Data Warehousing Assignment—Part II

Toon Calders Rohit Kumar

Deadline: 23 November 2015

1 Practical information

Deadline:	23 November 2015
Group:	Same groups as for assignment part I
How to submit:	Upload solution at `uv.ulb.ac.be`

2 Objectives

The goal of this assignment is:

1. Create an ETL script for the initial load of the data warehouse.

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 the first loading script. 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 dw2015 database as input and produce a data warehouse reflecting the current state of the operational database. Given that you only have the snapshot of December 31st, 2007, it is clear that for the slowly changing dimensions, for every object (e.g. customer) there will only be one version, being the current one. You can set the start date for these objects to December 31st, 2007 in the data warehouse. Later on, for parts III and IV of the assignment, further snapshots (one every two days) will be provided starting from January 1st, 2008. 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 based on part I of the assignment.

4.1 Database Structure (taken from part I description)

The iNG bank is maintaining a simple MSSQL database of its customers and their daily transactions. For every customer personal information is stored (First name, last name, gender, date of birth, marital status, number of children), as well as the accounts he or she has. For every account the date it was opened, the current balance, account type, and branch where it was opened is stored. There is a current city for every customer (where the customer is currently residing), and for every branch the city where it is located. The customer can have more than one account in more than one branch. Later on there will be some accounts which are external accounts and do not belong to any customer of the bank. For those the owner id of the account will be *null*. Every transaction (deposit or withdraw) done by every customer is stored in a transaction table. This table contains the account for which the transaction was performed, the datetime and the amount (positive for deposit and negative for withdrawal) of the transaction. Further, the transactions have a ‘from account’ which maintains the account id from where the money came (if it was a deposit) or went (if it was a withdraw). For every city the state and for every state the country is stored. Figure 1 shows the database tables used by the bank. A copy of this database is available on the SQL Server CS-MSSQL under the name dw2015. This is the database snapshot of January 1st 2011. Notice that the transaction table is empty because the bank will start recording transactions in the data warehouse only from January 1st 2011 on. Later on further snapshots will be released. In these future snapshots the transaction table will gradually be filled.

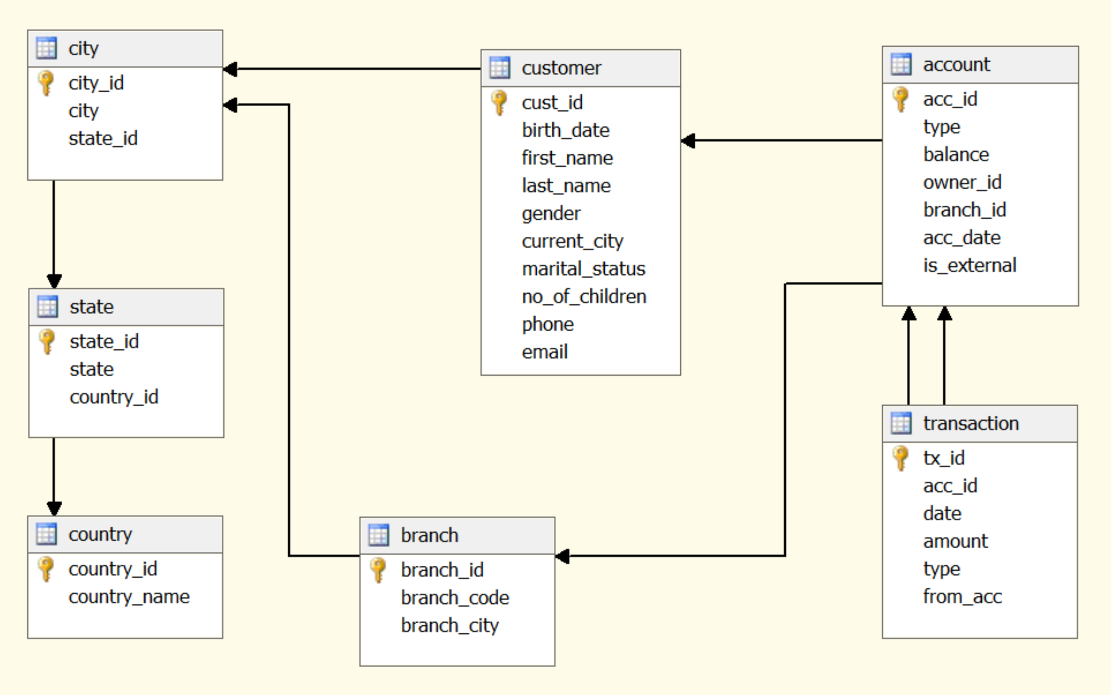

Figure 1: Database schema of the database for assignment part I. This database is available on the CS-MSSQL server as dw2015.

4.2 Datawarehouse Structure

Based on the database structure and the requirements stated in part I of the assignment, a relational data warehouse schema was constructed. The create table statements for this data warehouse structure are available on the course website. The conceptual Dimensional Fact Model (DFM) on which the model is based, is given in Figure 2.

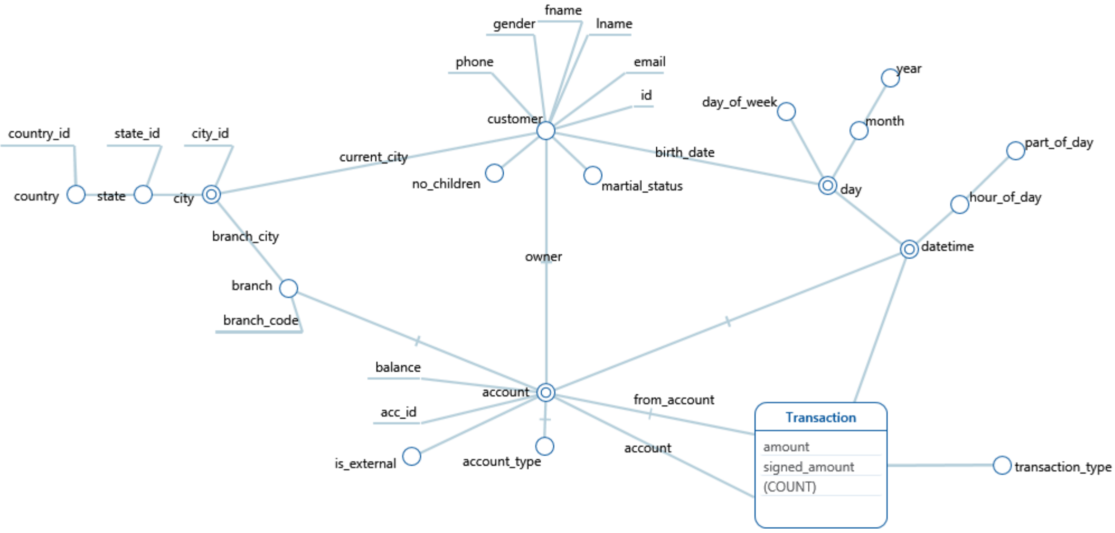

Figure 2: Dimensional fact model for the data warehouse of part I of the assignment.

For every transaction fact in the factTransaction table we have two accounts: the account for which the transaction is recorded identified by its surrogate key AID, and the account AID of a potential other involved account other_account_AID. Both are foreign keys into the dimAccount dimension table. Furthermore, there is a transaction_type which corresponds to the type attribute in the operational transaction table. This type can be for instance “C” (credit), “D” (debit), “W” (withdrawal), and “I” (initialization). Type is a degenerate dimension and hence no separate table is created for it, but the value is immediately included in the transaction table. The datetime of the transaction is split over two attributes: the date itself, and the time in seconds. These values can be obtained in SSIS by using a derived column and using the following expressions:

```
(DT_I4)(DATEPART("Hour",acc_date) * 3600 + DATEPART("mi",acc_date) * 60
                                            + DATEPART("s",acc_date))
(DT_DBDATE)acc_date
```

The first expression gives the number of seconds for datetime acc_date since the start of the day. This value at the same time serves as the key for the dimTime dimension. The second expression removes the time-part from the datetime attribute acc_date. For instance, the date 1971-08-11 20:02:24.000 is divided into two parts: the number of seconds 20*3600+2*60+24=72144 and the date 1971-08-11. Furthermore in the factTransaction we keep the new balance of the account after the transaction, and the absolute value of the amount (unsigned_amount) of the transaction, and the signed amount (signed_amount), which is -unsigned_amount in case money leaves the account identified by AID, and unsigned_amount if money is received in the account. We decided to keep the balance in the fact table instead of in the account table because it changes with every transaction. In some sense one could argue that balance is a rapidly changing data value and we opted for capturing this change using the type-4 strategy for dealing with rapidly changing dimensions. dimTime and dimDate contain the date and time dimension with their corresponding hierarchies. The meaning of the attributes is self-explanatory. As we need to store history for the account dimension dimAccount, two datetime attributes start and end have been added that

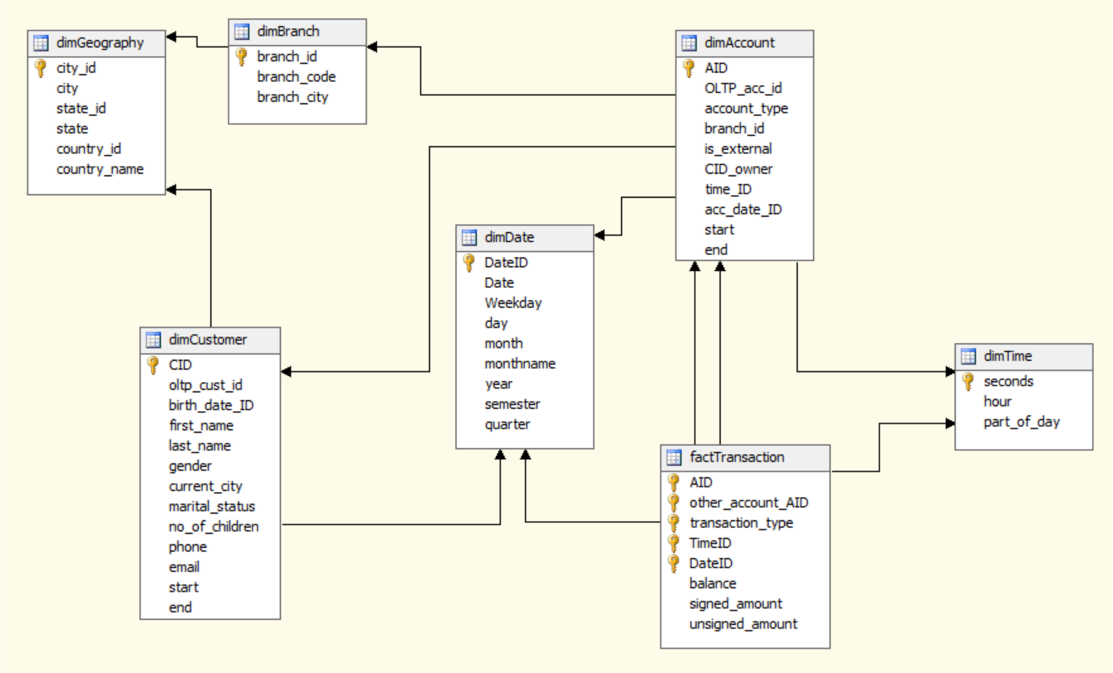

Figure 3: Relational scheme for the data warehouse of assignment part I.

will maintain the history of accounts. Recall that branch_id and account_type are attributes for which the history needs to be kept. Changes in the other attributes represent corrections. dimCustomer is the other versioned dimension. Here the history of the attributes current_city, marital_status, no_of_children, phone, and email need to be kept. For the other attributes, again, changes represent corrections that have to be propagated to the data warehouse, but for which the old values can be overwritten. Similarly all changes in any attribute of a non-versioned dimension should be considered as corrections. Exceptions are the dimDate and dimTime dimensions which are fixed.

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)
   (c) A description of all connection managers that your script is using, and how to set them to the correct values. For instance, if you are using a flat file resource, explain which flat file connector refers to it.

2. The ETL package for SSIS that performs the initial load of the data into the data warehouse. Your connection manager should contain 2 “OLE DB” data sources, one named “**source**”, referring to the OLTP source database (that is: the dw2015 database on CS-MSSQL, or your local copy), and one named “**target**” that refers to your target database in which the data warehouse will be stored. If you have further connection managers, for instance flat file connection managers, then give them descriptive names and make sure to clearly describe in your document to which files they refer.

   You can assume that the target data warehouse exists and contains all tables of the model solution. These tables will be empty in the target database; that is, when testing your script, we will follow this procedure: we will create a new database and run the create table statements given in the model solution of part I. Then we will manually update the connection managers such that they are set to the correct values for the testing environment. This may involve copying flat files provided by you to the local disk of the testing environment, and updating the path of the flat file connectors. **Please make sure your script can be run without requiring an extraordinary complicated configuration procedure.**

3. Possibly additional files (.csv, .txt, ...) that you are using in your ETL package.

Submit all files in a single .zip-file on the université virtuelle course website.