Escuela Politécnica Superior, UAM

Estructura de Datos 2017-2018

Lab 1. Designing a relational data base.

Due: Oct. 17

Labs are to be turned in using the moodle platform. In the "prácticas" section of the class page there will be various links (one per group) for this purpose. Please upload the file using the link relative to your group. Delays will incur a penalty of 1 point per day.

USEFUL ADVICE (surprisingly, it is actually useful...)

1. Reports are important: they are not just so that your lecturer may have some idea of what you have done, it also gives important indications on the kind of reasoning you have followed and why you did things the way you did. This will help us better evaluate your work. Besides, writing a good report is a crucial skill for those who, like you, will work in a technical field: let your boss, your colleagues, and your customers understand your work will be one of the most important aspects of your profession. You will have to write good reports, eventually. You might as well start now.
2. This might seem trivial, but... put the name of ALL members of your group in ALL files! You have no idea in how many different ways is it possible to lose track of who did what.
3. Turn in all documents in pdf format. Turning in files in a editable format is a very bad practica, even if you are using such a common format as MS Word. Cruel and unusual punishment will be applied to those who upload documents in format other than pdf. The only exception is plain ASCII text files: those can be uploaded in text format.

Objetives

In this lab we shall develop a simple design based on a ER diagram and we shall translate it into a relational design. We test the model by expressing some simple queries in it.

NOTE: a lot of people are a bit nonplussed since in this lab we ask to create queries but we give no data. This is done on purpose: *Queries are created based on the schema of*

the data base, and NOT based on the data it contains. With the schema alone, it is possible to write perfectly valid SQL queries. You won't be able to try then out, of course, but the very fact of being able to write them indicates that the schema contains all the information necessary to solve a problem.

If you really really really (really) need to try the queries out, your best option is to create (by hand or with a text file) a data base with very few data: it is foolish to try to understand if a query works by trying it out on a data base with 100.000 tuples. It is better to create a data base with three or four tuples, execute the queries by hand and then, when you know what the answer must be, execute the query. Debugging is an art that you will eventually have to master.

The problem that we shall consider in this lab is the creation of a relational model (and data base) of a bookstore, one that will manage the book catalog, the authors, sales, discounts, etc. Here are some general considerations:

1. The bookstore sells books (duh!), but a book is a rather complex and structured thing. One has to distinguish between a "title" (a book as an abstract object: *Of mice and men*, for example), which has one or more authors (John Steinbeck, in this case) and its various editions: there are editions in different languages (*De ratones y hombres*, *Uomini e topi*, *Des souris et des hommes*, etc.), hard cover and paperback editions, special editions, etc. Each edition of a book will in general have a different publisher and a different price.
2. The bookstore has special sales. A special sale is a discount that is applied to a set of books if they are bought between two specified dates (the discount is fixed: each special sale applies the same discount to all the editions it applies to). The bookstore must have a record of all the special sales it does.
3. The bookstore has a certain number of registered used, each one with a unique id and a (and only one) credit card (note that different users may share the same credit card). Each registered used receives a 10% discount on each book he or she buys, including books on sale. The bookstore needs to keep a record of users and of how much money each of them spent.
4. Other people, people who are not registered can, of course, buy books: the bookstore needs a record of all the sales, indicating which editions have been bought, how much it was spent, whether the purchased was paid in cash or credit card, the date of the purchase, etc. It must also be possible to determine whether some offer has been applied. This same information must be recorded for register users as well.

Thou shalt begin by creating an ER model with all the entities and relations necessary to model the bookstore. Once the model is complete, and once you have verified that all

necessary information is present, you will transform the ER model into a relational model using the standard methods.

In order to valídate the model check that it provides enough information and relations to answer the following questions:

1. Given a title, hoa many editions does it have? In how many languages?
2. How many books of author X were sold?
3. How many books of author X were sold at a discount?
4. How much money was earned by selling books of author X?
5. How many books were sold to registered users?
6. How many registered users have bought English books?
7. How much money was earned by selling books in French?
8. In which days were books of the publisher Adelphi on sale?
9. Which registered users have never bought paperback books?

Write the corresponding SQL queries for the relational model you have created.

What must you upload:

a. A report in which you analyze your design. Specifically, the report must analyze posible redundancies in the design and points which have the potential to introduce inconsistencies in the data. (Redundancies are not the devil and are not necessarily evil: sometimes they might help improve the efficiency of the data base and may simplify queries; but it is important to know that they are there, where they are, and what problems they might cause.)

b. In the report, you should do a brief analysis of the proposed queries; for each one, write down the relations and entities that are involved in its execution.

c. A ER diagram of your design.

d. The list of the headers of the relations of teh relational model derived from the ER diagram (with the suitable optimizations).

e. The text of the SQL queries corresponding to questions 1-8 (query 9, more complex, is optional).

Important: If you include diagrams, drawings, schemas or other graphical manifestations of your fancy that are not included in the report (e.g. the ER diagram), they must be included as a separate pdf file, one file per drawing. It is acceptable to draw the diagrams on paper and scan them (do not take pictures with your smartphone: the quality is usually terrible due to lighting), as long as the diagram is reasonably well done and readable: a maze of tortured lines on a dirty pizza napkin is not quite acceptable!

Even more important: All your files must be tar-ed or zip-ed in a single file in .tgz or .zip format with name EDAT1718_p1_XXX_AAA_BBB.tgz (or .zip) where XXX is the number of your lab group (if you are reading this in English it will probably be 1251 or 1291), and AAA and BBB are the last names of the members of the group (don't be overly Spanish: one last name is enough). The file must be uploaded using moodle and the ink that will be published in due time. Only one of the members of the group must upload the file. Please avoid double uploads: don't you worry; we are going to grade both of you! Especially if you follow an advice which is

Tremendously important: the name of the tgz file allows us to know who you are, but it is a good idea to **put both your names in all the files that you turn in**. Let me tell you again: write your name in all the files, **write your name in all the files, WRITE YOUR NAME IN ALL THE FILES**. I hope I made myself clear… (This is not just for EDAT: do it in all your assignments; you will avoid a lot of potential trouble).