DM26 Database Systems

Fall 2004 Project

Department of Mathematics and Computer Science
University of Southern Denmark

October 20, 2004

Introduction

The purpose of this project is to try in practice the process of creating a relational database application. This process includes design in the ER-model, transfer to the relational model, normalization of relations, implementation in a DBMS, and programming of an application controlling user interaction with the database.

The project should be done in groups of two persons (single person groups and three persons groups only by special arrangement with the lecturer). PostgreSQL should be used as DBMS, and Java with JDBC should be used for programming.

System Requirements

The subject of the project is an electronic cookbook. The idea is to keep information about recipes and kitchen inventory in a system which can suggest recipes, make list of things to buy, and ensure a minimum inventory at all times.

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

Recipe Required features: name, list of ingredients, number of persons served, the actions for making the dish, total preparation time, actual working time, type (e.g. hors d'oeuvre, game dish, dessert), picture of dish (if available), comments on recipe.

Ingredient Required features: name, type (vegetable, fruit, etc.), price.

Menu Required features: list of recipes, comments.

Minimum Inventory Required features: list of ingredients and their allowed minimum amount.

Current Inventory Required features: list of ingredients and their current amount.

Several minimum inventories can be envisioned to exist, for use in different situations (e.g. everyday, Christmas time, guests in house), whereas there is only one current inventory (most houses have only one kitchen).

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

Print recipe scaled to a given number of persons.

Show recipes containing a given set of ingredients.

Show recipes possible to make with the current inventory.

Show recipes possible to make within a given time limit.

Show recipes possible to make within a given price limit (when serving a given number of persons).

Show recipies that are vegetarian.

Show total preparation time and price for menu.

Write out list of things to buy to be able to make a given recipe or menu, given the current inventory and a number of persons to serve.

Write out list of things to buy to ensure a given minimum inventory.

Besides queries, updates should of course be allowed. It should be possible to update inventories based on the use of a given recipe, or based on shopping according to a list produced by the system. Additionally, manual updates of amounts of single ingredients should be possible. To limit the amount of programming, no further facilities for updates needs to be included in the application (i.e. in the project, you add recipes, ingredients, etc. to your database simply by using `psql`).

The system described above is the minimal required solution. Groups are free to extend the project to include more features (further objects and functionality).

Tasks

Your tasks are:

- To develop an appropriate ER-model the system.
- To transfer this to the relational model.
- To ensure that all relations are in BCNF form (if necessary, decomposing relations, which again should lead to a refinement of the ER-model).
- To create these relations in a database in the PostgreSQL DBMS.
- To program (using Java and JDBC) an application controlling 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 is required for user interaction. For instance, choices by the user can be input by showing a numbered list of alternatives, after which the user inputs the desired number. Project groups are of course free to make a more elaborate (graphical) user interface (which will be more fun, but no extra credit is given for it).

Formalities

A printed report of 10 to 20 pages should be handed in. Its main aim should be to describe the design choices made during development, the reasoning behind these choices, and the structure of the final solution.[1] Specific items that must also be included in the report are: A diagram of your ER-model, the schemas of your relations, proofs/arguments showing that these are in BCNF form, and a short user manual for the application.

Actual Java code should not be handed in (besides small excerpts in the report text), but should be made publicly accessible on you departmental account (with the location specified in the report). You should have a copy of your final database on the PostgreSQL server of the department (at the machine `dbhost`), and your code should work on this copy (i.e. it should be possible to try out your system).

You must hand in the report by

Wednesday, December 1

[1] To prepare for the writing of the report, it is a good idea to keep a log of the discussions within the group and of the work done.