Introduction to Databases

– Mock Exam –

Prof. Werner Nutt

January 19, 2010

- The exam comprises 5 questions some of which consist of several subquestions. There is a total of 60 points that can be achieved in this exam. You will have 2 hours time to answer the questions.
- Please, write down the answers to your questions in the exam booklet handed out to you.
- For drafts use the blank paper provided by the university.
- If the space in the booklet turns out to be insufficient, please use the university paper for additional answers and return them with the booklet.
- For students that submitted assignments for the labs the final mark for the course will be a weighted average of

  $$70\% \text{ exam mark} + 30\% \text{ lab mark}$$

  or the exam mark, whichever is higher. For students that did not submit assignments the final mark will consist solely of the mark for the exam.

1 Relational Database Design

A database needs to be developed that supports a multiplex cinema centre.

(i) Design an entity-relationship diagram that captures, as far as possible, the requirements stated below. If you make any assumptions in your design, please write them down. Assumptions, however, must not contradict the requirements.

1. The multiplex center operates several cinema rooms. A room has a number, a name, and a number of seats. A room is identified by its number.
2. A session takes place in a room on a date at a time. At a given date and time, only one session can take place in a room. During a session, exactly one film is shown.
3. It is possible to make a reservation for a number of seats for a specific session. A reservation is made by a person and backed up a credit card number. The person has to give their name, address and phone number. For a given session, a reservation is uniquely identified by the credit card number.
4. A film has a title and a code. The code identifies the film. The film has been produced in a spefic year and is available in a specific language.

(10 Points)

(ii) Based on the ER-diagram from above, develop a corresponding relational database schema.

List tables with their attributes. For each table, underline the attributes that make up the primary key. Identify referential integrity constraints (i.e., foreign keys).

(5 Points)

2 Relational Algebra Queries

The database of a sailing club contains the following three tables about boats, sailors, and boat reservations made by the sailors.

Boat(<u>bid: `integer`</u>, name: `string`, colour: `string`)

Sailor(<u>id: `integer`</u>, name: `string`, rating: `real`, age: `integer`)

Reservation(<u>bid: `integer`</u>, <u>sid: `integer`</u>, <u>day: `date`</u>).

Attributes belonging to the primary key of a table are underlined.

Express each of the following queries in relational algebra.

(i) Return the names of all red boats.

(2 Points)

(ii) Return the names of the red boats that have been reserved by a sailor of rating 10.

(3 Points)

(iii) Return the names of the red boats that have never been reserved by a sailor of rating 10.

(4 Points)

3 SQL Queries

Consider the relational schema of Question 2. Write SQL queries over this schema that answer the following questions.

(i) How many red boats does the club own?

(2 Points)

(ii) How many of the red boats have been reserved at some point in time by a sailor of rating 10?

(3 Points)

(iii) Return the names of the red boats that have been reserved by at least 20 different people during the course of the year 2006 (that is, between 1 January and 31 December 2006)?

(4 Points)

(iv) Return the names of the boats that have only been reserved by sailors of rating 10?

(5 Points)

(v) Which red boats had the highest number of reservations in 2006?

(3 Points)

4 Query Execution

We consider a database with the sailing club schema of Question 2:

Boat(<u>bid: `integer`</u>, name: `string`, colour: `string`)

Sailor(<u>id: `integer`</u>, name: `string`, rating: `real`, age: `integer`)

Reservation(<u>bid: `integer`</u>, <u>sid: `integer`</u>, <u>day: `date`</u>).

We assume that the database management system created B-tree indexes for the primary key of each relation.

Suppose that queries of the following type are posed frequently

```
SELECT b.name
FROM   Boat b, Reservation r, Sailor s
WHERE  b.bid = r.bid AND
       r.sid = s.sid AND
       b.colour = 'red' AND
       s.age >= 80 AND
       s.rating = 5;
```

where the values of colour, age, and rating are varying.

(i) If you could create as many indexes as you liked to speed up the execution of this query, which ones would you choose?

(3 Points)

(ii) Explain how those indexes would support the execution of the query. Draw a relational algebra tree to represent the optimal execution plan. Annotate each node with the algorithm by which the operation of the node is executed (for example, "index lookup" or "sort merge join").

(5 Points)

(iii) If you could only create a single index, which one would you choose to obtain the greatest speed-up? State the assumptions under which that index would be the best one and explain why that choice would be better than other possible choices.

(3 Points)

5 Normalisation

Consider the following relation that keeps track of the exams taken by students at a University department:

Exam(studID, studName, courseID, courseTitle, acadYear, examSession, mark, degreeCourse)

Suppose the following functional dependencies hold on the relation:

$$
\begin{aligned}
\text{studID} &\rightarrow \text{studName, degreeCourse} \\
\text{courseID} &\rightarrow \text{courseTitle} \\
\text{studID, courseID, acadYear, examSession} &\rightarrow \text{mark} \\
\text{studID, courseID} &\rightarrow \text{acadYear, examSession}
\end{aligned}
$$

Task:

Decompose the relation into smaller relations such that each of the smaller relations is in BNCF with respect to the projection of the original dependencies.

(8 Points)