University of Southern Denmark
Department of Mathematics and Computer Science

DM505
Database Design and Programming

Take-home Exam

Handed out:
15:00 March 31st

Deadline:
15:00 April 1st

Spring 2011

Introduction

You have 24 hours to answer this take-home exam. During this period you are allowed to use books, but not discuss the tasks with anyone. If anything is unclear you are welcome to discuss it with the lecturer.

The weight of each problem is stated in the header of the problem (as a percentage of the total take-home exam). The take-home exam accounts for 50% of your grade, the other 50% is from your project.

Notice, that a 'correct' answer without sufficient argumentation generally will *not* be considered a full solution.

You may answer this exam in Danish or English. And it must be handed in at the secretarys office at IMADA before 15.00 April 1, 2011. You must hand in 2 copies, each maked with name and birthday.[1]

[1] If you write your solution by hand, an extra copy made on a photocopier is fine.

1 Creating an E/R Diagram (15%)

The Chief for an Los Angeles Personel agency has asked you to help with the design of a database for his company, to registrer movies, actors, studios etc.

The system build on top of the database should support the following cases:

- The Personel agency has both Actors and Directors in their stable, and for a premiere, invitations must be sent out to al actors and directors via phone or email.
- The IRS requires the name and the social security number when reporting taxable income.
- The same year, multiple movies having the same name could premiere, but not from the same movie studio. Movies with many actors have bigger premiere parties, than movies with few or no actors.
- For promotion, the personel agency must be able to contact the manager of each movie studio.
- All actors have an artist name, a favorite film type and a minimum daily salary. Directors on the other hand have a filming method and a member number to their organisation (the Directors Guild of America). Actors below the age of 15 are not allowed to appear in movies with a rating of R.

Draw an E/R diagram that handles the cases described. Describe the design choises and the chosen constraints. Describe also if there are something that you cannot model in the E/R diagram. Please use the notation for E/R diagrams from the course book.

2 Converting an E/R Diagram (15%)

Convert the diagram in figure 1 to the relational model. You must describe the process and any constraints that you cannot transfer.

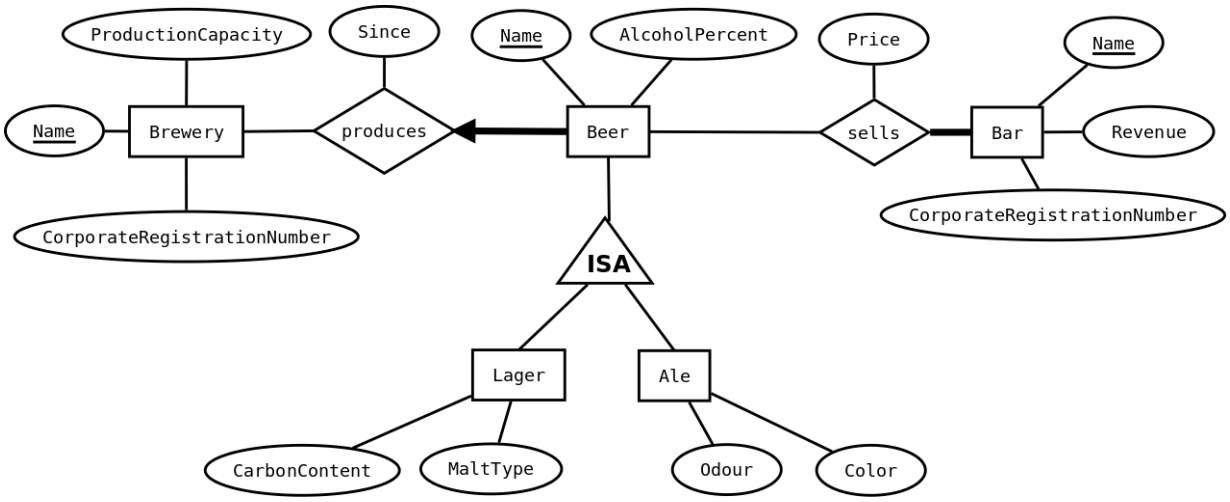

Figure 1: The E/R diagram for problem 2

3 Relational Algebra (10%)

In this question, you have the following relations:

Ingredient(ingredient_id , iname, price)

Recipe(recipe_id, rname, type, portions)

Contains(recipe_id, ingredient_id, amount)

Write the query or draw the query tree that will output the following:

1. A list with ingredient name and amount for the recipe called 'Meatballs'
2. A list of all recipe names which contain only ingredients that cost less than 20
3. Recipe names for recepies that contains the ingredient named 'Spaghetti' and has type 'starter'
4. Recipe names for recipies that contain the ingredient named 'Bacon' but not the ingredient named 'Mushroom'

4 SQL (15%)

Consider again the Recipe relation from question 3:

Recipe(recipe_id, rname, type, portions)

1. Write an SQL statement that creates the table Recipe, with the following constraints:
   (a) The name is a string that is never longer than 64 characters.
   (b) Type is one of { starter, maincourse, dessert }
   (c) portions is an integer number $\in [1, 20]$
2. Write an SQL query that selects all recipe desserts which names starts with 'Ice'
3. Write an SQL update statement that doubles the number of portions for all starters

5 Normalization (15%)

Part 1

Consider the relation **R (A B C D E)** with the functional dependencies:
F := { D → B , E → C, B → A, E → D).

1. Find all candidate keys, and argue why there can be no others.
2. Is R in Boyce-Codd Normal Form? If it is, argue why, if it is not, decompose R until it is in BCNF, and decide if the decomposition is dependency preserving

Part 2

Consider the relation **R (A B C D E)** with the functional dependencies:
F := { B → A , D → C, B → E, D → E).

1. Find all candidate keys, and argue why there can be no others.
2. Is R in Boyce-Codd Normal Form? If it is, argue why, if it is not, decompose R until it is in BCNF, and decide if the decomposition is dependency preserving

6 Tree-based Indexing (10%)

Perform the following operations. Each time you must use the original tree from figure 5. Describe the steps, you do, not just the tree after the operation. If the operation only affect a part of the tree it is fine to just draw the affected part, as long as it is clear which part it is.

1. Insert the entry with key 57
2. Insert the entry with key 118
3. Delete the entry with key 72
4. Delete the entry with key 3

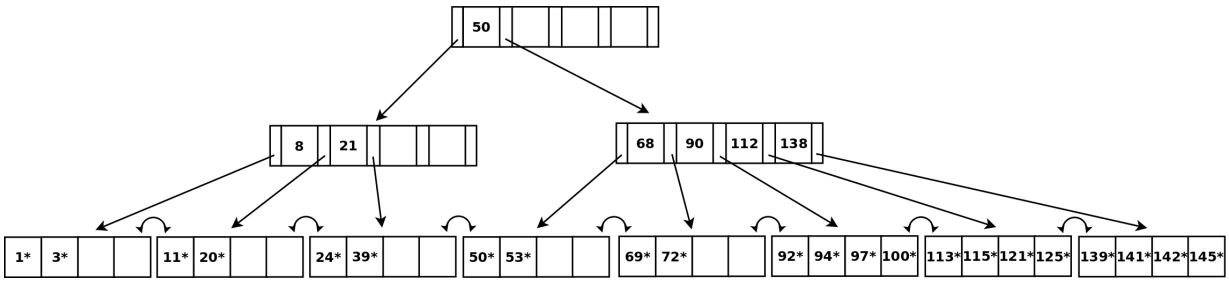

Figure 2: The B+ Tree for Problem 6. It is also placed on the last page of this exam, in a larger version.

7 Hash-based Indexing: Extendible Hashing (10%)

The following hash indices uses the hashfunction $h(x) = x$ for simplicity. That is the hash value of a number, is the number itself.

Figure 3 is an extendible hashing structure. Perform the following operations. Each time you must use the original hash structure from figure 3. Describe the steps, not just the structure after the operation.

1. Insert the key 108
2. Insert the key 57
3. Insert the key 46

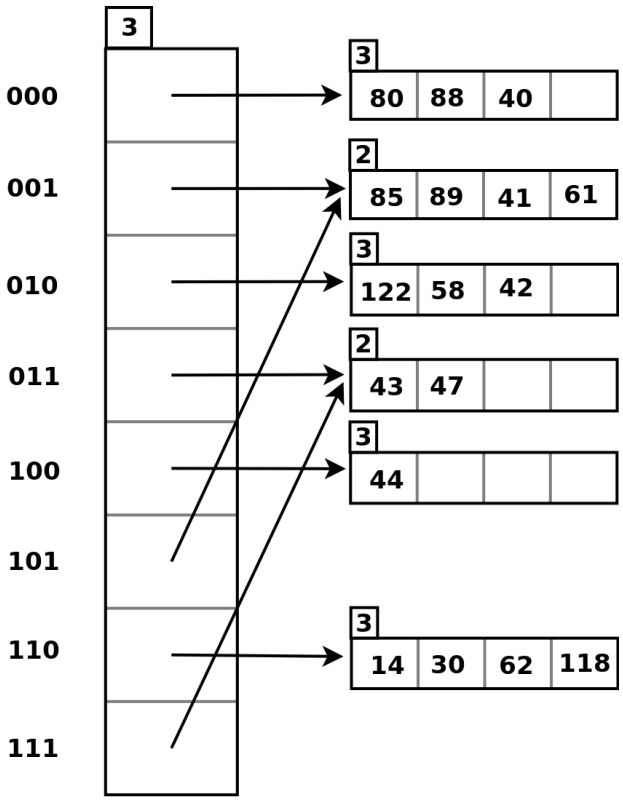

Figure 3: The extendible hashing structure for use in problem 7

8 Hash-based Indexing: Linear Hashing (10%)

The following hash indices uses the hashfunction $h(x) = x$ for simplicity. That is the hash value of a number, is the number itself.

Figure 4 is a linear hashing structure, just starting a new round. Perform the 3 operations below, but continue working on the same structure (i.e. in 2 use the result from 1, and in 3 use the result from 2)

1. Insert the key 91
2. Insert the key 38
3. Insert the key 89

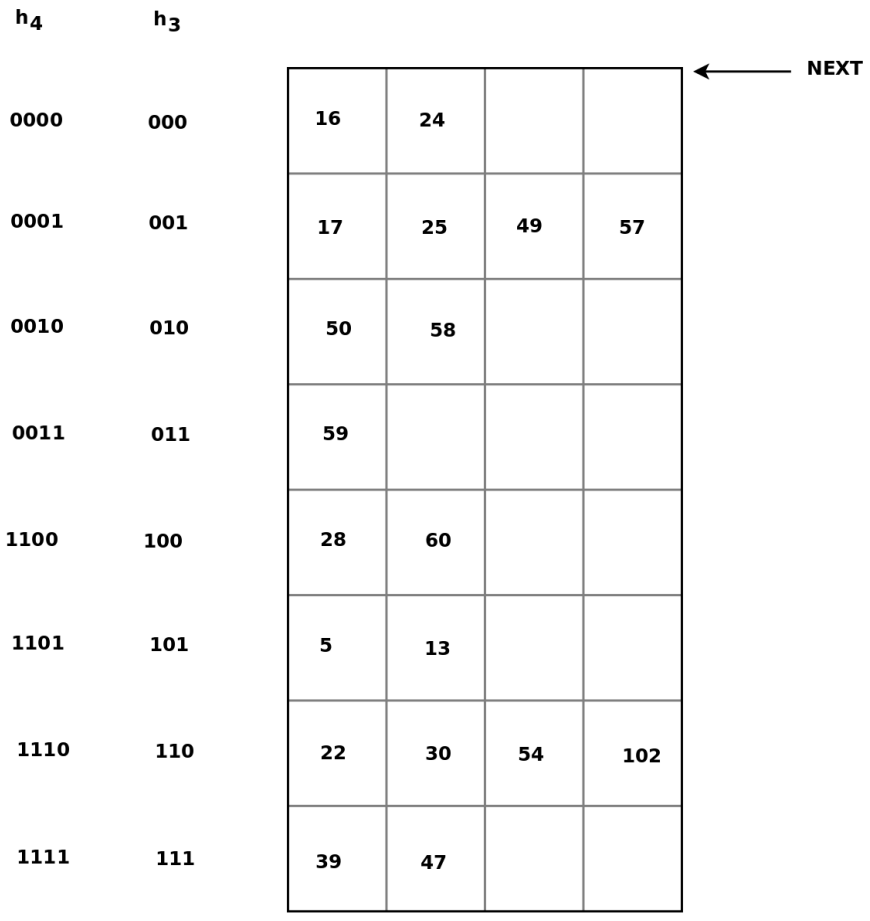

Figure 4: The linear hashing structure for problem 8

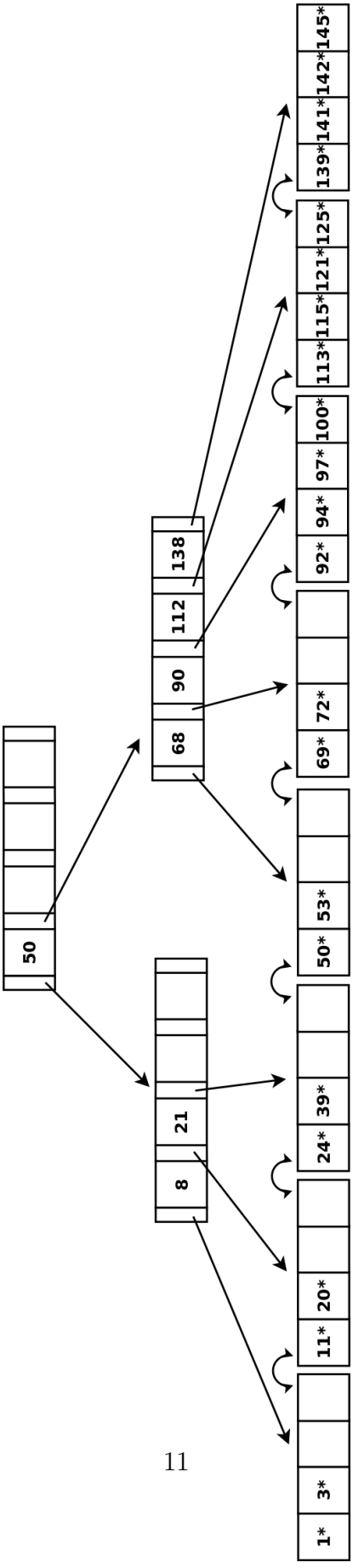

Figure 5: The B+ Tree for Problem 6.