Introduction to Database Design, Fall 2014

Carsten Schuermann

17. juli 2014

1 A simple database – with and without DBMS

Suppose that you must implement a system that contains information on the food in a canteen. The following information should be stored:

- Information on the dishes that can be bought. Each dish has a name and a price.
- Each day it is registered which dishes are for sale, and how many of each are sold.
- There is a list of ingredients, each having a name and a supplier. For each dish it is recorded what ingredients go in, and in what quantity.

a) How would you store this information *without using a DBMS*? Sketch an implementation in an object-oriented, imperative language. Think about that the implementation should be flexible and e.g. allow that you change the supplier of milk. You are not supposed to write any code, just give an overall design.

You can imagine many queries on such a database, e.g.:

- Find all dishes containing eggs.
- Find the total sales amount (in DKK) of today.
- Find the most sold dish of today.

b) How would you implement these queries? Sketch a solution. Especially for students who have taken the algorithms class: Think about how the solution scales to large data sets. Can you avoid repeated linear traversals of data?

c) Create a relational data model for the database. Write down the database schema.

Later in the course we will see how the above queries can be done with little effort, and high performance, using a DBMS.

2 Relational modeling

Read the case description in problem 1 of the exam of January 16, 2006. (Available in the collection of exam problems linked to above the course schedule.) Make a relational data model for the described database. If you make assumptions on data that are not given by the description, state them in words.

3 Universal relation (advanced)

Design a relation schema that can contain the information of any relation in any schema. This construction is related to the concept of a "universal relation". As an example, convert the following relation instance into a relation instance in your "universal" schema:

accountNo	*name*	*address*
12345	Scrooge	Money Tank
67890	Donald Duck	Apple Rd 13
67890	Daisy Duck	Apple Rd 13
32178	Gyro Gearloose	Inventor's lane 1

In view of the above, why do database designers bother to design relation schemas for specific applications?