DM 505 Database Design and Programming

Spring 2009 Project

Department of Mathematics and Computer Science
University of Southern Denmark

February 21, 2009

Introduction

The purpose of this project is to try in practice the process of designing and creating a relational database application. This process includes development of an E/R model, transfer to the relational model, normalization of relations, implementation in a DBMS, and programming of an application providing user interaction with the database.

The project is the exam for HA(Dat) students and part of the exam for IMADA students. Thus, the project must be done individually, and no cooperation is allowed beyond what is explicitly stated in this document.

PostgreSQL should be used as the DBMS. IMADA students should use Java while HA(Dat) students should use PHP. For more details on the deliverables, see the sections IMADA Students and HA(Dat) Students, respectively.

System Requirements

The subject of the project is to create an electronic inventory for a computer store. The idea is to keep information about complete computer systems and components in a system which can calculate prices for components and computer systems, make lists of components to order from the distributor, and ensure a minimum inventory.

At least the following objects should be modeled in the system.

component: name, kind, price
: Here, kind is one of CPU, RAM, graphics card, main board, case. The individual kinds have the following additional attributes:

  CPU: socket, bus speed

  RAM: type, bus speed

  mainboard: CPU socket, RAM type, on-board graphics?, form factor

  case: form factor

computer system: catchy name, list of components
: A computer system requires a case, a mainboard, a CPU, RAM, and, optionally a graphics card. It is required that sockets, bus speed, RAM type, and form factor match. Furthermore, if there is no on-board graphics, a graphics card must be included.

current stock: list of components and their current amount

minimum inventory: list of components, their allowed minimum amount and their preferred amount after restocking

Note that the above is a specification from the user – the objects may or may not be entities in the E/R diagram you develop.

The computer store restocks his inventory every Saturday at his main distributor. The intended use of the system is to print a daily price list for components and computer systems, give quotes for custom orders, and print out a list of components for restocking on Saturday morning.

The selling price for a component is the price plus 30%. The selling price for a computer system is computed as the sum of the components' selling prices. It is then rounded up to the next number ending in "99". To encourage customers to buy more than one computer system at a time, the total price of a custom order is reduced by 2% for each additional computer system, with a limit of 20% total reduction.

For example, a computer system for which the components are worth DKK 1984 and, thus, the selling price of the components is 1984*1.3 = 2579.2, would be sold for DKK 2599 . A customer ordering 3 systems would pay DKK 7485, i.e., DKK 2495 per system. The price per system for 11, 23, or 42 computer systems would be DKK 2079.

At least the following functionality should be provided by the application:

List of all components in the system and their current amount

List of all computer systems in the system and how many of each could be build from the current stock

Price list including all components and their selling prices grouped by kind as well as all computers systems that could be build from the current stock including their components and selling price

Price offer given the computer system and the quantity

Sell a component or a computer system by updating the current stock

Restocking list including names and amounts of all components needed for restocking to the preferred level

In a real system, facilities for updating inventories after restocking at the distributor as well as updating computer systems are necessary. However, to limit the amount of programming, these do not need to be included in the application. In the project, you can update computer systems and components simply by using `psql` or the web frontend. You are of course welcome to add these facilities, but this will not influence the final grade.

Tasks

Your tasks are:

1. To develop an appropriate E/R model of the system.
2. To transfer the E/R model to a relational model.
3. To ensure that all relations are in 3NF. If not, you have to decompose the relations and refine the E/R model accordingly.
4. To create these relations in a database in the PostgreSQL DBMS. The database should ensure the constraints described above.
5. To program in Java or PHP, respectively, an application providing the user interaction with the system. The application should provide the functionality described above.

Input and Output

To limit the amount of programming, only a simple command-line based interface for user interaction is required. For instance, choices by the user can be input by showing a numbered list of alternatives or by prompting for component names, computer systems etc. You are welcome to make a more advanced user interface, but this will not influence the final grade.

Test Data

The test data can be made up as you need it. A decent amount of test data must be made (at least in the order of 8 computer systems and 30 components). Sharing data with other participants in the course is explicitly allowed and encouraged.

Formalities

In the end, a printed report of 10-15 pages should be produced. Its main aim should be to

- describe the design choices made during the development as well as the reasoning behind these choices, and
- the structure of the final solution.

Specific items that must also be included in the report are: A diagram of your E/R model, the schemas of your relations (probably in an appendix), arguments showing that these are in 3NF, the central parts (with explanation) of your SQL code, and a (very) short user manual for the application. The emphasis of the project is not on testing, so no documentation of testing is required in the report. However, your program will be tested for the grading of the project.

The report should be produced in two stages. In the first stage, the preliminary report describing the design choices, the E/R model and the resulting relational model are handed in. The preliminary report will be commented on and handed back. In the second stage the preliminary report is corrected and extended to the final report. This is then handed in together with the original preliminary report including the comments. The grade for the project will be based both on the preliminary and on the final report.

The stages and their deadlines are:

1. Task 1-3, deadline March 6, 2009
2. Task 4-5: deadline March 20, 2009

IMADA Students

IMADA students should use Java with JDBC for programming. The SQL and Java code should not be given as a printed appendix, but rather be handed in using the `aflever` command on the IMADA system: Move to the directory containing your code and issue the command "`aflever DM505`". This will copy the contents of the directory to a place accessible by the lecturer. Repeated use of the command is possible. Later uses overwrite the contents from earlier ones. You should have a copy of your final database on the PostgreSQL server 10.110.4.210, and your code should work on this copy, i.e., the test during grading will be made with the program you delivered using this database.

HA(Dat) Students

HA(Dat) students should use PHP and deliver a fully functional website. The SQL and PHP code should not be given as a printed appendix, but rather be uploaded to the server 10.110.4.210 using WebDAV. During grading the website on this server will be accessed for testing. For details consult the lecturer in his office.