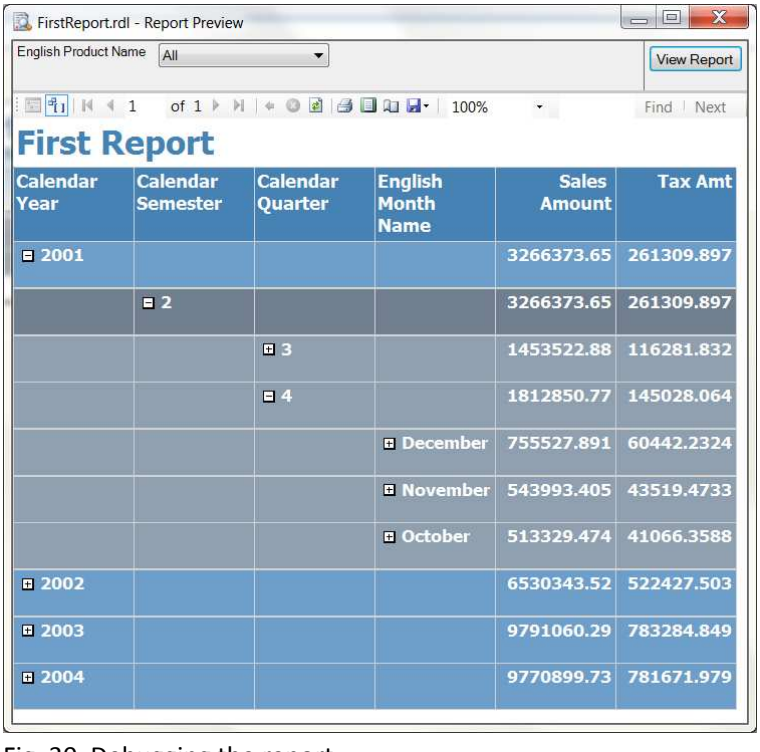


Fig. 30. Debugging the report

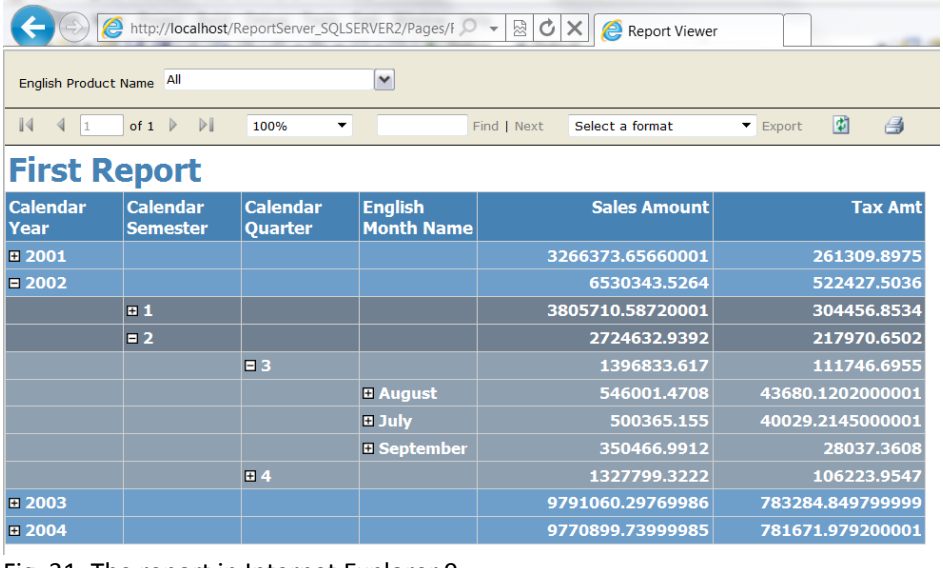
16. The last step is to deploy (Publish) our new report to *Report Server* (remember that you have to turn on *Report Server*). Just right click on the *Project* and select *Deploy*.

```
------ Build started: Project: Report Project, Configuration: Debug ------
Build complete -- 0 errors, 0 warnings
------ Deploy started: Project: Report Project, Configuration: Debug ------
Deploying to http://localhost/ReportServer_SQLSERVER2
Deploying data source '/Data Sources/DataSource'.
Deploying report '/Report Project/FirstReport'.
Deploy complete -- 0 errors, 0 warnings
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
========== Deploy: 1 succeeded, 0 failed, 0 skipped ==========
```

Fig. 31. Successful deployment.

17. Now we can just open a web browser (*IE7* or newer is advised), type our *Report Server* address (if you have used the default settings during *SQL Server 2008* installation it should be http://localhost/ReportServer), select the appropriate folder (by default our project name) and launch the report. Now you can send the link to your boss who can check the report every day for

new data. The report is updated automatically when the cube (the data source for the report) is updated.



First Report

Calendar Year	Calendar Semester	Calendar Quarter	English Month Name	Sales Amount	Tax Amt
⊞ 2001				3266373.65660001	261309.8975
⊟ 2002				6530343.5264	522427.5036
	⊞ 1			3805710.58720001	304456.8534
	⊟ 2			2724632.9392	217970.6502
		⊟ 3		1396833.617	111746.6955
			⊞ August	546001.4708	43680.1202000001
			⊞ July	500365.155	40029.2145000001
			⊞ September	350466.9912	28037.3608
		⊞ 4		1327799.3222	106223.9547
⊞ 2003				9791060.29769986	783284.849799999
⊞ 2004				9770899.73999985	781671.979200001

Fig. 31. The report in Internet Explorer 9.

18. Using *Reporting Services* you can not only create reports but also charts. To do so right-click on the *Reports Folder* and create a new empty report.

19. Next go to *Toolbox*, drag and drop *Chart* element and select the type of the chart you would like to create.

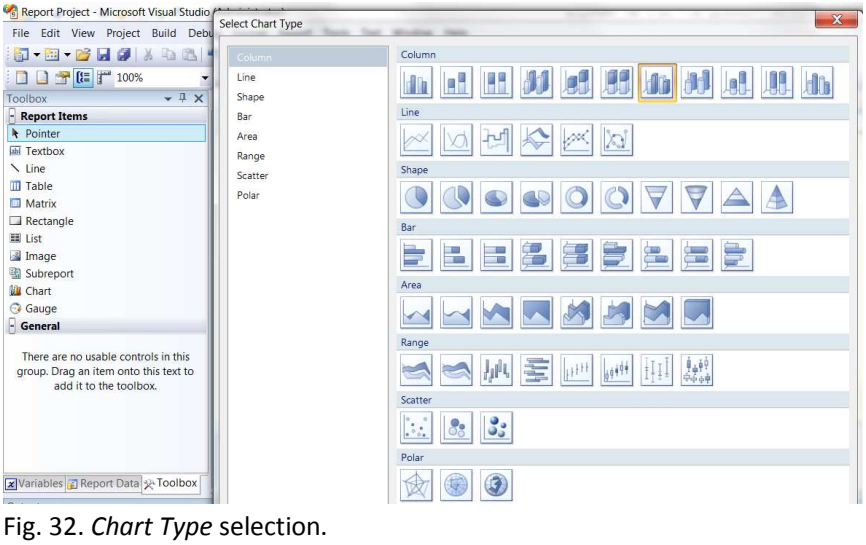


Fig. 32. *Chart Type* selection.

20. Now you have to create the data source and build the query in the same way as during the report creation. In this example we have created a little bit less complicated query and instead of selecting the whole *Time* hierarchy we select just *Year*. When the form is filled, select *OK*.

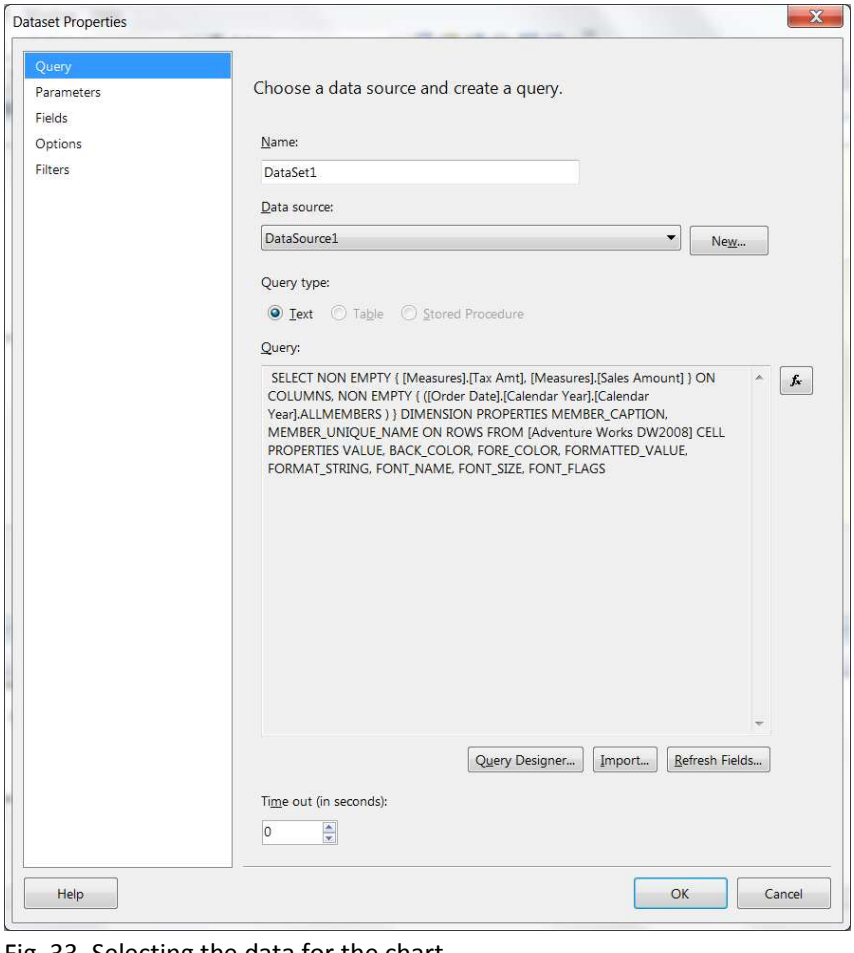


Fig. 33. Selecting the data for the chart.

21. Next we have to add data (drag and drop *Sales* and *Tax Amount* to the "*Drop data fields here*" field) and category (drag and drop *Year* to "*Drop category fields here*" field) to the chart. Now we have to improve the chart's layout. We can change the bar colours, names of axis and the chart, change the scale (e.g. to logarithmic), add bars labels, change the data aggregation method (the default is Sum), add filters (e.g. by *Products*) and so on. In other words, we can remodel our chart.

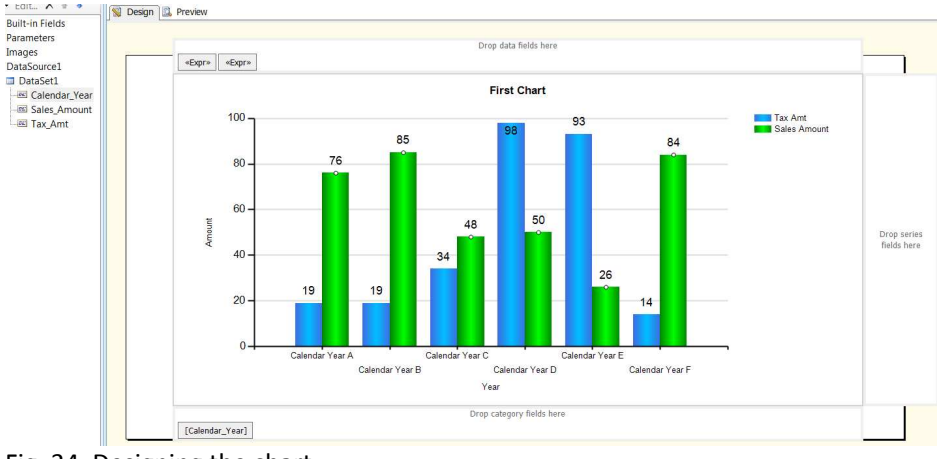


Fig. 34. Designing the chart

22. After adding finishing touch to our chart we can deploy (*Publish*) it to *Report Server* in the same way in which we have deployed the report and we can present our chart in the web browser.

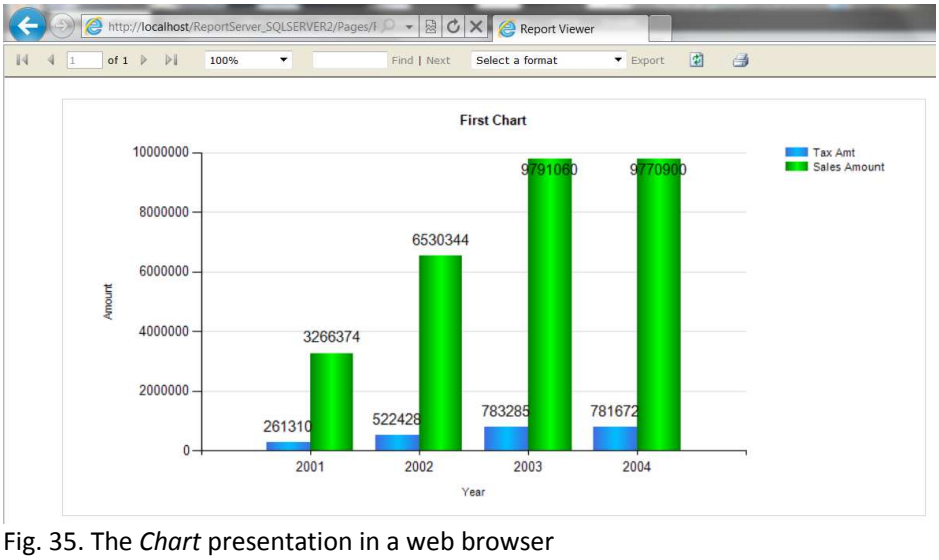


Fig. 35. The *Chart* presentation in a web browser

Additional Examples

For additional examples go to http://sqlserversamples.codeplex.com and download All Microsoft Product Samples in a Box or:

- Microsoft Integration Services Samples
- Microsoft Samples (OLAP, Data Mining, Administration)
- Microsoft Reporting Services Samples

Next install the downloaded packages and go to the installation catalogue (if you have used default settings your examples should be in C:\Program Files\Microsoft SQL Server\100\Samples), choose appropriate services (*Integration Services, Reporting Services, Analysis Services*) and select *Tutorials*.

References and Additional Information

1. SQL Server Integration Services:
   http://msdn.microsoft.com/en-us/library/ms141026.aspx
2. SQL Server Analysis Services:
   http://msdn.microsoft.com/en-us/library/bb522607.aspx
3. SQL Server Reporting Services:
   http://msdn.microsoft.com/en-us/library/ms159106.aspx
4. Introducing Business Intelligence Development Studio:
   http://msdn.microsoft.com/en-us/library/ms173767.aspx
5. Sample Microsoft Databases:
   http://msftdbprodsamples.codeplex.com
6. Text datasets - UCI Machine Learning
   Repository:http://archive.ics.uci.edu/ml/index.html
7. Sample Microsoft Databases:
   http://msftdbprodsamples.codeplex.com
8. Reporting Services Tutorial:
   http://msdn.microsoft.com/en-us/library/ms167305.aspx

Task List 1. SQL Server Integration Services, ETL - Extraction, Transformation, Loading

1. General Task List Information

In this task list students have to prepare an ETL process using *SQL Server Integration Services*. To perform this task, students have to have the appropriate software packages installed: *SQL Server 2008* with *SQL Server Integration Services* and *Microsoft Visual Studio 2010*.

2. Schedule

The report with the solution worked out by students should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. The email subject should be of the pattern: *DW_FullName_L01.*

3. Tasks – The asterisk symbol (*) denotes optional items

a) Browse and analyze 3-5 different text datasets and describe them in the report.

b) Choose two datasets containing numerical and textual data. In the report, justify your choice – the complexity of the datasets will affect the final grade.

c) Prepare an ETL process with two Data Flows (one for each dataset) using *SQL Server Integration Services*. **Document all steps with screenshots and description**. Data Flows should be executed one after another. The second Data Flow should be executed only if the first one is completed successfully. It should contain the following steps:

i. *Extraction* – Using appropriate Data Flow Sources extract data from the data sources. Remember to convert each column to the appropriate data type.

ii. *Transformation* – Prepare necessary Data Flow Transformation. You should use:

1. Split one text column into two columns (e.g. “John Doe” into “John” and “Doe”).

2. Merge a few columns into one (e.g. merge first name, surname and address into description).

3. Mathematical operation. Sum/multiply/divide – based on a few columns create a new one (e.g. Net price plus VAT into Gross price).

4. Consider also different other useful transformation. For example transform PESEL number (Polish: Powszechny Elektroniczny System Ewidencji Ludności, Universal Electronic System for Registration of the Population) into date of birth.

5. (*)Aggregation – Create a new table containing aggregated numerical data (e.g. sum of purchases of the particular customer).

iii. *Loading* – prepare a database with the appropriate tables using *SQL Server 2008* and *SQL Server Management Studio*. Next, prepare *Data Flow Destinations* to load your transformed data into your database. In the report, justify your database design.

iv. At the end, the ETL process should send an email with the report containing information whether *Data Flows* have finished with success and how many rows have been loaded (use Variables to store necessary information).

d) Prepare the report (*DW_FullName_L01.*DOCX/PDF format) containing description of all of the above issues.

4. Presentation

During classes, each student will have 5 minutes to present their task list to the supervisor. Students should be prepared to present data sets they have chosen, ETL process they have designed. Students should also point out a few difficulties they have encountered and how they resolved them.

Task List 2. Introduction to SQL Server Analysis Services – build your first cube

1. General Task List Information

In this task list, students will learn how to create *OLAP Cubes* with *MS SQL Server 2008* (*SQL Server Analysis Services*).

2. Schedule

The report containing the solution worked out by students should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L02.*

3. Tasks – The asterisk symbol (*) denotes optional items

   a) Create at least one cube for the sample *AdventureWorksDW* or *FoodMart* databases (*Sample Microsoft Databases*). Prepare screenshots with description for each following step.

   b) Define a data source.

   c) Define a fact table, measures and dimensions for the cube.

   d) Define the method of storing data (*MOLAP, ROLAP, HOLAP*). Justify your choice.

   e) Consider different aggregation options and justify your choices. Analyze granularity and the possibility of data explosion.

   f) Display and analyze the schema of the cube (*Cube Builder*). Modify it, if necessary.

   g) Process the cube.

   h) Analyze the content of the cube in a browser (Cube Browser). Estimate the usability of extracted information (who may it be useful for, what kind of decisions they may affect).

i) (*) Extend your cube by linking to another cube. This is called Linked objects (measure group) in *SQL Server 2005/2008*. Can you link to the dimension group? What is the potential usefulness of this feature?

j) (*) Split the cube into partitions. Analyze various partitioning options. Justify your choice.

k) Prepare a report (*DW_FullName_L02.*DOCX/PDF document) containing description of all above items.

4. Presentation

During classes, each student will have 5 minutes to present their task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list (e.g. creating cubes, import).

Task List 3. SQL Server Analysis services – build your own cube

1. General Task List Information

In this task list, students will create *OLAP Cubes* with *MS SQL Server 2008* (*Analysis Services*) and process them using Integration services.

2. Schedule

The report with the solution worked out by students should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L03*

3. Tasks – The (*) symbol denotes optional items

a) Create at least one cube using any online transactional database (*OLTP*). You can use either sample *OLTP* databases (e.g. *AdventureWorks* or *FoodMart*) or any external database (Sample Microsoft Databases). This is a repetition of the previous task list but for another data source (a transactional one instead of the one already prepared for data warehousing). Prepare screenshots with description for each following step.

b) Define a data source.

c) Define a fact table, measures and dimensions for the cube.

d) Define the method of storing data (*MOLAP, ROLAP, HOLAP*). Justify your choice.

e) Consider different aggregation options and justify your choices. Analyze granularity and the possibility of data explosion.

f) Display and analyze the schema of the cube (*Cube Builder*). Modify it, if necessary.

g) Process the cube

h) Analyze the content of the cube in a browser (*Cube Browser*). Estimate the usability of extracted information (who may it be useful for, what kind of decisions they may affect).

i) (*) Extend your cube by linking to another cube. This is called *Linked objects* (measure group) in *SQL Server 2005/2008*. Can you link to the dimension group? What is the potential usefulness of this feature?

j) (*) Split the cube into partitions. Analyze various partitioning options. Justify your choice.

k) Import data into the cube created in the previous tasks from this list using Integration Services. Prepare screenshots with description for all steps.

l) Create your own *SSIS package* to manipulate data (define the source, target and necessary mappings).

m) Define *Analysis Services Processing Task*.

n) Import data and check whether the process has finished successfully.

o) Using *Microsoft SQL Server Management Studio* and defined *Analysis Services Processing Task* create *Job*, which will process your cube automatically (*SQL Server Agent -> Jobs -> New Job*…).

p) (*) Write your own *Visual Basic* script that transforms the data.

q) Prepare a report (*DW_FullName_L03.*DOCX/PDF document) containing description of all of the above issues.

4. Presentation

During classes, each student will have 5 minutes to present their task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list (e.g. creating cubes, import).

Task List 4. Processing and accessing your cube –MDX, Excel, SQL Server Management Studio, SQL Server Reporting Services

1. General Task List Information

In this task list, students will learn how to use the multidimensional query language (*MDX*) and access *MS OLAP* database from external tools.

2. Schedule

The report and (*) *MDX* application should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L04.*

3. Tasks – The (*) symbol denotes optional items. Use one of the cubes created while solving the previous task lists. Prepare screenshots with description for all steps.

    a) Create at least three calculated members, also a bit more complex, e.g. percentage contribution within dimensions. Test their functionality in a browser (*Cube Browser*) – prepare screenshots for tests. In the report, list and describe created calculated members and justify their usefulness.

    b) Create at least four queries using multidimensional expressions (*MDX*). Test their functionality – prepare screenshots for tests. In the report, list and describe created queries and justify their usefulness.

    c) Optimize the cube. In the report, describe and justify each optimization, add also a screenshot of table structure. Consider the following aspects:

        i. frequent queries to the cube

        ii. database schema

        iii. (*) normalization and denormalization of the tables (star vs. snowflake schema)

iv. (*) indexes

v. granularity and the number of aggregations

vi. partitioning

d) Establish access to the cube from *MS Excel*. Use a pivot table. Analyze the cube in *Excel* using its functionality (functions, charts). Find some interesting charts (at least three) and interpret them. How can the acquired information be used by the management authorities? Provide screenshots with description from an analysis process.

e) Design five different reasonable reports using *SQL Server Reporting Services*. Reports should access your cubes and allow filtering. Prepare screenshots with description for each report.

f) (*) Implement your own *MDX* application that uses the cube. *ActiveX Data Objects Multidimensional* (*ADO MD*) and *Decision Support Objects* (*DSO*) technologies are advised. You can use any tool, any language and any database access technology. It can be either a desktop or web application. Test your application – prepare screenshots with description from tests.

g) Prepare a report (*DW_FullName_L04.*DOCX/PDF document) containing description of all of the above issues. Add your opinion about the Microsoft technology, point out advantages and disadvantages.

4. Presentation

During classes, each student will have 5 minutes to present their task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of the techniques described in the task list (e.g. creating *MDX queries*, using *Pivot Table* in *MS Excel*).

Part II. SAS

SAS (*Statistical Analysis System*) Institute software is another tool utilized in task lists no. 5 to 8. In particular, students will make themselves familiar with *4GL* – a manipulation language specialized for statistical analysis as well as multidimensional modelling and OLAP analysis performed by means of this tool.

Introduction to SAS and 4GL

A fourth-generation programming language (abbreviated *4GL*) is a programming language or a programming environment designed with a specific purpose in mind, such as the development of commercial business software. All *4GLs* are designed to reduce programming effort, the time it takes to develop software, and the cost of software development.

When you first start *SAS*, the five main *SAS* windows open: *Explorer*, *Results*, *Program Editor*, *Log*, and *Output*.

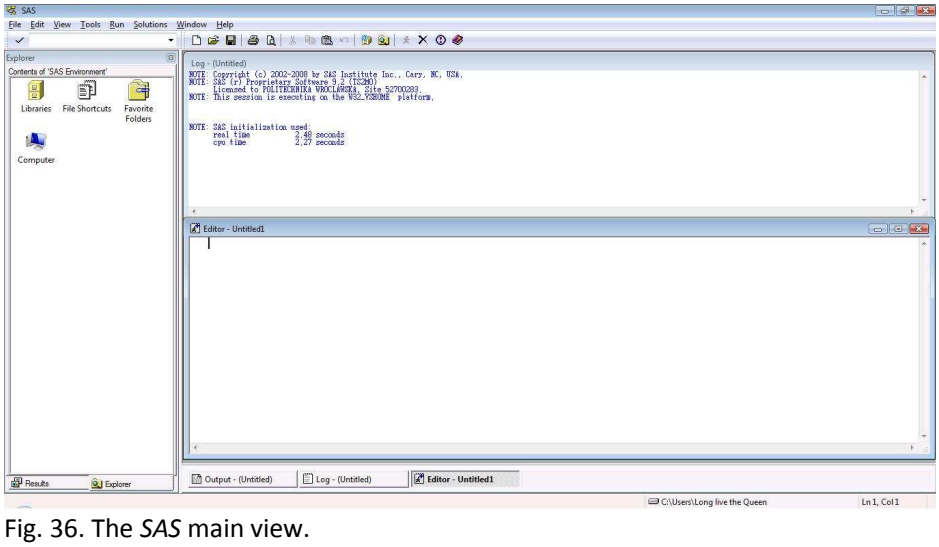


Fig. 36. The *SAS* main view.

SAS build-in libraries.

1. *Sashelp* is a permanent library that contains sample data and other files that control how *SAS* works at your site. This is a read-only library.

2. *Sasuser* is a permanent library that contains *SAS* files in the Profile catalogue that store your personal settings. This is also a convenient place to store your own files.

3. *Work* is a temporary library for files that do not need to be saved from session to session. **Note that files stored in this library are erased after the session is terminated!** Store your files in your own library to preserve them.

How to create your own library:

1. You can create *SAS* libraries using a point-and-click interface.

   a) Click *View* → *Explorer*.

   b) Click *File* → *New* in libraries.

   c) In the *New Library* window, specify information for the new library. If you want the library to be created at the beginning of each *SAS* session, click *Enable at startup.*

   d) Click *OK*.

2. You can also type the following code into the editor and press *Submit*

```
libname lib_name 'E:\path\on\the\disc';
```

How to import data into a library:

1. If you have *SAS/ACCESS Interface* to *PC Files licensed*, you can import *PC database files* using the *Import Wizard*:

   a) In *SAS*, click *File* → *Import Data*.

   b) When the *Import Wizard* opens, follow the directions for importing data. Choose the library it will belong to and give it a name in the *Member* field.

2. You can view and save the *PROC IMPORT* code that the *Import Wizard* generates.

How to view the structure of a database:

1. Type the following text into the editor and press *Submit*

```
proc contents data=my_library._all_;

run;
```

2. This will display the information about all tables in *my_library*. To choose a specific table, use `my_library.table_name` instead.

Functions you may find useful:

1. where not (xpesel is missing); /* only observations with non-empty pesel */

2. `SUBSTR(character-expression, position-expression [, length-expression ] ) /* creates a substring of a string; note: it starts with 1!*/`

3. `if expression1 then expression2; else expression3;`

4. `myString='Ala'||' MaKota'; /* string concatenation*/`

5. `INPUT(source, <? | ??>informat.)`

    a) ***source*** - contains the SAS character expression to which you want to apply a specific informat.

    b) **? or ??** - The optional question mark (?) and double question mark (??) format modifiers suppress the printing of both the error messages and the input lines when invalid data values are read. The ? modifier suppresses the invalid data message. The ?? modifier also suppresses the invalid data message and, additionally, prevents the automatic variable _ERROR_ from being set to 1 when invalid data are read.

    c) ***informat.*** - is the SAS informat that you want to apply to the source.

6. FORMAT variable(s) <format> - sets variable print format

7. INT(arg) returns floor of a number

8. LENGTH(arg) – returns length of the argument

9. MAX(argument,...) returns the maximal value from arguments

10. MIN(argument,...) returns the minimal value from arguments

11. MEAN(arg1,arg2,...) returns the mean from not missing values arguments

12. MDY(m,d,y) returns the number of days since 01.01.1960
13. _N_ – the actual DATA STEP number (variable)
14. RETAIN x; – saves the value of variable x to the next data step
15. ROUND(arg,unit) – rounds a number to the nearest 'unit'
16. SUBSTR(string,pos,len) – substring
17. SET <[library.]name of data set> - determines active data set
18. WHERE (condition) – determines which observation will be processed
19. OUTPUT – in DATA STEP saves actual values of all variables as observations to output data set

EXAMPLES

1. assignment

   x = 5;

2. set, load a table or tables (in the given order),

   set table1 table2;

3. if-else

   if x < 5 then group = "A";
   else if x > 100 group = "B";
   else group = „C";

4. Where clause

```
Data step Test.ZKara;

set Test.Czytelnik;

  where not (xpesel is missing) and payment between 1.00
and 22.00;

keep iden1 payment;

run;
```

5. do-end

a. groups commands into blocks:

```
if years > 5 then do;
        months = years * 12;
end;
```

b. loops: `do while(condition); statement; end;`

```
n = 0;
do until(n >= 5);
        put n=; n + 1;
end;
```

6. Basic statistics

```
proc means
      data=Test.Readers2
      maxdec=2 n mean std min max stderr median;
      var AGE;


title 'Info on reader age.';
run;
```

7. Sorting

```
proc sort
      data=Test.Czytelnik( where =( not (xpesel is missing)
and payment>0))


      out=Test.SortujKare;
    by payment;
run;
```

8. Sql queries

```
proc sql;
create table Test.Ukarani as
      select Czytelnik.iden1, xpesel, wydzial, kara
```

```
      from Test.Czytelnik inner JOIN Test.Wypozyczenia on
Wypozyczenia.iden1=czytelnik.iden1

      where (xpesel is not null) and kara>0

      order by kara;

quit;
```

9. Merging

```
data sortIden;

set Test.Wypozyczenia;

where not(iden1 is missing);


proc sort data=sortIden;

by iden1;


PROC MEANS DATA=sortIden NOPRINT;

by iden1;

OUTPUT OUT=Test.FeqTable N=iden1;


data sortCzytelnik;

      set Test.Czytelnik;


proc sort data=sortCzytelnik;

by iden1;


data Test.ReaderData;

merge Test.FeqTable sortCzytelnik;

 by iden1;

run;
```

SAS Multidimensional Database

A *multidimensional database* (*MDDB*) is a specialized storage facility that enables data to be pulled from a data warehouse or other data sources for storage in a matrix-like format. The *MDDB* enables users to quickly retrieve multiple levels of previously aggregated data through a multidimensional view.

How to create first MDDB in SAS

1. Choose *EIS*/OLAP A*PPLICATION BUILDER* from the *Solution* menu.

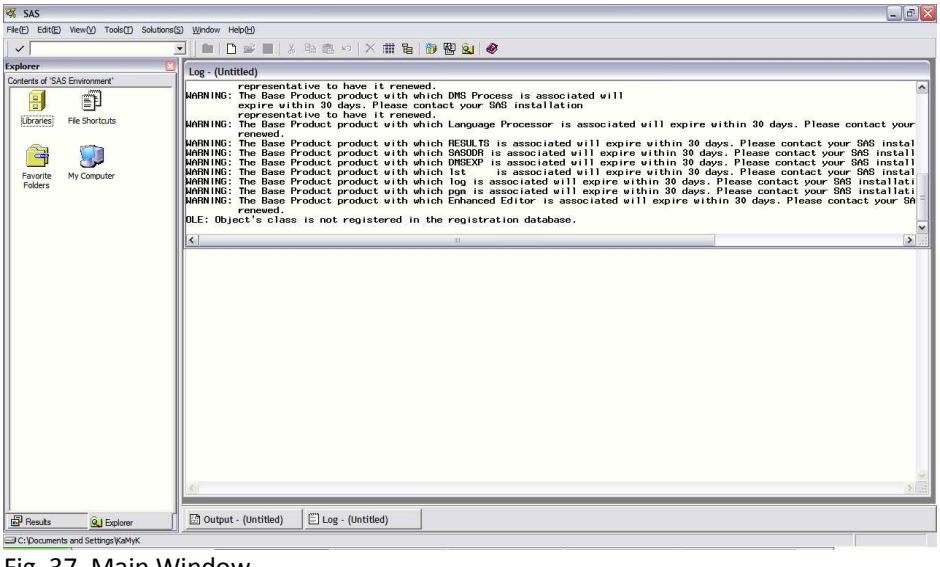


Fig. 37. Main Window

2. From *EIS Main Menu* choose *METABASE* to register your table in metabase.

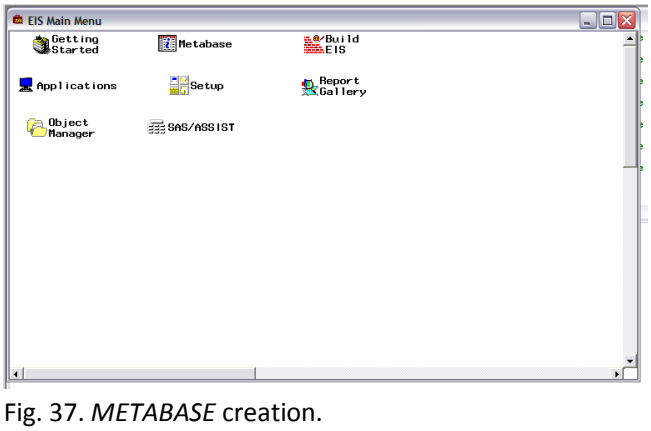


Fig. 37. *METABASE* creation.

3. Click the *Add* button in the tables box.

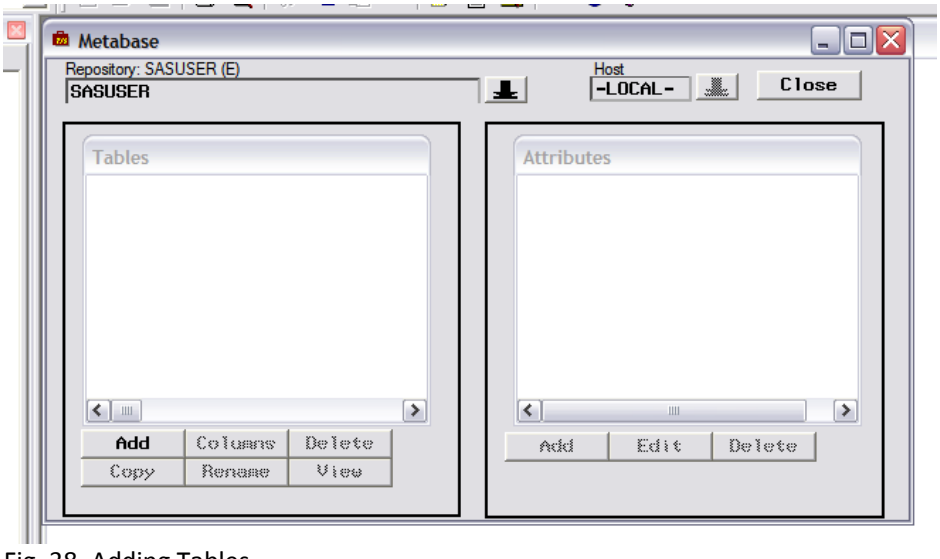


Fig. 38. Adding Tables

4. Select the table which you are interested in from the *Available* box. If there is no table which you are interested in, change the path value to another library. Table names will appear in the *tables box* (in the *metabase* window, Fig. 38.)

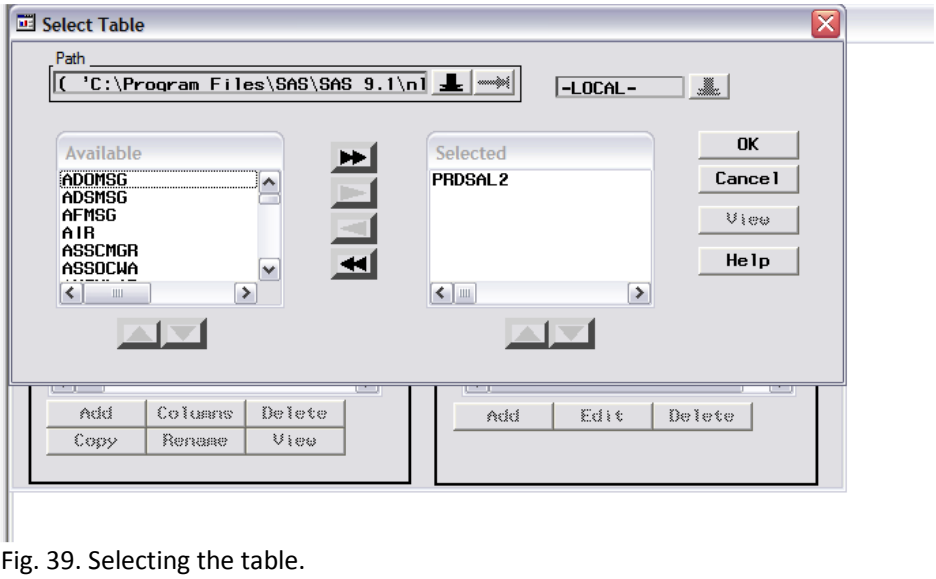


Fig. 39. Selecting the table.

5. Select the table which you added in the previous step. And click the *COLUMN* button. The Column window will appear. In the *Columns box* you can see all the available columns from the table which you selected in *point 4*. Select the

columns which you are interested in and press *Add* from the *attributes box*.

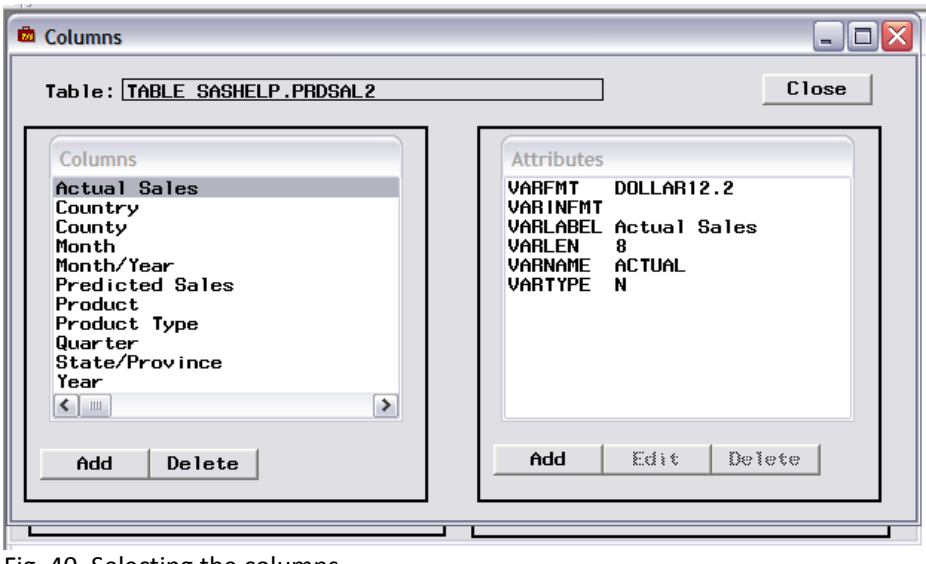


Fig. 40. Selecting the columns.

6. Select the attributes for your columns. Click the *OK* button when you finish.

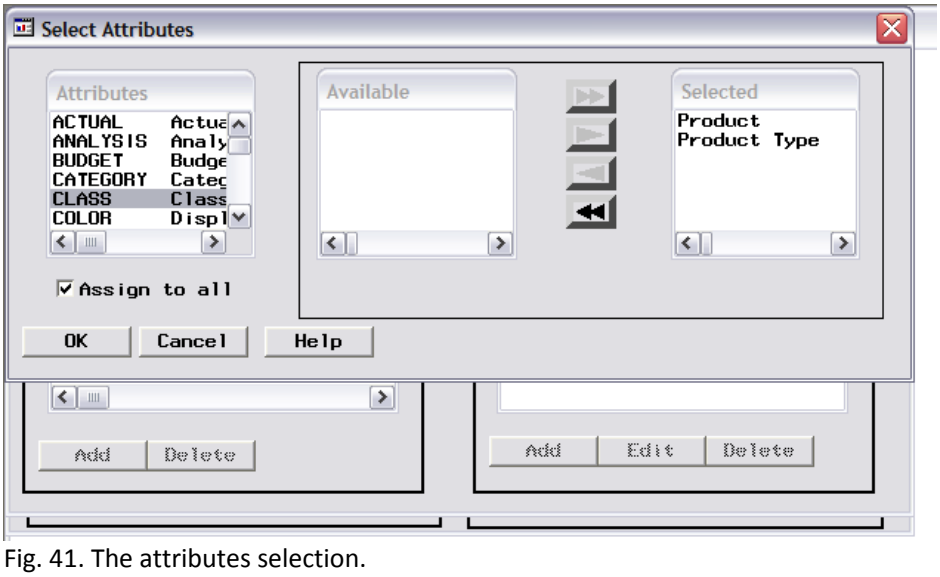


Fig. 41. The attributes selection.

If you choose the analysis attribute you also have to select an analysis type such as *SUM*, *MIN*, *MAX* etc.

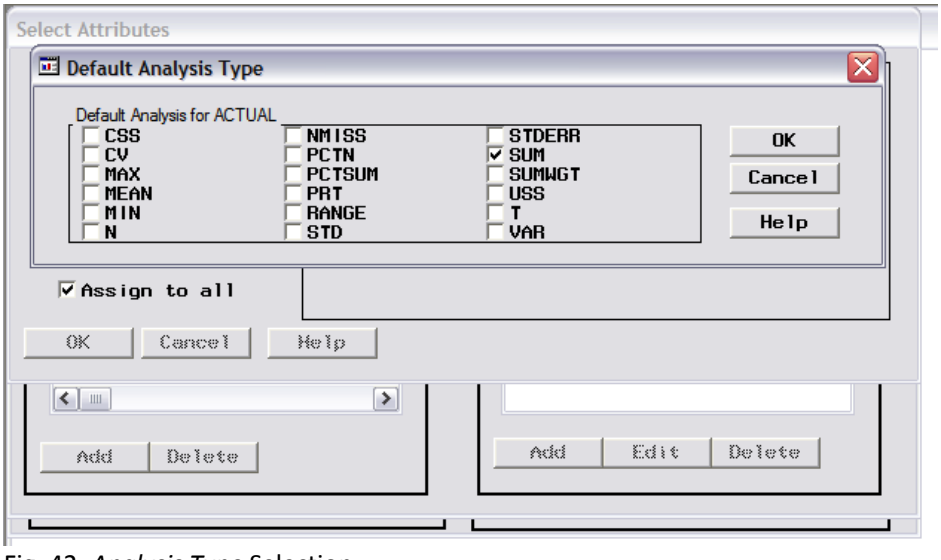


Fig. 42. *Analysis Type* Selection

7. If you want to create hierarchical values press the *Add* button from the *attributes box* in the *metabox* window. Select *HIERARCH* from the *Select Attributes* window.

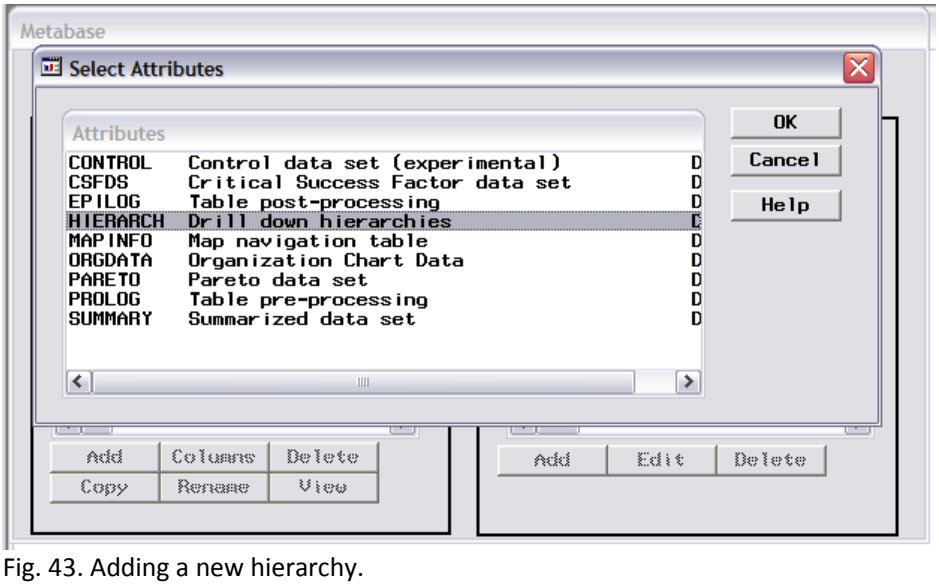


Fig. 43. Adding a new hierarchy.

8. The *Table Hierarchies* window will appear. In the *Available box* you will see all columns which can be used to build hierarchical values. Choose columns which you are interested in and move them (by clicking the arrow) to the *Selected box*. Type values name and press *OK* (if you want to create more

hierarchical values press *Add* instead of *OK*) and you have registered the table in the *metabase*.

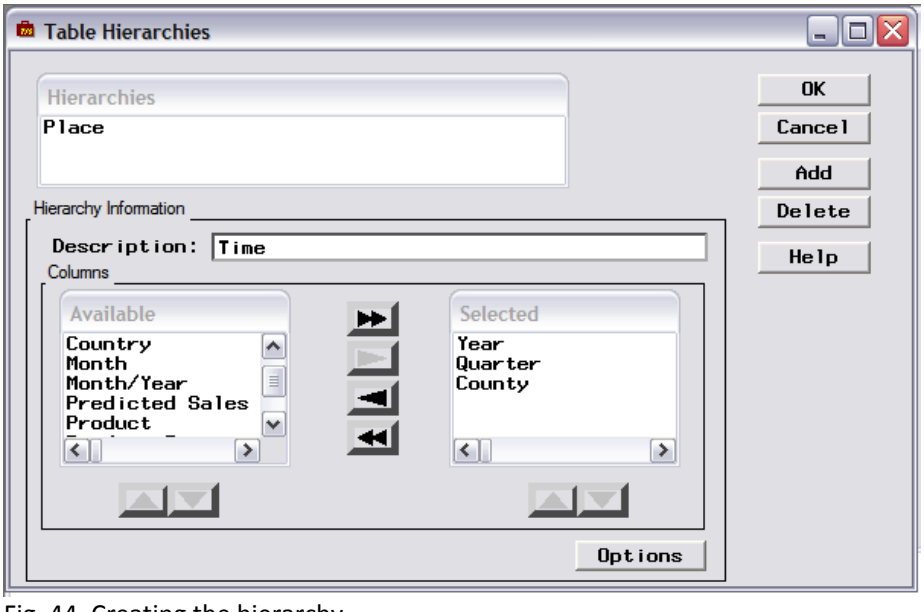


Fig. 44. Creating the hierarchy.

9. Select *DATA ACCESS* from the *Object Database* box and *MULTIDIMENSIONAL DATABASE* from the *Objects box*. Press the *Build button*. The *SAS/EIS Multidimensional Database* window will appear.

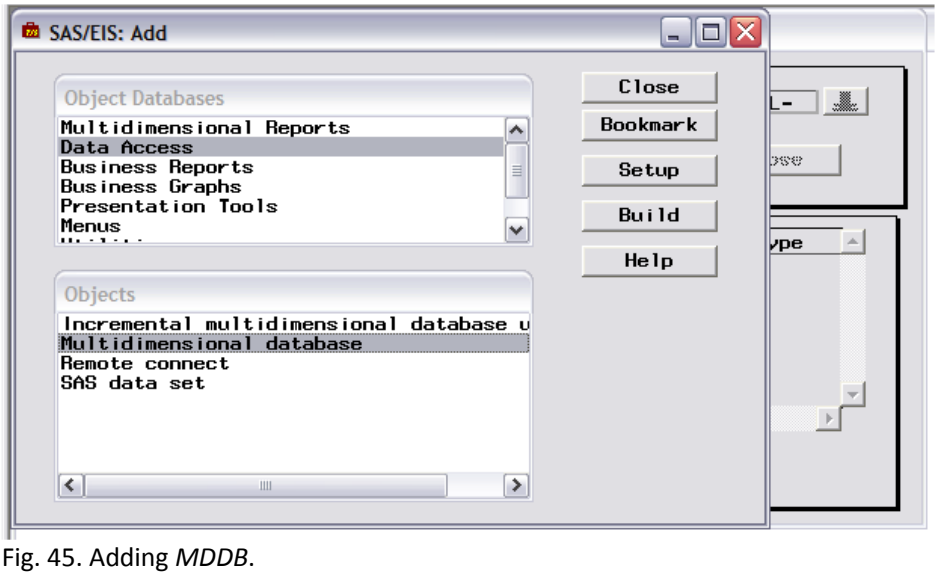


Fig. 45. Adding *MDDB*.

10. Now you have to fill in all fields. Type the name and description in proper fields. Press the arrow near *MDDB* to select the name and path (*library*) where *MDDB* will be saved. Press the arrow near the table to select the table from *metabase* on which you want build *MDDB*. Select *Dimension* and *Analysis* attributes by clicking the arrow near this fields. When you have all fields filled press *Create*.

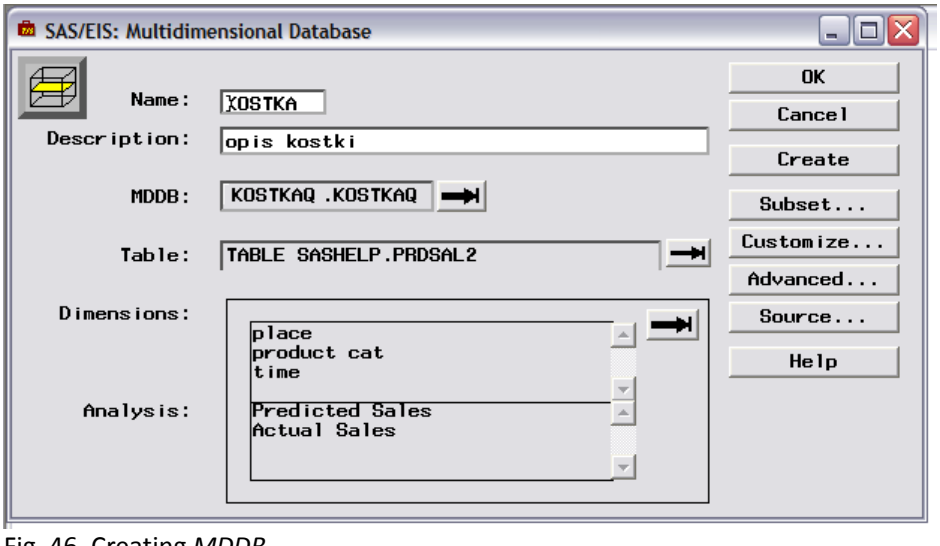


Fig. 46. Creating *MDDB*

11. And finally you have created *MDDB* in *SAS*. You can see your *MDDB* in the library which you set in this step.

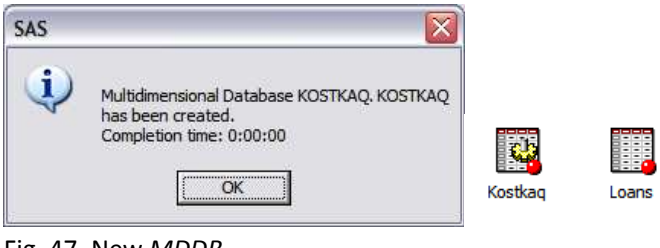


Fig. 47. New *MDDB*

Browsing MDDB – multidimensional report

Now probably you want to view your table. You can click on its icon in the library but it is rather useless if you want to analyze data. On the following pages you will find out how you can present data from your *MDDB*.

1. From *SAS/EIS Build EIS* window (the same as in point 9 in the previous section). Click *Add*. Select *Multidimensional Reports* from *Object database* and press the *Build* button. *SAS/EIS Multidimensional Report* window will appear.

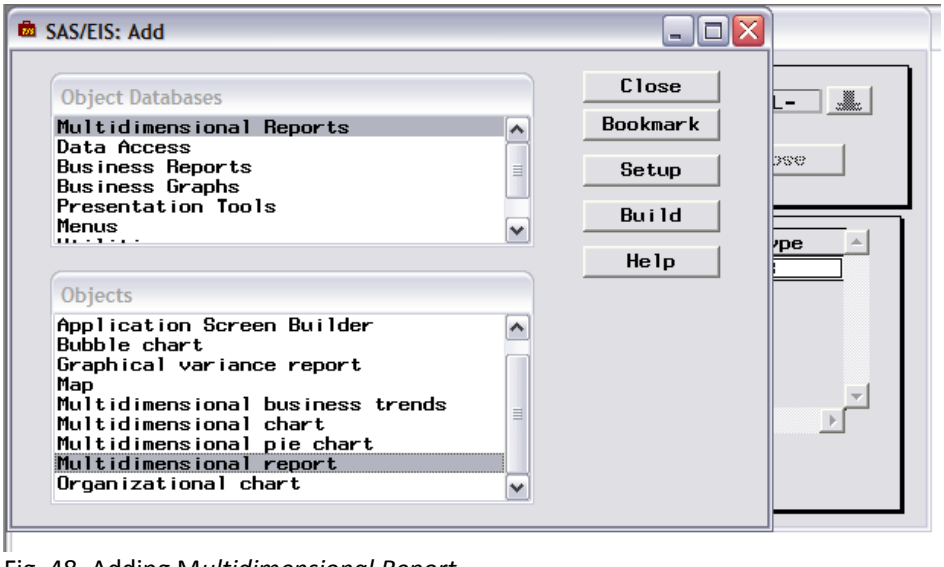


Fig. 48. Adding M*ultidimensional Report.*

2. Fill in name and description fields. Press the arrow near *Table* to select your cube. Notice that in the *Select table* window there is also table which you added to *metabase* in the first steps of the previous section. Select columns and statistic by pressing the arrows near the *proper fields*. You can test your report by pressing the *Test* button. When you have filled in all field press *OK*.

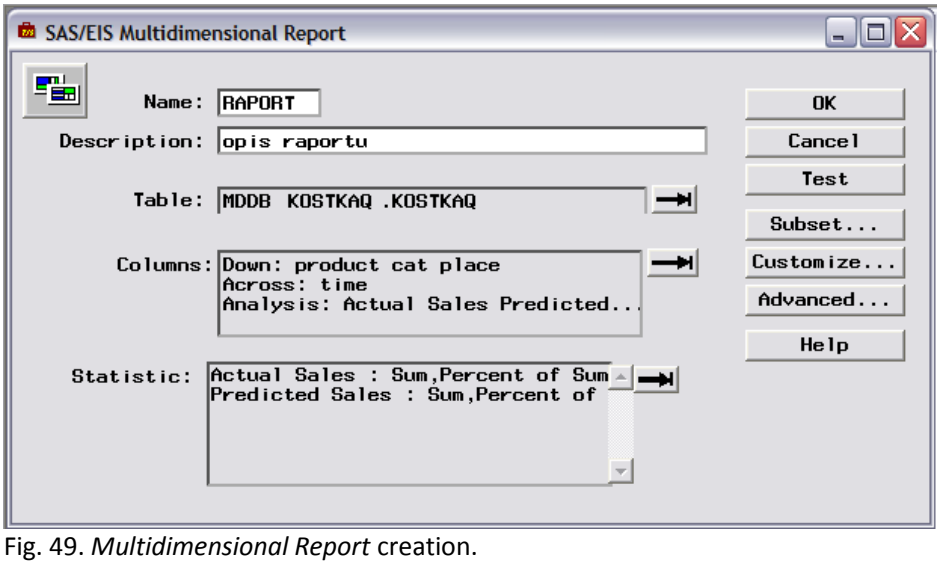
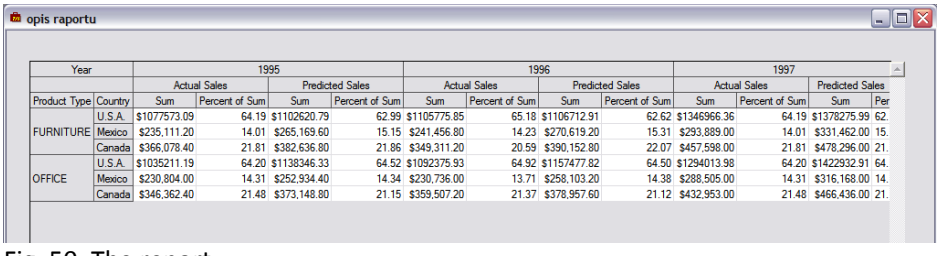


Fig. 49. *Multidimensional Report* creation.

3. And you have created a *Multidimensional Report*. Now you can drill down (or up) your hierarchical values by doubleclicking their names.



Year		1995				1996				1997			
		Actual Sales		Predicted Sales		Actual Sales		Predicted Sales		Actual Sales		Predicted Sales	
Product Type	Country	Sum	Percent of Sum	Sum	Percent of Sum	Sum	Percent of Sum	Sum	Percent of Sum	Sum	Percent of Sum	Sum	Per
FURNITURE	U.S.A.	$1077573.09	64.19	$1102620.79	62.99	$1105775.85	65.18	$1106712.91	62.62	$1346966.36	64.19	$1378275.99	62.
	Mexico	$235,111.20	14.01	$265,169.60	15.15	$241,456.80	14.23	$270,619.20	15.31	$293,889.00	14.01	$331,462.00	15.
	Canada	$366,078.40	21.81	$382,636.80	21.86	$349,311.20	20.59	$390,152.80	22.07	$457,598.00	21.81	$478,296.00	21.
OFFICE	U.S.A.	$1035211.19	64.20	$1138346.33	64.52	$1092375.93	64.92	$1157477.82	64.50	$1294013.98	64.20	$1422932.91	64.
	Mexico	$230,804.00	14.31	$252,934.40	14.34	$230,736.00	13.71	$258,103.20	14.38	$288,505.00	14.31	$316,168.00	14.
	Canada	$346,362.40	21.48	$373,148.80	21.15	$359,507.20	21.37	$378,957.60	21.12	$432,953.00	21.48	$466,436.00	21.

Fig. 50. The report.

Browsing MDDB – charts

1. From *EIS Main Menu* choose *Report Gallery* and then *Advanced Graphic Library*. You can see the *Advanced Graphic Library* window. Drag your *MDDB* from your library (explorer's window) and drop to the charts window. Next choose the type of chart and drop your cube on the icon.

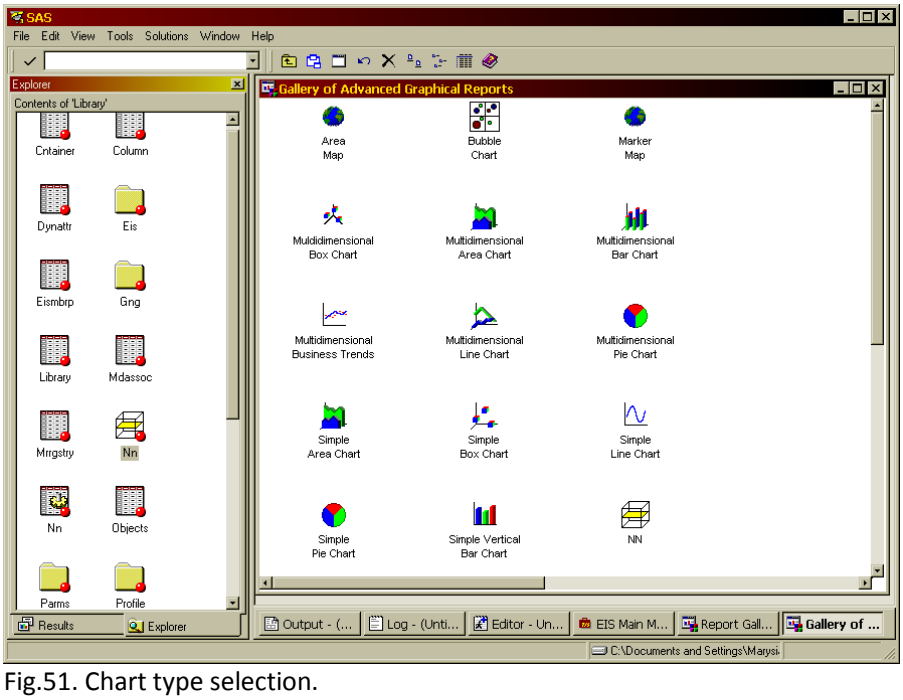


Fig.51. Chart type selection.

2. Choose dimensions and measures that you want to show on the chart. For different chart types a different number of dimension and measures could be available. When you choose everything click *OK* to create the chart.

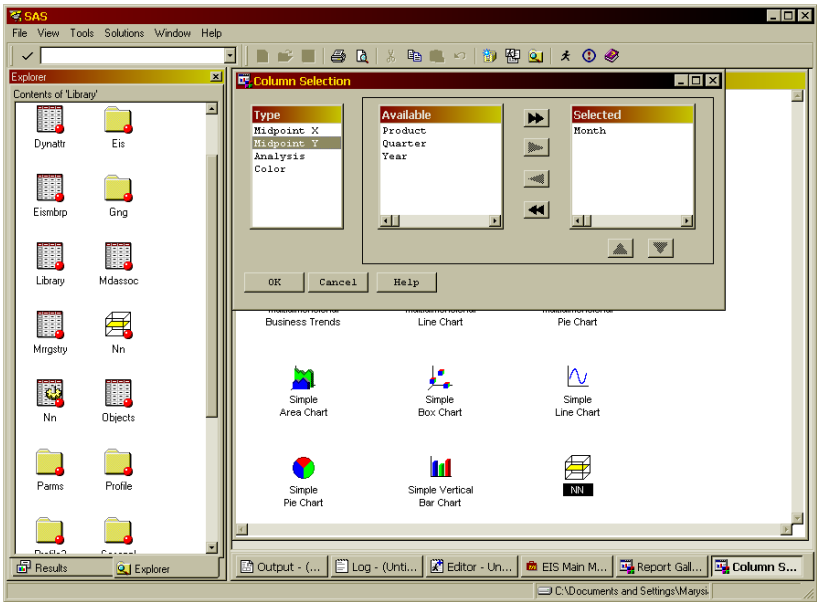


Fig. 52. Columns selection.

3. And finally you have created the chart. When you right-click on the chart you will see a pop-up menu with chart options. You can change the layout, colors and many more attributes of your chart. If you use hierarchical values to create you can now drill down (up) these attributes to receive more or less specific data.

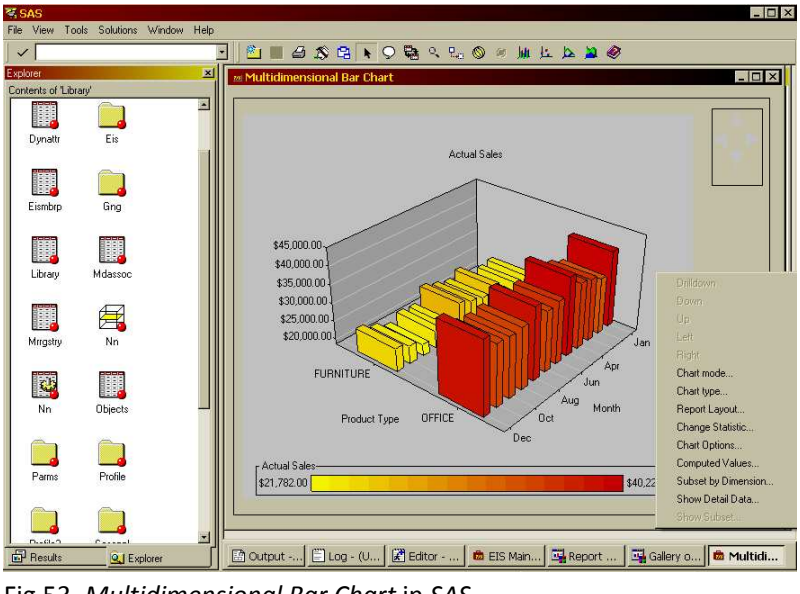


Fig.53. *Multidimensional Bar Chart* in *SAS*.

References and Additional Information

1. SAS online documentation:
   http://rush.ict.pwr.wroc.pl/sasdoc/sashtml/onldoc.html
2. Tutorials:
   http://www.sas.com/apps/elearning/elearning_courses.jsp?cat=Free+Tutorials - you need to register (for free) and log-in *add to card*
3. SAS home page: http://www.sas.com
4. SAS Tips and Techniques:
   http://www.amadeus.co.uk/sas-technical-services/tips-and-techniques
5. Data Analysis using the SAS Language:
   http://en.wikiversity.org/wiki/Data_Analysis_using_the_SAS_Language
6. SAS Customer Support: http://support.sas.com
7. Installing, Configuring, and Migrating to SAS® 9.2:
   http://support.sas.com/software/92/deployment.html

Task List 5. Introduction to SAS and 4GL

1. General Task List Information

In this task list students will be introduced to the *SAS* environment and *4GL* language.

2. Schedule

The report containing the solution prepared by students should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L05.*

3. Tasks – The (*) symbol denotes optional items. Prepare screenshots with description for all steps.

    1. Find the appropriate data set – the regular one is the library data (in the dBase format); it is published on the e-learning platform available for the

course, e.g. *Stopka 3*. Another data set will be rewarded with the additional score.

2. Create your own library using your surname. A library is equivalent to a folder. Unlike all others, the library *work* is temporal - it used to be erased at launching and closing the system.
3. Import your source data set into the SAS environment. In the case of the library data set, import from DBF format and use *reader* (*czytelnicy*) and *loan* (*wypożyczenia*) data. **Work by adding new issues to a single *data step*. Preserve the final *data step*, e.g. in your personal folder. In order to launch a piece of code, mark it and press F8 (or the appropriate icon).**
4. Observe the structure of your data (*proc contents*). In the case of library data use e.g. the *reader*s database. Columns (*fields*) in *SAS* are called *variables*. Rows (*records*) are called *observations*.
5. In the case of *library data*, use only observations (*rows*) with a non-empty *pesel* identity number (variable/field *xpesel*) for further processing. Derive some additional variables from others, e.g. from *xpesel* create two new: year and month of the date of birth; first two digits and the following two digits, respectively. Estimate the age of a *reader*.

```
Data step readers2;   /* the final set (it creates
data set readers2); If you use the same name as your
source data (czyt), the system overwrites its
previous version  */

set czyt;    /* adds source set "czyt" to data set
*/

where not (xpesel is missing); /* only observations
with non-empty pesel */

year=...;

age=111-year;       /* creates new variable "age" in
the data set */

month=...;          /* adds a new variable month to
final data set */

keep iden1 month age year;   /* preserve only useful
```

```
variables */

run;
```

Change the type of the *month* variable from character to numeric by multiplying by 1 (not the best method) or by using *input* function. Instead of *111–year* use another statement with better results (*if*) or something else.

In the case of other source database, perform some adequate processing.

6. Exploit other parameters of *where* e.g. *contains*, *between … and …*, *in (…)*. You can *drop* or *keep* variables.
7. Prepare a report (*DW_FullName_L05.*DOCX/PDF document) containing documentation from the exercises, including screenshots, 4GL codes, SAS system logs and description of analytical data used (in case of data other than library).

4. Presentation

During classes, each student will have 5 minutes to present their task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of the techniques.

Task List 6. Analysing data using SAS tools

1. General Task List Information

In this task list, students will learn how to perform basic statistical analysis of the data and how to cleanse the data.

2. Schedule

The report with the student solutions should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L06.*

3. Tasks – The (*) symbol denotes optional items. Prepare screenshots with description for all steps.

    1. Make some statistics on your data (*proc means*). For library data set, it should be at least:
        a) By the year of birth.
        b) What is the top month of birth?
        c) What is the average, maximum, minimum, and standard deviation of the readers' age?
        d) Remove readers with wrong month or year of birth.
        e) Save in a new set all months of year with the appropriate number of readers born

```
proc means;

by month;

output out=my_output mean
(month)=month;
```

    2. **Sorting**. Select two tables to join. Sort each of them by the appropriate variable (key), e.g. in the case of library data:

```
proc sort data=my_library.sourceData
out=my_library.outputData;

     by keyVariable1 keyVariable2;

run;
```

The `data` parameter is *Data Set*, which can be either a concrete external file on the hard disk or a view – a logical set of data obtained from the output of the certain code.

3. In the case of library data, calculate the number of loans for each reader (*means*). Join this information with the reader data (*merge*). What is the age of the most active reader?
   Use `merge` for joins:

```
data work.my_result;

merge my_library.first_set
my_library.second_set;

      by my_id_variable;

run;
```

   In case of other source database, perform the adequate processing.

4. How procedure `freq` can be useful? It delivers the set of values for the sorted source *data set*.
5. (*) In the case of library data attach the gender data (the separate DBF file) and create some reasonable statistics (an optional item).
6. (*) Give some reasonable examples for procedures `tabulate` and `univariate`.
7. (*) What do you suggest to clean your data? (an optional item)
8. (*) Imagine you need to create a new variable which values are calculated from other observation or observations (not only from the currently processed one). How can you reach it? Give an example for your data. (an optional item)
9. (*) How can the standard SQL queries be invoked in the SAS environment? Give some examples for your data (at least 3). (an optional item)
10. (*) Give an example for the proper usage of `SYMPUT` and `PUT` statements
11. Prepare a report (*DW_FullName_L06.*DOCX/PDF format) containing documentation from the exercises, including screenshots, 4GL codes, SAS system logs, statistics, descriptions of own solutions, conclusions and other remarks related to the subject.

4. Presentation

During classes, each student will have 5 minutes to present the task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list.

Task List 7. The idea of multidimensional databases – cubes in SAS

1. General Task List Information

In this task list, students will learn how to create cubes in SAS.

2. Schedule

The report containing the students' solutions should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L07.*

3. Tasks – The (*) symbol denotes optional items. Prepare screenshots with description for all steps.

    a. Create some new reasonable hierarchical variables derived from other ones. In the case of the standard library data these can be the following variables: day, month and year of loan (*datawyp*) and return date (*datazwr*); functions *day*, *month*, *year*, and expected as well as real loan period, e.g.:

```
data my_lib.loans_new;

/* the source is the loan data set (table) -
wypożyczenia, with non-empty observations. Can you the
difference compared to the previous task list?  */

  set my_lib.loan (where=(not (datawyp = . or datazwr =
.)));

  loan_year = year (datawyp);   /* new variable */

  loan_month = month (datawyp);

  loan_day = ...

  ...    /* ...and similarly for the return date
(datazwr) */
```

```
  loan_period = datazwr - datawyp;

  expected_period = termzwr - datawyp;

    /* the number of days exceeded. Return before the
deadline - 0 */

  if loan_period - expected_period > 0

               then overdraft = loan_period -
expected_period;

               else overdraft = 0;

run;
```

Note that another, non-standard data set will be rewarded with an additional point. To the report attach the listing of the 4GL source code.

b. Merge the obtained loan data set with reader data (in the case of library data set). Keep only valuable variables (*keep*), e.g. fine, faculty, reader's age. This way you get the input data set for the multidimensional database (*MDDB*). Attach to the report the listing of the *4GL* source code.
c. (*) Suggest your own interesting and new variables, reasonable for data you have chosen. Attach your source code and the description of new variables to your report. (an optional item). To the report attach the listing of the *4GL* source code.
d. Create the multidimensional database, in the case of standard library data by using **proc MDDB**:

```
   proc mddb data=my_lib.my_source
   out=my_library.my_output;

      /* dimensions and measures i.e. all used
variables */

     class loan_day loan_month loan_year ... fine
   reminder loan_period,
            expected_period overdraft ...;
```

```
    /* dimension hierarchies */

  hierarchy loan_year loan_month loan_day
/name='Loan date';

  hierarchy return_year return_month return_day
/name='Return date';

      /* Measures (numeric values) */

  var loan_period /n nmiss sum uss min max;

  var fine /n sum;

  var expected_period;

run;
```

Attach to the report the listing of the *4GL* source code.

e. Prepare a report (*DW_FullName_L07.*DOCX/PDF format) containing documentation from the exercises, including screenshots, listings of *4GL* code, *SAS system logs*, statistics, descriptions of own solutions, conclusions and other remarks related to the subject.

4. Presentation

During classes, each student will have 5 minutes to present the task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list.

Task List 8. Metadata, OLAP reports and charts in SAS

1. General Task List Information

In this task list, students will learn how to create metadata for *MDDB* using *EIS/OLAP Application Builder* and *OLAP* reports and charts.

2. Schedule

The report with the students' solutions should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L08.*

3. Tasks – the (*) symbol denotes optional items. Prepare screenshots with description for all steps. To complete task listed below you can use *MDDB* prepared during Task List 7.

    a) Create the appropriate metadata for your *MDDB* using *EIS/OLAP Application Builder*. Do not forget hierarchies for your dimensions, e.g. year-month-day.

    b) (*) Do you know another way to create a *MDDB* without `proc mddb` (an optional item)?

    c) (*) What other hierarchies have you proposed? (an optional item)

    d) What is the content of the obtained database. Change the layout (e.g. with the right mouse button) by pivoting i.e. the exchange rows and columns. Consider aggregations.

    e) Create some sensible *OLAP* charts for your data (at least 5), e.g. by drag and drop a *DB* onto one of charts available, *Reporting -> EIS/OLAP report gallery -> e.g. Advanced Graphical Reports.* The charts should provide different knowledge. Show the drill-down, roll-up and selection options for hierarchies. Change the layout of your charts (also the format of presented values). Save one of your charts in the JPEG format and add some other graphical objects (e.g. arrows with description for most important chart pieces) using an external graphical toll. What are your conclusions drawn from your charts?

Create one flat and one hierarchical report (in the form of a tree). This is the most significant item in the list. Every picture (screenshot) should have a caption that explains the meaning of its axis.

f) (*) Are you able to create any 4-, 5- or 6-dimensional charts? Provide an explanation (an optional item)

g) What probable conclusions (organizational, sociological, financial, etc.) can be drawn from your analyses?

h) (*) Is it possible to provide any rules for matching the kind of chart to data type? (an optional item)

i) (*) Create some measures which will be calculated on the fly (online). How have you achieved them? (an optional item)

j) (*) Are you able to merge reader data with the map of Poland available in *SAS* (*GIS*)? This item is valid only for the standard library data source. (an optional item)

k) (*) Note that you can exploit other source data instead of the regular one. You will get an additional point.

l) Prepare a report (*DW_FullName_L08.*DOCX/PDF format) containing documentation from the exercises, including screenshots, listing of *4GL* codes, *SAS system logs*, statistics, descriptions of own solutions, conclusions and other remarks related to the subject. Add your opinion about the *SAS technology*, point out advantages and disadvantages. Compare the *SAS tools* with *Microsoft tools*. Which one, in your opinion, is better – justify your choice.

4. Presentation

During classes, each student will have 5 minutes to present the task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list.

Part III. Project

Task List 9. Choose your Tool with justification and prepare data set for analysis

1. General Task List Information

In this task list students will have to choose the tool (*Microsoft*, *SAS* or another) for the rest of the classes. They also have to browse for and prepare a dataset (data cleansing process) with which they will work while preparing the following task lists (from 10 to 13).

2. Schedule

The report containing the solution worked out by students should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which task list ought to be presented. Email subject: *DW_FullName_L09.*

3. Tasks – The (*) symbol denotes optional items

    a. Based on your experience from the previous task lists choose tools with which you will accomplish the project. Please read and analyze the task lists 9 to 13 before you make your final decision. In the report justify your decision.

    b. Browse for a complex dataset for your project. Complexity of the dataset will affect the final grade. You can, for example, analyze your phone billings, your invoices, history of your activities on various web sites, logs from your own computer etc. In the report describe your dataset.

    c. Analyze your dataset and describe the data cleansing process you will have to perform. What new data you can extract, which data have to be cleansed and which can be aggregate?

    d. (*) Propose at least 2 different cubes you are planning to design.

e. (*) Propose at least five different and useful measures and 10 different and useful dimensions you will be able to design for your cubes.

f. Prepare a report (*DW_FullName_L09.*DOCX/PDF document) containing description of all above items.

4. Presentation

During classes, each student will have 5 minutes to present the task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list.

Task List 10. ETL process – transfer data to database, derived variables

1. General Task List Information

In this task list students will have to prepare an *ETL process* which will transfer their dataset from raw data files to *OLTP database*.

2. Schedule

The report containing the solution worked out by students should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L10.*

3. Tasks – the (*) symbol denotes optional items. Document all steps with screenshots and description.

    a. Prepare the *ETL process* - Extraction, Transformation, Loading.

        i. Extraction – extract data from your dataset and assign appropriate datatypes.

        ii. Transformation – transform data to appropriate data types, modify if needed and apply your cleansing process described in Task List 9.

        iii. Transformation – extract new data – derived columns, derived variables.

        iv. Loading – design the *OLTP database* which will be able to store your transformed data. Next, load this data to the *OLTP database*.

    b. Prepare a report (*DW_FullName_L10.*DOCX/PDF document) containing description of all above items. Include in the report screenshots and description of all transformations and the *OLTP database*. Attach also listings of all source code you have produced while working on this task list.

4. Presentation

During classes, each student will have 5 minutes to present the task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list.

Task List 11. Cubes, measures, dimensions

1. General Task List Information

In this task list student will have to design cube measures, cube dimensions and finally the cubes.

2. Schedule

The report containing the solution worked out by students should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L11.*

3. Tasks – the (*) symbol denotes optional items. Document all steps with screenshots and description. Design at least 3 cubes; for each of them:

    a) Define a fact table, measures (including calculated measure) and dimensions for the cube. It also covers hierarchies of dimensions.

    b) Define the method of storing data (*MOLAP, ROLAP, HOLAP*). Justify your choice.

    c) Consider different aggregation options and justify your choices. Analyze granularity and the possibility of data explosion.

    d) Create/process the cube.

    e) Prepare a report (*DW_FullName_L11*.DOCX/PDF document) containing description of all above items. Include in the report screenshots and description of your cubes, measures and dimensions (with reasonable justification why you have created them). Attach also listings of all source code you have produced while working on this task list.

4. Presentation

During classes, each student will have 5 minutes to present the task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list.

Task List 12. Reports and Charts – selection and adjustment, cubes efficiency issues

1. General Task List Information

In this task list students will have to analyze their cubes, optimize them and finally prepare reasonable reports and charts.

2. Schedule

The report containing the solution worked out by students should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L12.*

3. Tasks – the (*) symbol denotes optional items. Document all steps with screenshots and description.

    a) Optimize the cubes. In the report, describe and justify each optimization.

    b) Prepare 5 reports and 5 charts for each cube. Select only reasonable charts and reports, i.e. those that can provide potentially useful knowledge. Adjust their type (layout) to the nature of the presented values, e.g. a pie chart for percentages instead of a bar or linear chart.

    c) Analyze your reports and charts. Add conclusions, interpretations and usability of charts to the report.

    d) Prepare a report (*DW_FullName_L12.*DOCX/PDF document) containing description of all above items. Include in the report screenshots, description and interpretation of all reports and charts. Attach also listings of all source code you have produced while working on this task list.

4. Presentation

During classes, each student will have 5 minutes to present the task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list.

Task List 13. Processing and accessing your cube

1. General Task List Information

In this task list students will have to design their own application which will automatically process cubes and allow accessing data in cubes.

2. Schedule

The report containing the solution worked out by students and the application source code should be sent via email to the supervisor by the last Monday (6:00 AM CET) before the classes during which the task list ought to be presented. Email subject: *DW_FullName_L13.*

3. Tasks – the (*) symbol denotes optional items.

    a) Prepare an application (web or desktop any programming language allowed) which will be able to process your cubes.

    b) Add to your application new functionality which will allow accessing your cubes, execute a query and present the results.

    c) Prepare a final report (*DW_FullName_L13.*DOCX/PDF document) containing extended and corrected reports from the task lists 9–12. Add screenshots and description of designed application.

4. Presentation

During classes, each student will have 5 minutes to present the task list to the supervisor. During presentation of the task list students should describe their own solutions and prove the knowledge of techniques described in the task list.