Introduction to Databases, ITU, Fall 2003

Rasmus Pagh

October 9, 2003

Exercises on October 21

You are unlikely to be able to do all the exercises in two hours. Therefore you are encouraged to prepare at home, and use the exercises for those parts you find difficult.

1. Define a view and write authorization commands that allows one of your fellow students to update information about him/her in the `MovieStar` relation (and to enter the information in the first place). Test your work in Oracle. Does it work as expected?

2. GUW 5.2.1 a), b), c), d), e), f), g). For each relational algebra expression, write a corresponding SQL expression, and try it out in Oracle (sample data can be copied and pasted from the files at `www.itu.dk/people/pagh/IDB03/data/`). Verify that the result agrees with the value of your relational algebra expression (interpreted as relational algebra on bags).

3. Consider the following SQL expressions:

   (a) `(S UNION (SELECT DISTINCT * FROM R WHERE` $A \geq 0$`))`
   `NATURAL JOIN (SELECT DISTINCT * FROM R WHERE` $B = 0$`)`

   (b) `(SELECT DISTINCT * FROM (R NATURAL JOIN S) WHERE` $B = 0$`)`
   `UNION (SELECT DISTINCT * FROM R WHERE (`$B = 0$ `AND` $A \geq 0$`))`

   Convert the expressions into relational algebra. Try to show that (a) and (b) are equivalent. What algebraic laws do you use?

To be handed in no later than October 30, 11.59 AM:

Note that the day for hand-in has changed from Friday to Thursday

This is the third and final in the series of hand-ins in which you have been developing a relational database for a data set of your own choosing. At this point you should have a relational database schema for your database. Please *include the textual description of your database and your relation schema with this hand-in.*

- Determine whether your relation schemas are in BCNF. If not, decompose them into BCNF, indicating which functional dependencies you base the decomposition on.
- Create the relations in Oracle, and enter a little sample data.
- Now it is time for some SQL programming. If your relation schema was already in BCNF do three of the following, otherwise do two:
  - Write an SQL query that suitably joins all your relations:
    - Using a single `SELECT-FROM-WHERE`, and
    - Using nested `SELECT-FROM-WHERE`s with only two parts in the `FROM` list of each.
  - Write two SQL queries that compute interesting aggregates, and define them as views.
  - Give privileges for the user `pagh` to access and/or modify suitable parts of your database. (NB! You do not trust `pagh` too much.)
  - Write two SQL queries that use subqueries in a condition, and define them as views.

You should document your work in Oracle with suitable printouts. For this you may record your database activities into a file, using the command `spool oracle.txt`. Then the terminal output is recorded in the file `oracle.txt` in the Unix file system. To print it, exit SQL*Plus and write the Unix command `a2ps -P`<printer name> `oracle.txt`.

As always, your hand-in must be completed individually.