Data Warehousing Assignment—Part III

Toon Calders Rohit Kumar

Deadline: 18 December 2015

1 Practical information

Deadline:	18 December 2015
Assignment package:	`https://www.dropbox.com/s/slca7pxhu69uiab/assignment_material.zip?dl=0`
Groups:	Same groups as for assignment parts I and II
How to submit:	Upload solution at `uv.ulb.ac.be`

Please read this complete document carefully before starting your assignment. In case of doubts, after carefully reading the document, do not hesitate to contact the lecturer. The complexity of the assignment is intended to be in the execution of it, not in its interpretation. Please note that Appendix B contains several hints that may be helpful in the execution of the assignment.

2 Objectives

The goals of this assignment are:

1. Creating ETL scripts for updating the database in SSIS.

For the assignment the 2008R2 or any later edition of the SQL Server tools must be used.

3 Assignment

The assignment part III starts were assignment II ended. In order to make sure that all student groups can start from the same starting point, a model solution for part II has been provided in the assignment package[1]. The name of the script is "initial_load.dtsx". This script uses as input the database "dw2015.bak", also included into the package, and is also available on the database server CS-MSSQL under the name "dw2015". This database corresponds to snapshot 1.

In order to test the benefits of the data warehouse, the company is asking for a proof of concept, based on 9 historical snapshots of the database that can be taken from backup. Your task is now to define a SSIS package to incrementally load these snapshots into the data warehouse. In the assignment package you will find a few snapshots (in directory "snapshots"; numbered 2 to 10), and a script called "skeleton.dtsx." This script will load the snapshots one by one. The snapshots represent the state of the database at one point in time. As such, next to new data, the snapshots will also contain customers, accounts, branches, transactions, etc. that were already in earlier snapshots or even in the original load. Therefore, in order to correctly update the content of the datawarehouse based on a snapshot, you first need to figure out what has changed since the last snapshot. In order to facilitate this operation, in the (larger) tables Customer and Account an additional attribute "lastupdated" has been added that contains the last date on which the tuple was changed. Table transaction already has a "date" attribute indicating when the transaction was entered. Furthermore, every snapshot has a table "snapshot_time" containing one tuple with the start (attribute "from") and end (attribute "to") time of the snapshot. These values can be used to correctly set the valid times for the versioned dimensions in the data warehouse and to test which tuples are new. It is guaranteed that all customers, accounts, and transactions

[1]Notice, however, that there are two deviations from the original assignment in the script: (1) the balance in no longer present in the factTransaction, and (2) at the end of the script an update query is executed to correct some data errors in snapshot 1. The package is available from `https://www.dropbox.com/s/slca7pxhu69uiab/assignment_material.zip?dl=0`

that are newly inserted or underwent changes since the last snapshot, will have respectively lastupdated and date in the interval $[from, to]$. This does **not** mean, however, that the other (smaller) tables, such as for instance "Branch" or the geography related tables, will not change. These tables may change as well, even though they do not have a lastupdated attribute. The schema of the snapshot databases is available in Figure 1.

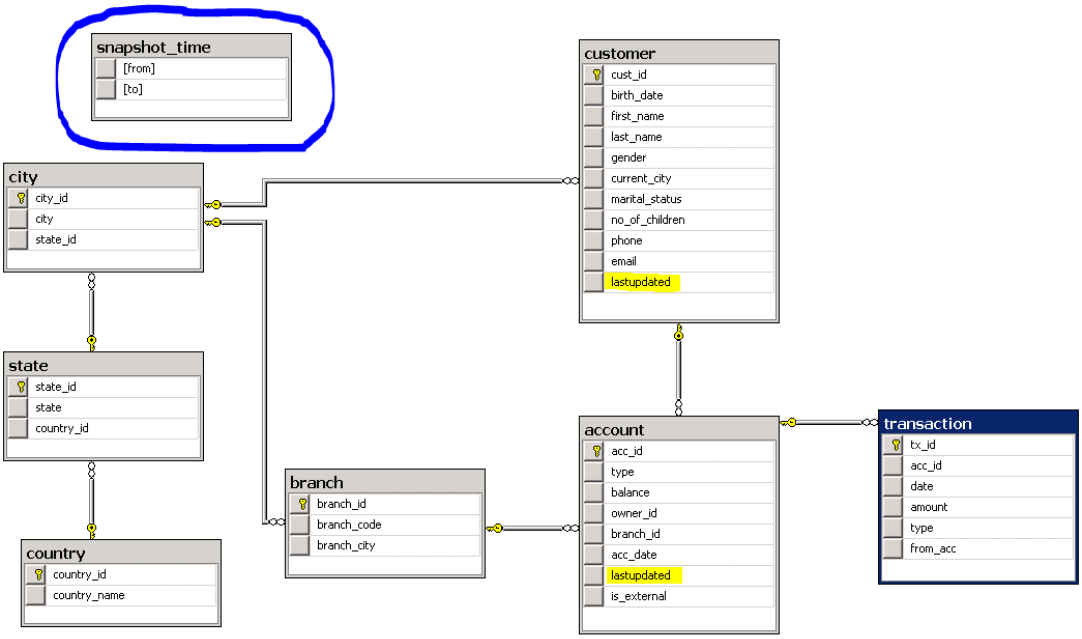


Figure 1: Schema of the snapshot databases

4 Obtaining the Data

All assignment data is available in the assignment package[2]. This package contains the following files:

1. The solution to assignment part II;
   (a) dw2015.bak : backup file of the database represetning snapshot 1; input for the initial load script;
   (b) initial_load.dtsx : package for loading the first snapshot into the data warehouse;
   (c) two csv files date.csv and time.csv used by the initial loading script.
2. The snapshots for assignment part III
   (a) Directory snapshots contains 9 sub-directories snapshot2 till snapshot10. These directories on their turn contain the table contents in .csv format. You do not need to worry about how to deal with this data; just copy the whole snapshot directory to your hard disk. The skeleton package will deal with loading it (next point).
3. The skeleton to be used for the assignment part III:
   (a) skeleton.dtsx : the skeleton for your solution (Figure 2). Only add components in the sequence container named "Your solution."

Appendix A contains step-by-step instructions on how to get all data and setup the script.

[2] `https://www.dropbox.com/s/slca7pxhu69uiab/assignment_material.zip?dl=0`

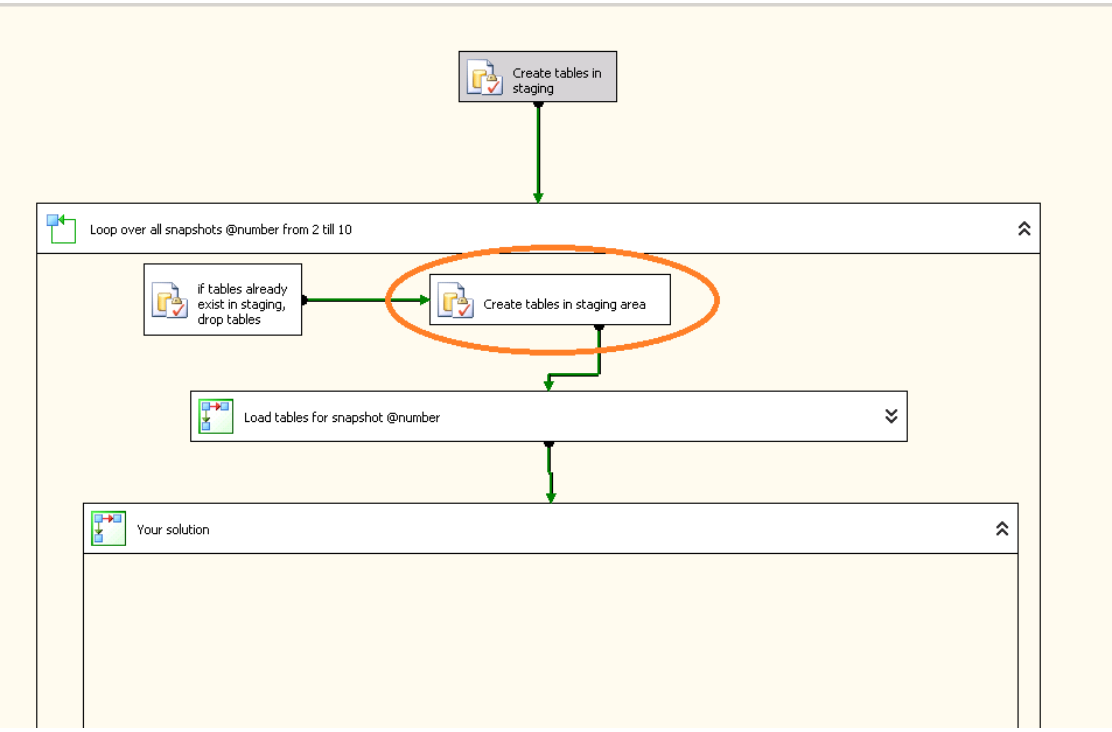


Figure 2: Control flow of the skeleton package. The for-loop container loops over all snapshots. Your ETL script goes inside the sequence container names "Your solution."

5 Deliverables

You should deliver the following elements.

1. A report (as a.pdf), containing (length indication is purely indicative):
   (a) A **cover page** with the list of group members, including student ID,
   (b) Figures showing all your data flows and control flow with a succinct explanation whenever needed (length depends on your ETL flow)
2. The ETL package for the incremental load of the data into the data warehouse. The package should be based on the helper package "skeleton.dtsx", in the sense that it should load the snapshots one by one and the package should end when all snapshots have been loaded.

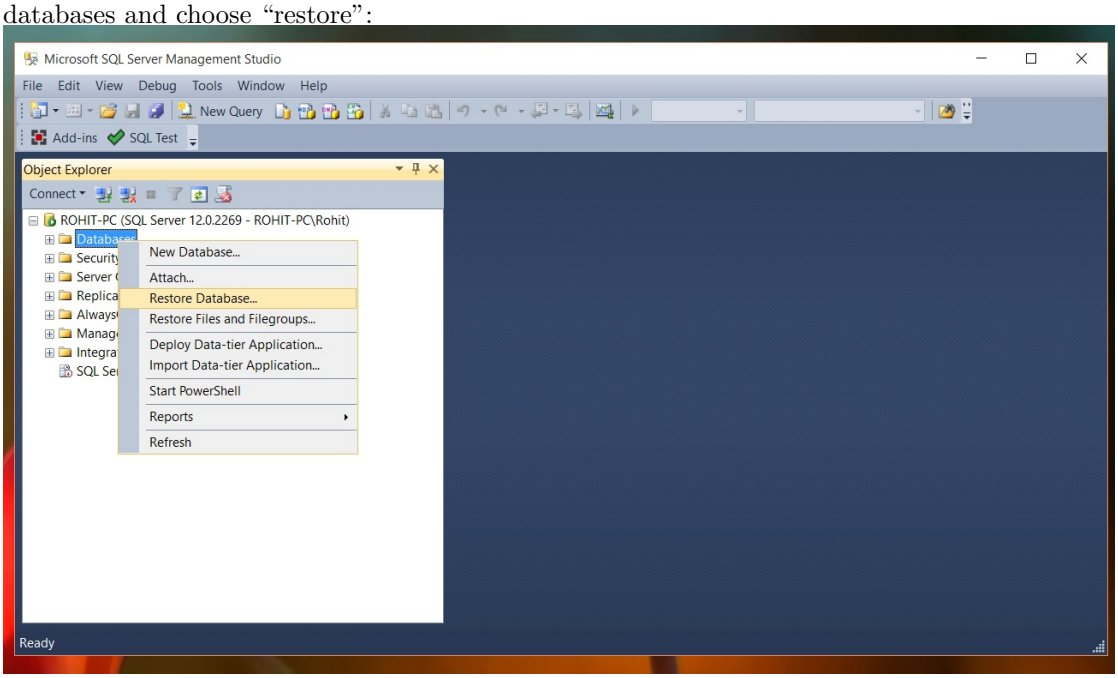
Submit both files as a single .zip or .rar file on the université virtuelle course website.

A Step-by-Step Instructions

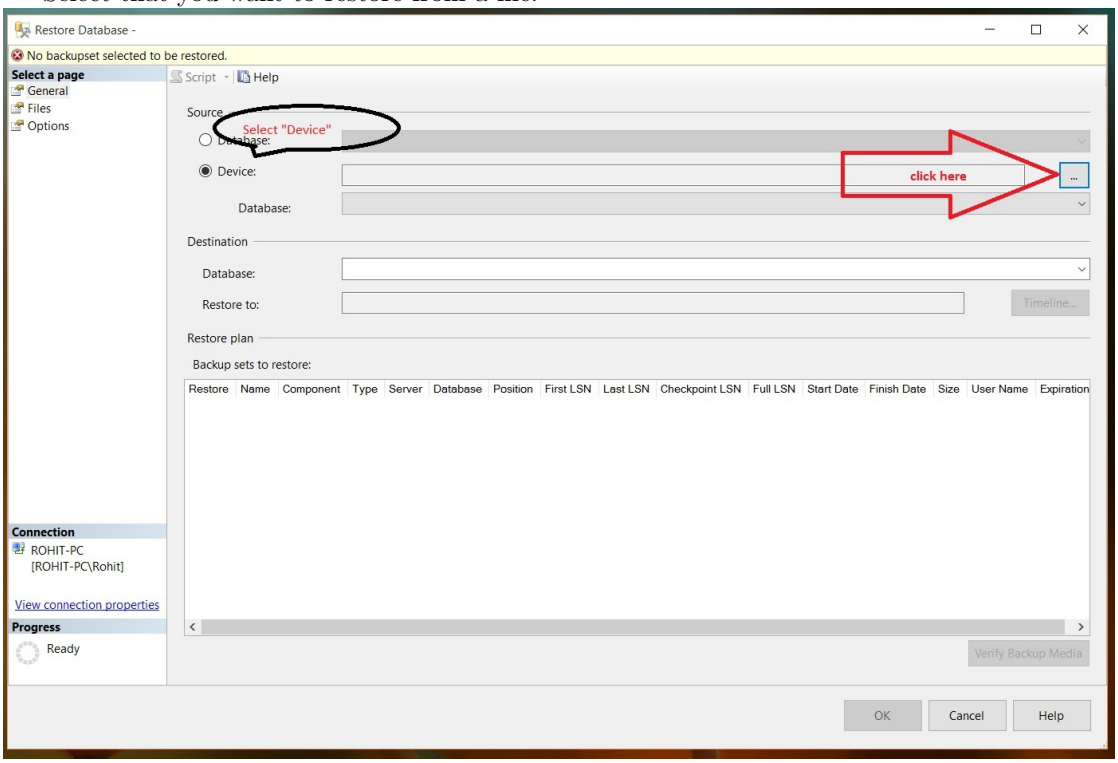
It is essential that you are able to execute the steps below correctly. Please test these steps ASAP and contact the lecturer should there be any problem.

A.1 Obtain the Snapshot 1 database

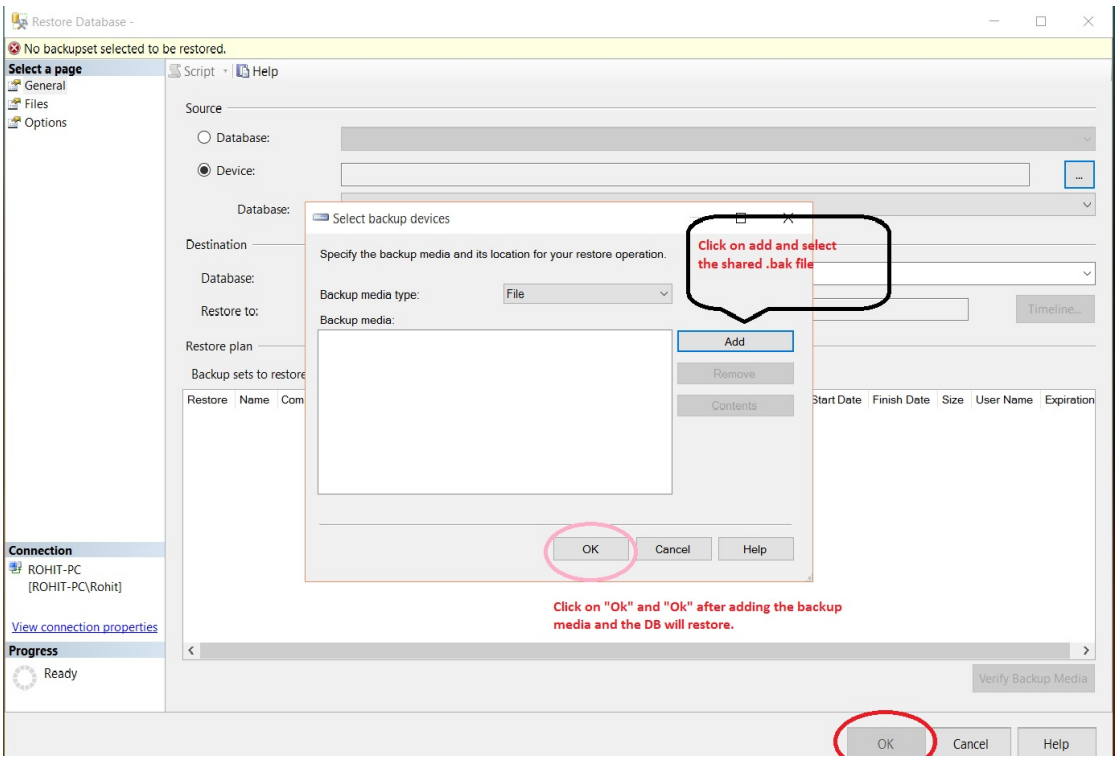
In principle you already have this database from the assignment part II. This database is also available on the database server cs-mssql used in the labs under the name "dw2015." If you are creating your solution in the lab you can directly use this database because you only have to read from it, never write to it. In case you wish to obtain this database in your own system, follow these steps: first copy the file dw2015.bak to a folder in your C drive (The follow the below steps to restore from this backup file). Then, open SQL Server management studio. Right-click on databases and choose "restore":



Select that you want to restore from a file:

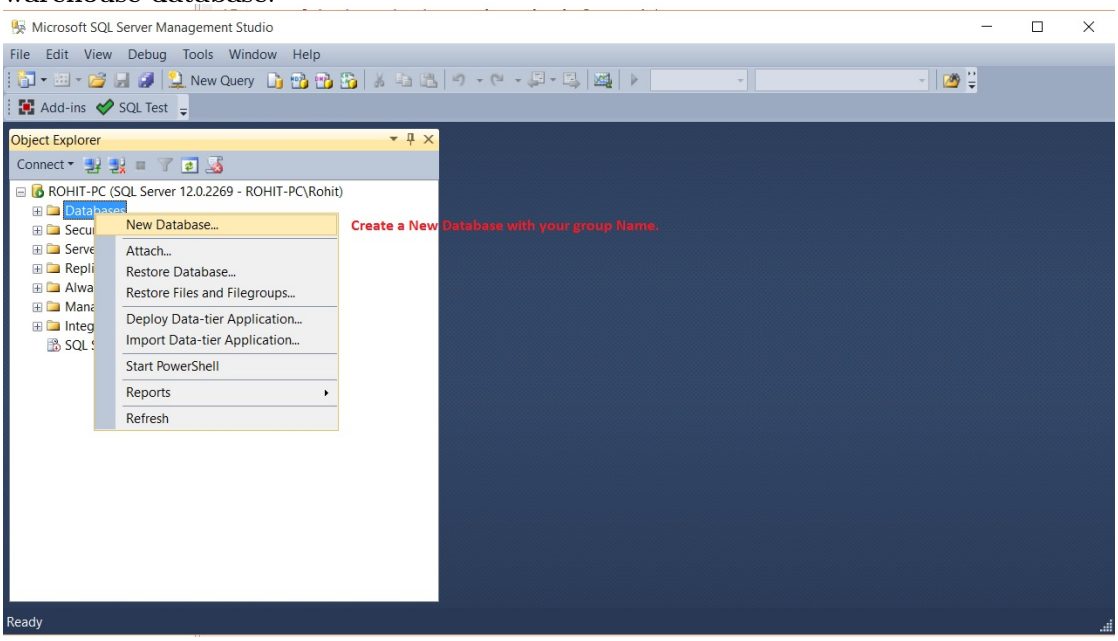


and navigate to the file:



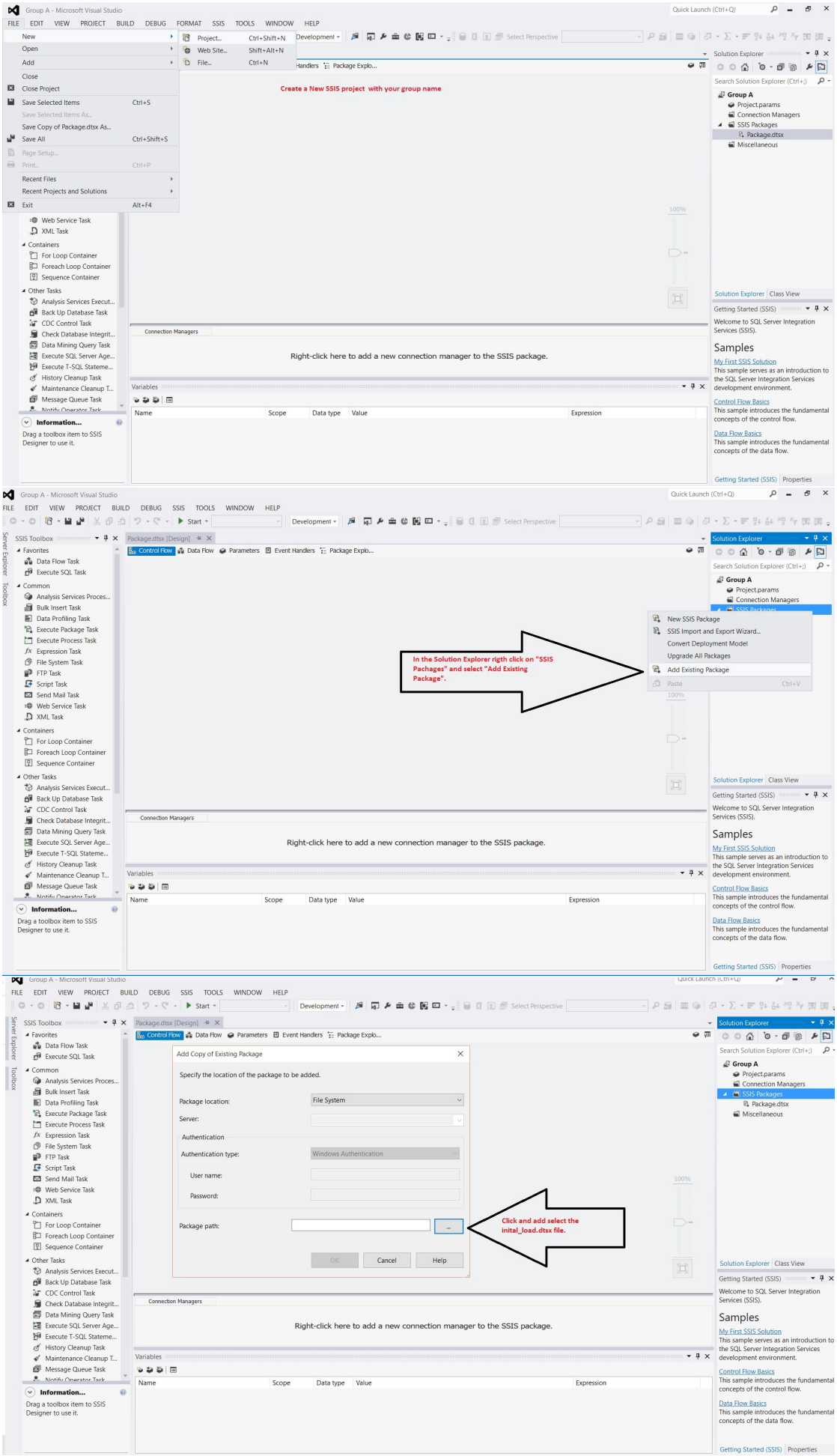
A.2 Creating the database that will hold your data warehouse

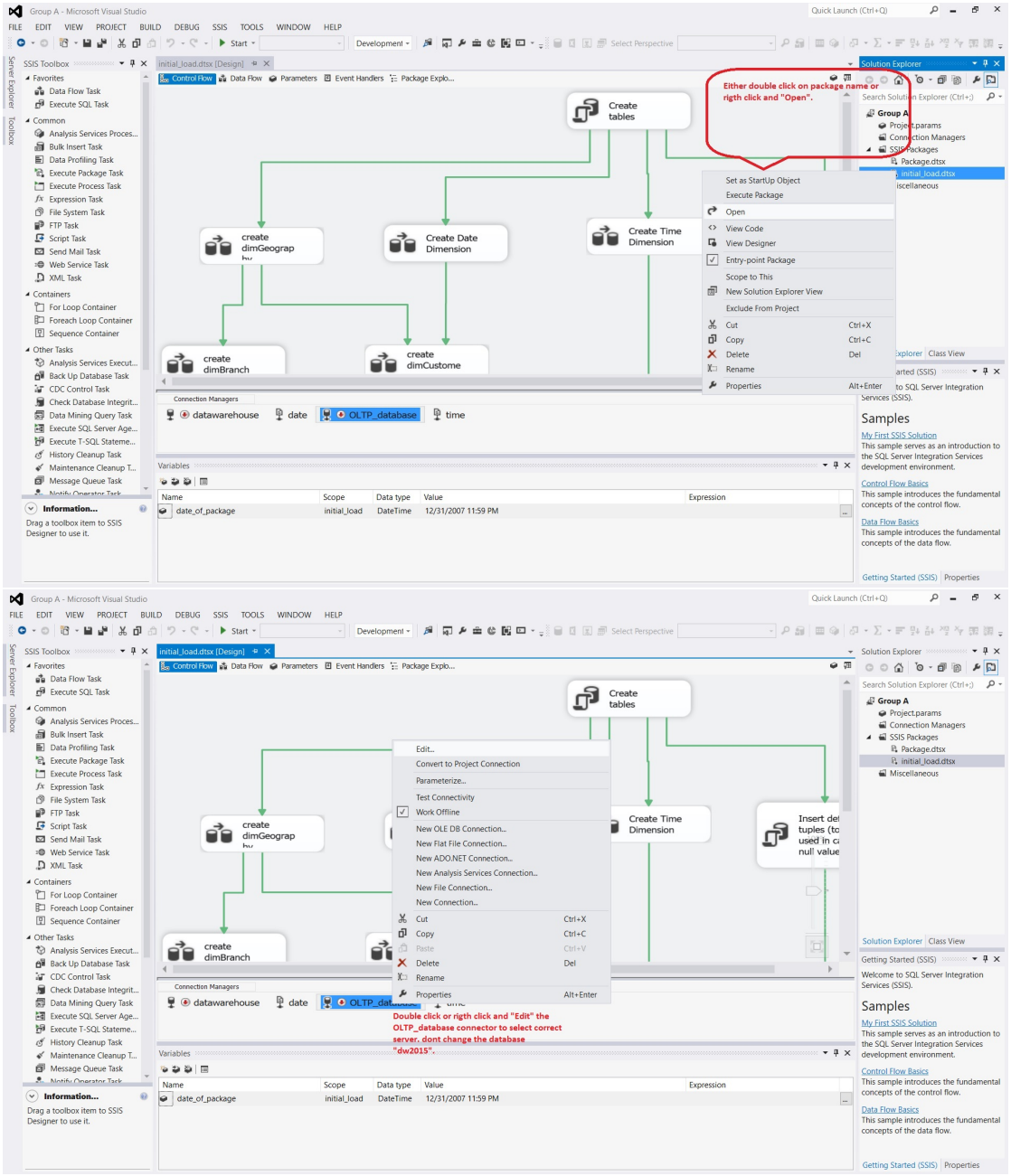
Create a new database with your group name; this database will contain your data warehouse. You can use the same database here as the one you used for part II of your assignment, or you can create a new one; that's up to you. Do realize, however, that the later steps will remove the current content of this database. We will refer to this database in later steps as the **data warehouse database**.

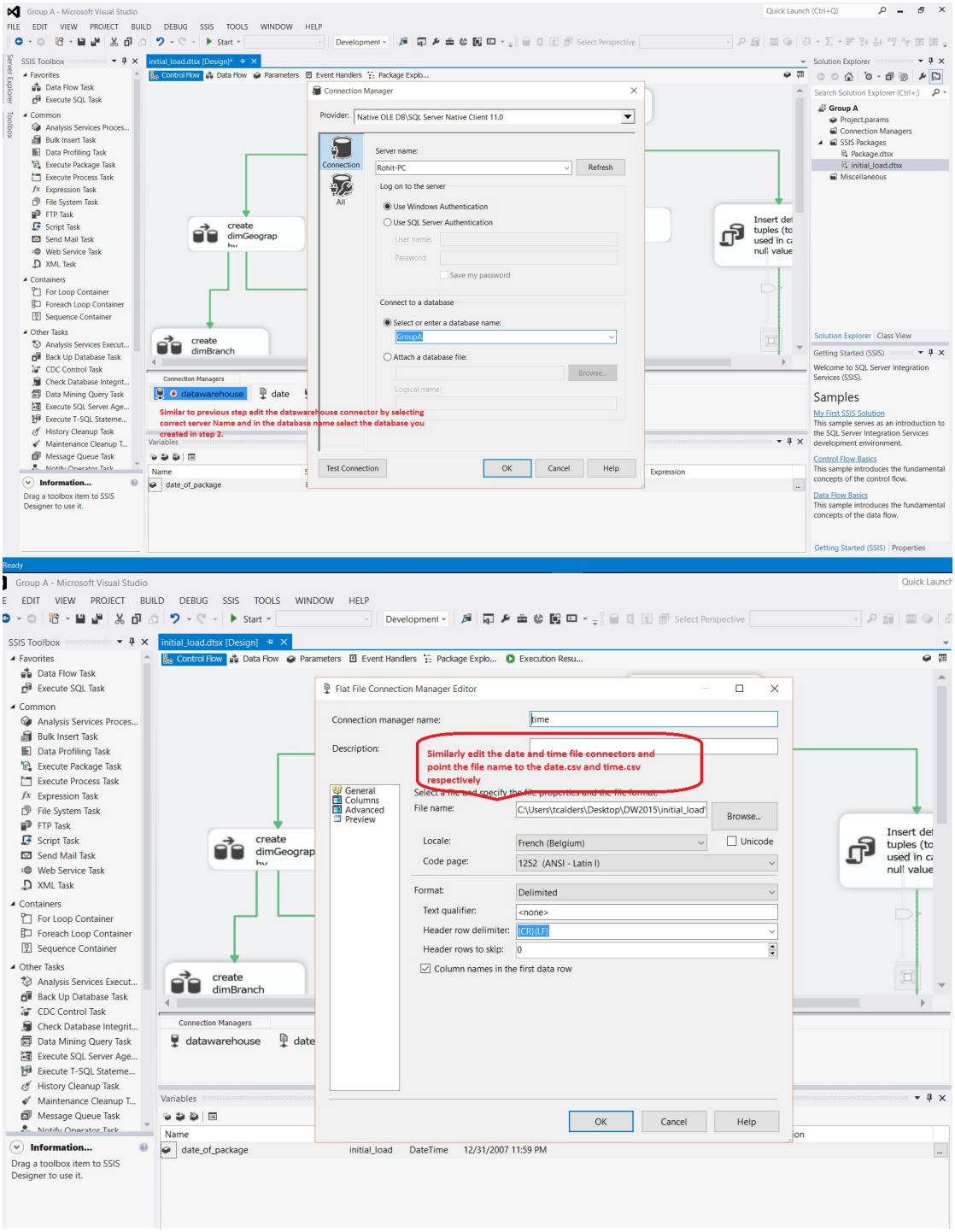


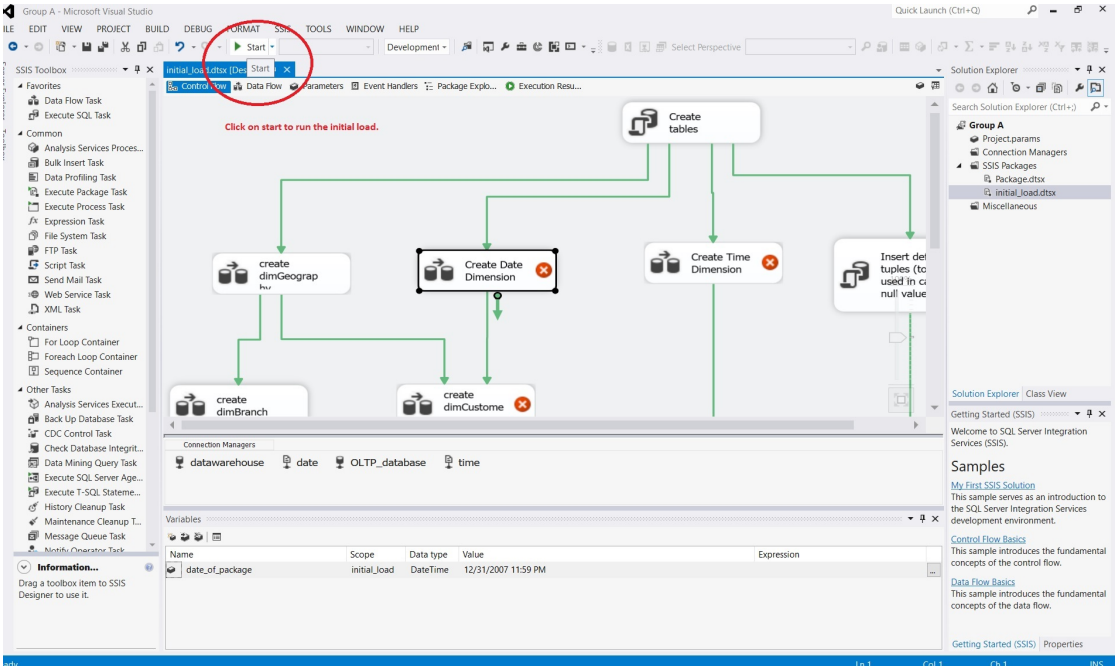
A.3 Loading and Running the initial load script

The following steps describe how to load and execute the initial load script.


Create a New SSIS project with your group name
Right-click here to add a new connection manager to the SSIS package.
In the Solution Explorer rigth click on "SSIS Pachages" and select "Add Existing Package".
Add Copy of Existing Package
Specify the location of the package to be added.
Package location:
File System
Server:
Authentication
Authentication type:
Windows Authentication
User name:
Password:
Package path:
OK
Cancel
Help
Click and add select the inital_load.dtsx file.

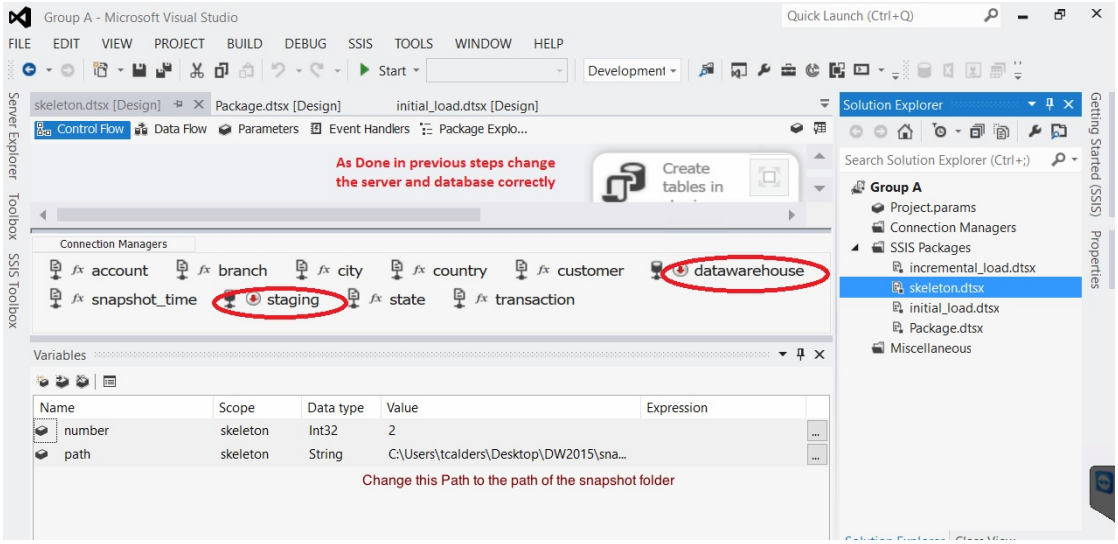

Either double click on package name or rigth click and "Open".
Create tables
create dimGeograp hy
Create Date Dimension
Create Time Dimension
create dimBranch
create dimCustome
Set as StartUp Object
Execute Package
Open
View Code
View Designer
Entry-point Package
Scope to This
New Solution Explorer View
Exclude From Project
Cut Ctrl+X
Copy Ctrl+C
Delete Del
Rename
Properties Alt+Enter
Connection Managers
datawarehouse
date
OLTP_database
time
Variables
Name Scope Data type Value Expression
date_of_package initial_load DateTime 12/31/2007 11:59 PM
Samples
My First SSIS Solution
This sample serves as an introduction to the SQL Server Integration Services development environment.
Control Flow Basics
This sample introduces the fundamental concepts of the control flow.
Data Flow Basics
This sample introduces the fundamental concepts of the data flow.
Edit...
Convert to Project Connection
Parameterize...
Test Connectivity
Work Offline
New OLE DB Connection...
New Flat File Connection...
New ADO.NET Connection...
New Analysis Services Connection...
New File Connection...
New Connection...
Double click or rigth click and "Edit" the OLTP_database connector to select correct server, dont change the database "dw2015".


Connection Manager
Provider: Native OLE DB\SQL Server Native Client 11.0
Server name: Rohit-PC
Refresh
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
Connect to a database
Select or enter a database name: GroupA
Attach a database file:
Test Connection
OK
Cancel
Help
Similar to previous step edit the datawarehouse connector by selecting correct server Name and in the database name select the database you created in step 2.
Flat File Connection Manager Editor
Connection manager name: time
Description:
Similarly edit the date and time file connectors and point the file name to the date.csv and time.csv respectively
File name: C:\Users\tcalders\Desktop\DW2015\initial_load
Browse...
Locale: French (Belgium)
Unicode
Code page: 1252 (ANSI - Latin I)
Format: Delimited
Text qualifier: <none>
Header row delimiter: {CR}{LF}
Header rows to skip: 0
Column names in the first data row
OK
Cancel
Help



A.4 Starting the exercise

Follow now the same steps to load the skeleton. Update the connectors in the same way as you did for the initial load script. Notice that the skeleton uses an additional connector called "staging". This connector refers to the database in which the subsequent snapshots will be loaded. In principle you can set this connector to your datawarehouse database, or create another database for it, following the steps of A.2.



It is important to set the path variable to the correct directory containing the snapshots. The variable "path" should refer to the directory "snapshot" that contains the subfolders "snapshot2" till "snapshot10". You can set the variables as follows: menu SSIS >> variables, or menu View >> Other windows >> Variables. Once you have created the necessary databases, correctly set all connection managers, and the path variable, run the script in order to verify that the snapshots are correctly loaded. The script and files have very carefully been prepared and tested on different versions of SQL Server and operating systems with different regional settings. Nevertheless, it may occur that unforseen compatibility issues arrise. Do not hesitate to contact the lecturer in case of problems. Setting up the environment is not part of the assignment and any help you receive to correctly load and execute the skeleton script will not be taken into account when grading the assignment.

B Hints

- You can execute a single task by right-clicking on it and selecting "Execute task."
- You can disable/enable tasks by right-clicking on it and selecting "enable"/"disable."
- Before starting to create your script, run the create table statements once. In this way you can use the tables and their definitions when configuring the components.
- Keep an eye on the course website; if a question arrives that is of interest to everyone, it will be posted with a response on the message-board of the course web-site (normally you will get an email notification).
- A workflow that worked very well for myself to create the solution for the assignment is the following:
  - Run the initial_load script once to clear the data warehouse and fill it using the first snapshot data (contents of the dw2015.bak database that you restored). The script has been designed in such a way that it will reset the database completely. In this way errors you introduced while creating and debugging your incremental_load script will be cleared;
  - Modify your incremental_load script (the one you are constructing starting from the skeleton), and run it. You can always restrict the snapshots that are loaded by changing the conditions in the container "Loop over all snapshots @number from 2 till 10." If you want to only load snapshot 2 for instance, change the "EvalExpression" to "@number¡=2". Make sure however to reset the value to the original value when submitting the script.
  - After executing the script, check the content of the data warehouse using the SQL Server management studio. The skeleton was designed in such a way that the staging tables will still be present and filled with the contents of the last snapshot that was loaded. The tables are empty only in the next iteration, just before the next snapshot gets loaded.
  - While debugging a script, one can easily get an overview of the data sent over the links in the data flows. To monitor what is sent over the link in the data flow view, double-click on it, and enable the data viewer (how to do it exactly depends from version to version, but it is self-explanatory)