Data Warehousing Assignment—Part II

Toon Calders   Hatem Haddad

Deadline: 07 December 2016

1 Practical information

Deadline:	07 December 2016
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.

2. Deploy a data cube based on 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 dvdrent database as input and produce a data warehouse reflecting the current state of the operational database. 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. Later on, for parts III of the assignment, further snapshots will be provided.

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)

A DVD rental company maintains a database with the current information about its rentals. For every customer, personal information is stored (first name, last name). The address table contains address information for customers, staff and stores. The film table is a list of all films potentially in stock. The inventory table is about the actual in-stock copies of each film. A film can have one or more categories. The payment table records each payment made by a customer with information about the payment date and the amount. The rental table is about rentals with information about who rented what item, when it was rented, and when it was returned. The attribute last_update in each table represents the date and time that the row was created or most recently updated. Stores have multiple inventories. A rental has zero or one payment. Figure 1 shows the database tables used by the DVD rental company.
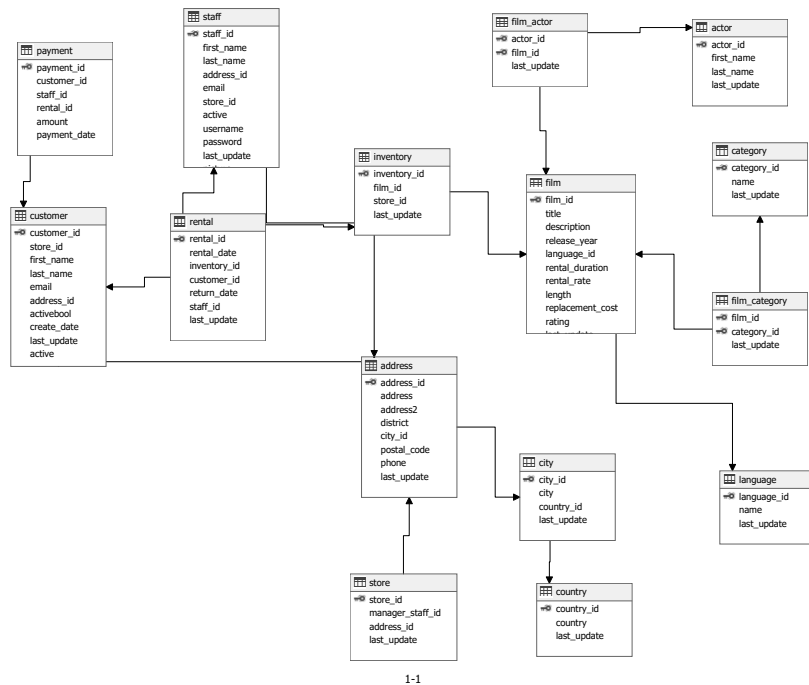

Figure 1: Database tables of the Assignment1 database.

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 fact table is RenatlPayment. The dimensions are: rental, Date, staff, customer, store and film. The attribute payment_status defines if a rental is payed or not. CID, staffID and storeID are surrogate keys of customer, staff and store dimensions. The data cube should be based upon the relational scheme for the data warehouse (Figure 2).
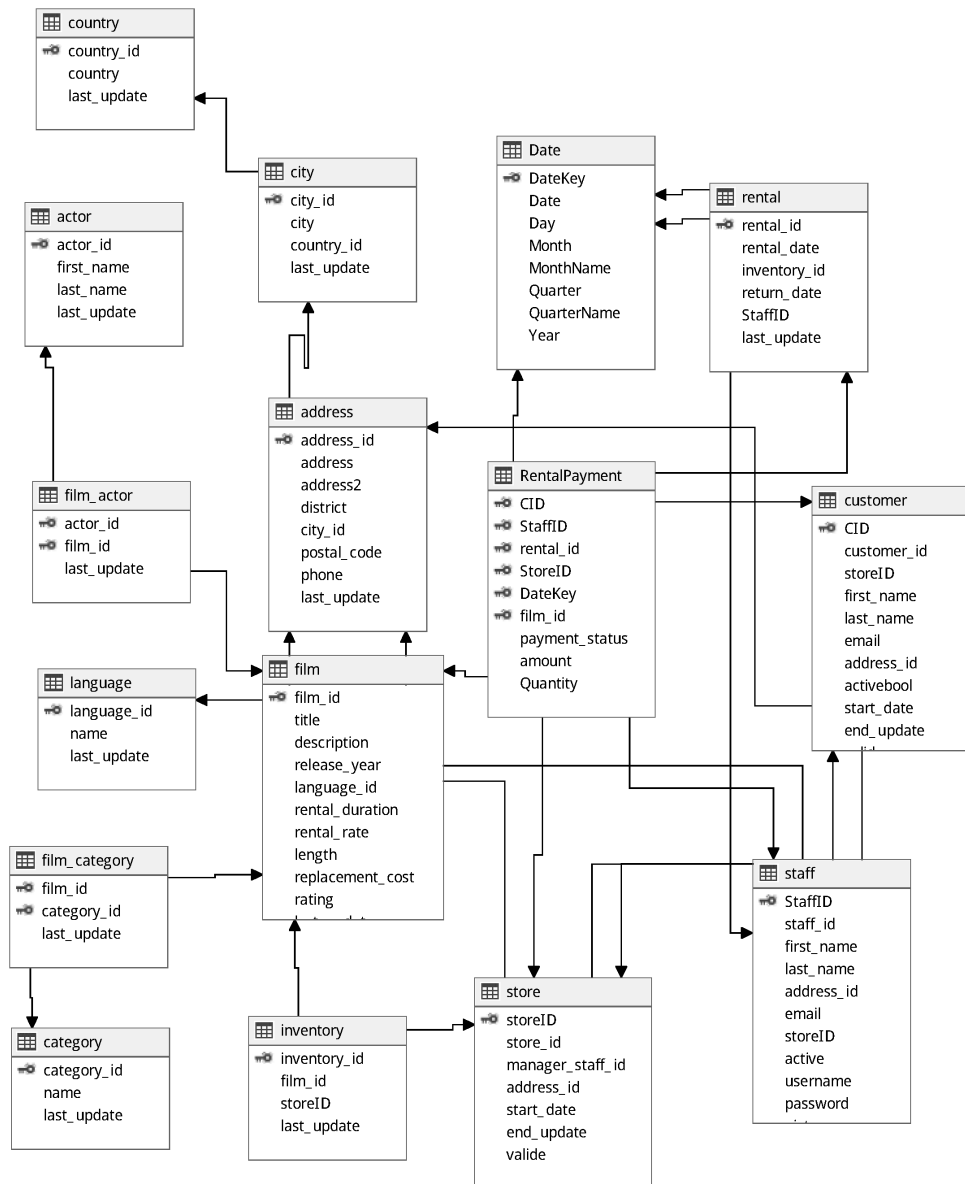
5 Deliverables

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 dvdrent database, 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.

4. Figures showing your data cube in the cube browser (Make sure dimensions and hierarchies are visible) and additional explanations you think are necessary.

5. The SSAS project for creating the cube.

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

country
- 🔑 country_id
- country
- last_update

city
- 🔑 city_id
- city
- country_id
- last_update

actor
- 🔑 actor_id
- first_name
- last_name
- last_update

Date
- 🔑 DateKey
- Date
- Day
- Month
- MonthName
- Quarter
- QuarterName
- Year

rental
- 🔑 rental_id
- rental_date
- inventory_id
- return_date
- StaffID
- last_update

address
- 🔑 address_id
- address
- address2
- district
- city_id
- postal_code
- phone
- last_update

RentalPayment
- 🔑 CID
- 🔑 StaffID
- 🔑 rental_id
- 🔑 StoreID
- 🔑 DateKey
- 🔑 film_id
- payment_status
- amount
- Quantity

customer
- 🔑 CID
- customer_id
- storeID
- first_name
- last_name
- email
- address_id
- activebool
- start_date
- end_update

film_actor
- 🔑 actor_id
- 🔑 film_id
- last_update

language
- 🔑 language_id
- name
- last_update

film
- 🔑 film_id
- title
- description
- release_year
- language_id
- rental_duration
- rental_rate
- length
- replacement_cost
- rating

film_category
- 🔑 film_id
- 🔑 category_id
- last_update

staff
- 🔑 StaffID
- staff_id
- first_name
- last_name
- address_id
- email
- storeID
- active
- username
- password

category
- 🔑 category_id
- name
- last_update

inventory
- 🔑 inventory_id
- film_id
- storeID
- last_update

store
- 🔑 storeID
- store_id
- manager_staff_id
- address_id
- start_date
- end_update
- valide

1-1

Figure 2: Relational scheme for the data warehouse of assignment part I.

4