Data Warehousing Assignment—Part II

Toon Calders Hatem Haddad

Deadline: 12 December 2017

1 Practical information

Deadline:	12 December 2017
Group:	Same groups as for assignment Part I
How to submit:	Upload solution at `uv.ulb.ac.be`
Project presentations:	From December 13 to December 22

2 Objectives

The goals of this assignment are:

1. Creating an ETL script for the initial load of the data warehouse.
2. Creating ETL scripts for updating the database in SSIS,
3. Predicting how the size of the data warehouse will grow over time,
4. Deploing a data cube on top of the data warehouse and create a report .

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.

3 Problem Description

In Part I of the assignment you were asked to produce a dimensional fact model based on a textual description of the data warehouse and the operational database for which the data warehouse needed to be built. The constructed dimensional fact model for the data warehouse then needed to be implemented in the relational model. In the second Part of the assignment, we will start from a model solution for Part I and construct an ETL package for the initial load and an ETL package for the incremental load of the data into the data warehouse.

The model solution is given on the course website and will be described below. Notice that there are many different solutions, depending on the interpretation of the description, and certain choices that were made. Therefore, if your solution deviates from the solution given below, this does not necessarily mean that your solution is incorrect. In order to guarantee a homogeneous level of difficulty for Part II of the assignment among the different groups, however, **the starting point for Part II of the assignment is the model solution for Part I**.

The description for Part II of the assignment now is as follows: Make a SSIS package that performs the initial load of the data warehouse. That is, the script should take the database as input and produce a data warehouse reflecting the current state of the operational database. Then, make the ETL package for the incremental load of the data into the data warehouse

Given that you only have a snapshot, it is clear that for the slowly changing dimensions, for every object there will only be one version, being the current one.

In the data warehouse, you can set the start date for these objects as following:

- 01/01/2014 for customer
- 01/01/2007 for film
- 01/01/2007 for inventary
- 01/01/2014 for payment
- 01/01/2014 for rental

- 01/01/2007 for staff
- 01/01/2006 for store

Given that we do not have access to all past transactions that led to the current balances in the accounts, in the initial load you will have to add for every account one “artificial” transaction of type “I” (of Initialization), with the amount set to the current balance of the account. In this way, for any account it will always hold that the sum of all transactions in the data warehouse for that account equals its balance.

4 Database and Datawarehouse Description

In this section we repeat the database description of Part I of the assignment, and we detail the data warehouse structure.

Every employee personal information is stored (first name, last name, gender, date of birth, marital status, number of children), as well as the date he or she was hired, his or her current salary, title, and department. There is a current work location for every employee and also his or her base location (city he or she was hired from). The employees can work for one or more clients. For client, the client name and the category (large,medium,small) is stored. For every city the state and for every state the country is stored. Figure 1 shows the database tables used by human resources. This is the database snapshot of January 1st 1991 (`"1991-01-01 00:00:00.000"`).

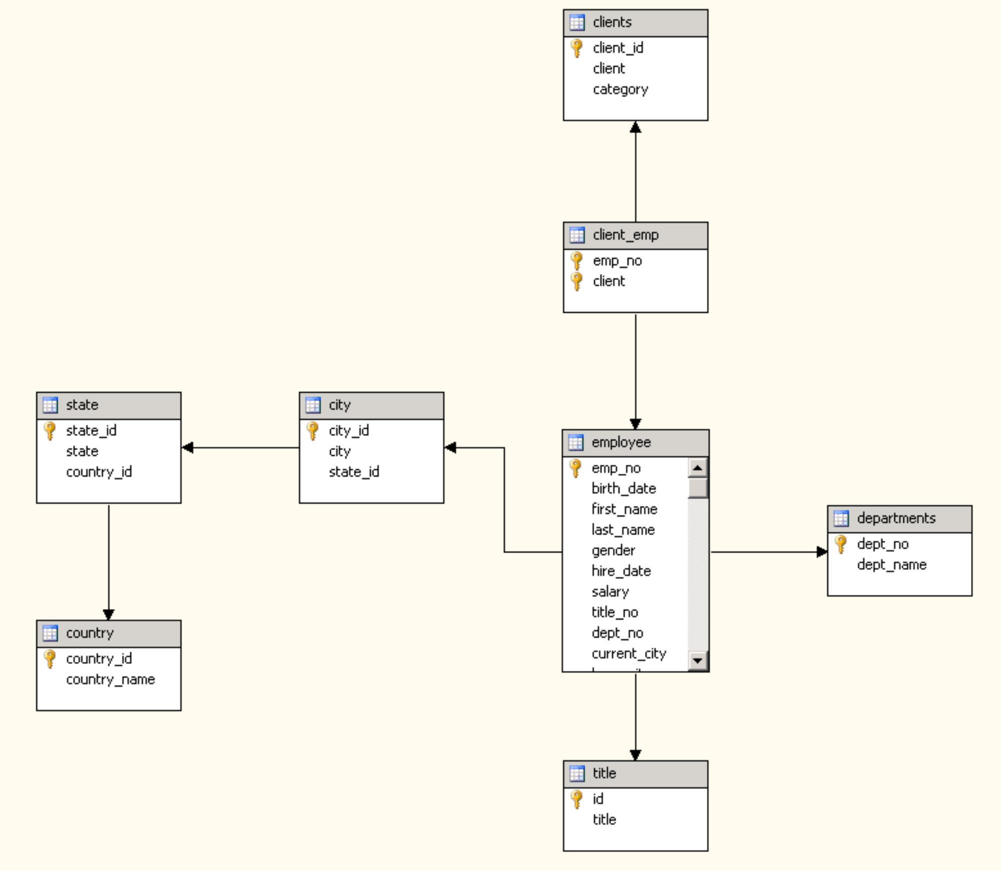

Figure 1: Database tables of the Assignment1 database.

5 Data Warehouse Design

We continue working with the database of the first assignment. Based on feedback of the prospective users of the data warehouse, the description was further fine-tuned by adding the following constraints:

- For employees the attributes first name, last name, gender, date of birth, base location, and the date he or she was hired should in principle not change; if they change it indicates a correction in the database.
- Marital status, number of children, current salary, title, department, and current work location may change over time for employees. These changes should be recorded.
- The clients for which an employee works may change.
- On exceptional occasions client names and categories may change. Since this event is so rare, it was decided to treat a change of name or category in the data warehouse as if it concerns a new client. Furthermore it is not necessary to be able to produce a historical overview of who worked for a company at what time. Given a salary fact of an employee, however, you should be able to produce the list of companies for which the employee worked at the time of the fact.

After considering all possible designs the data warehousing team selects the design in Figure 2 for the data warehouse to be constructed[1].

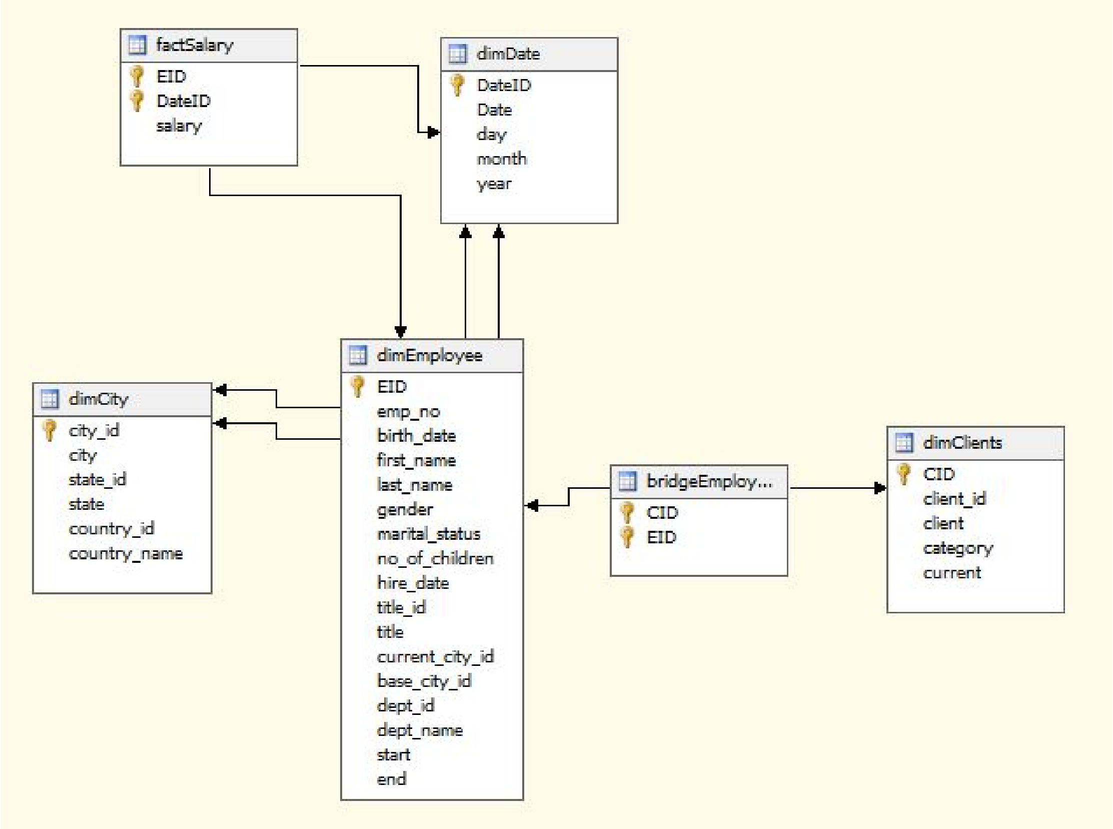

Figure 2: Schema of the Data Warehouse.

6 Snapshots

In order to test the benefits of the data warehouse, the company is asking for a proof of concept, based on 15 historical snapshots of the database that can be taken from backup (see next section). Your task is now **to define a SSIS package to incrementally load** these snapshots into the data warehouse.

The snapshots have an increasing level of difficulty. There are four levels of difficulty.

[1] Notice that some of the requirements of the original assignment have been slightly updated. The choice for this Particular design hence does not necessarily imply that the design submitted by your group is incorrect if it diverges from the chosen design.

Level 1: snapshots 1-10 You can assume and use in the development of your script that the many-to-many relation between existing employees and companies is stable; that is: although employees may be added and new employees may be connected to new companies, the association of existing employees with companies does not change. Furthermore, you may assume and exploit that the attributes of a client are static. Of course new clients can be added as long as they are associated with new employees.

Level 2: snapshots 11-12 Similar as for level 1, except that now the attributes of client may change. Recall that in such a case you may treat the changed company as if it is a new company.

Level 3: snapshots 13-14 In these snapshots also the mapping between employees and clients may change.

Level 4: snapshot 15 Do not assume anything. There may be data quality problems.

Obviously grading will be related to the level of complexity reached. The following table gives an *indication* of how grading is affected by the complexity of the different variations of the assignment in case of a good solution for the specific variation. Obviously, grading will be highly influenced by the quality of your solution. Submit a solution *for one variant* and clearly indicate which variant you have chosen.

	level 1	level 2	level 3	level 4
reduced size	not a passing grade	border 7/10	good 8.5/10	excellent 9.5/10
full size	border 6/10	good 7.5/10	very good 9/10	excellent 10/10

7 Obtaining the Data

The snapshots are available from the course website and from the link `http://k6.re/tUVl_`. The snapshots have an extra file a **snapshot.csv** which has the from-date and to-date of the snapshot. For a given snapshot, the start-date denotes the date of the last snapshot before the given one and the end-date denotes the date of the given snapshot.

In order to facilitate the loading of the snapshots a package **"assignment2.dtsx"** has been constructed. The following instructions explain how to import and use the package:

- Open assignment2.dtsx in SSIS (File > Open > Open File)
- Open Variables window (SSIS > Variables)
- Edit the value of variable "path" to the folder where you extracted the zip file.
- **Make sure you do not edit any of the *existing* connection managers or the for loop iterator.**

If you open the package you will see the control flow in Figure 3. In the control flow you can see a for-loop container. This for-loop container loops over all snapshots. Your ETL script for loading one snapshot will go inside the for-loop container, more specifically inside the sequence container. Before the sequence container there are some tasks that load the data into temporary tables in the data warehouse. These tables can be recognized by the prefix "new." Outside the loop-container you should not put anything except maybe create table statements should you decide to maintain some tables between incremental loads to facilitate the loading process; for instance OLTP_key to surrogate_key mapping tables[2]. Notice that there are two tasks disabled. If you enable them, the problem will be simplified by reducing the data size.

[2] This remark does not imply you *have* to maintain such tables, only that you *can*.

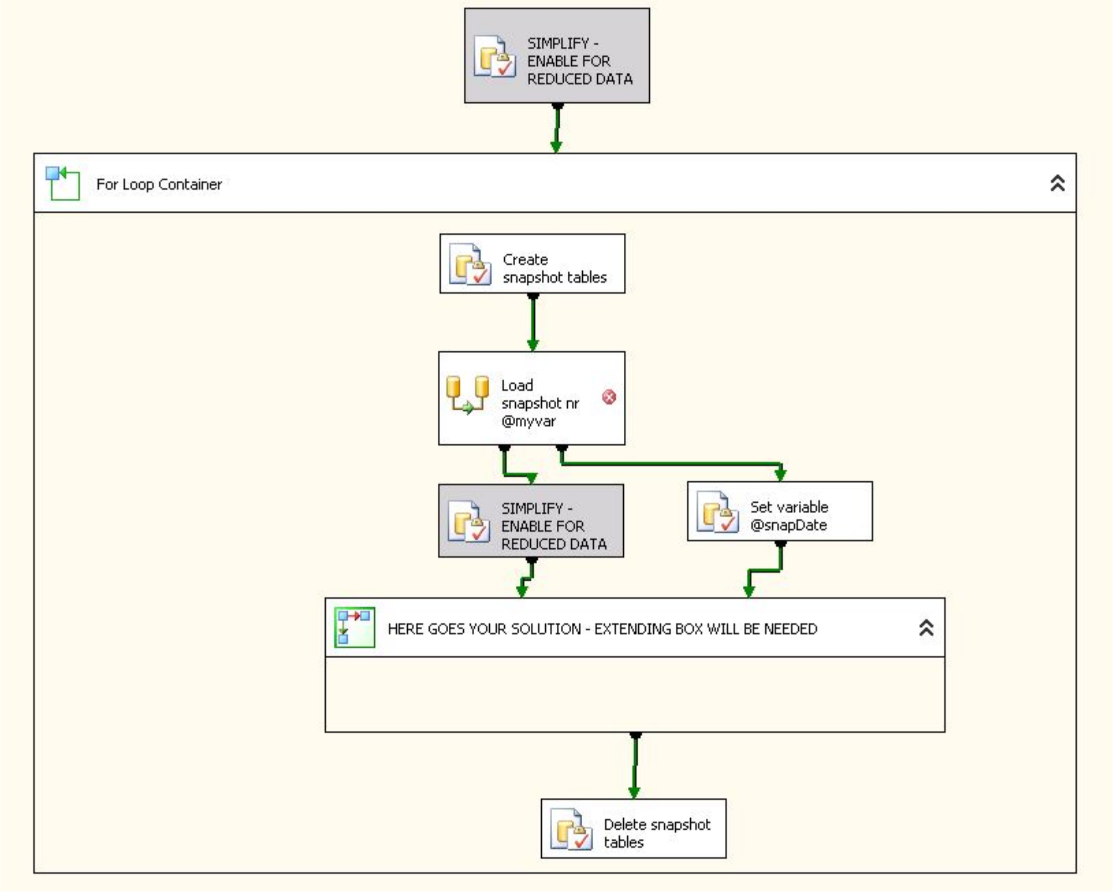
SIMPLIFY - ENABLE FOR REDUCED DATA
For Loop Container
Create snapshot tables
Load snapshot nr @myvar
SIMPLIFY - ENABLE FOR REDUCED DATA
Set variable @snapDate
HERE GOES YOUR SOLUTION - EXTENDING BOX WILL BE NEEDED
Delete snapshot tables

Figure 3: Control flow of the helper package. The for-loop container loops over all snapshots. Your ETL script should normally go inside the sequence container. The two disabled tasks simplify the data a bit for those who want a reduced-complexity assignment. Enable them for an easier exercise.

8 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.
- The variable @snapDate contains the date of the currently loaded snapshot. You can use it to set start- and end-dates of versioned tuples.
- 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).

9 Deliverables

You should deliver a report containing:

1. A **cover page** with the list of group members, including student ID,
2. Your estimation of the growth of the data warehouse size over time (1 point),
3. Figures showing all your data flows and control flow with a succinct explanation whenever needed (1 point),

4. The SSIS package that does the initial load of the data warehouse based on the database that was used in the first Part of your assignment and on the design in Figure 2 (8 points).
5. The ETL package for the incremental load of the data into the data warehouse. The package should load the snapshots one by one and the package should end when all snapshots have been loaded (8 points),
6. The SSAS project for creating the cube. Deploy a data cube on top of the data warehouse and create a report. In your cube you should offer a view to the end-user that is consistent with the conceptual schem in Figure 1 (2 points),.

Submit all files in a single .zip-file on the université virtuelle course website. After submitting your solution, please to fix a date between December 11 and December 22 and a time to present your solution. All group members must be present during the presentation.