CS3 Database Systems

Assigment 3. Due Monday 29 November, 2010

Please staple together your answers and put your NAME, ID and COURSE on the first page. Unless the web site says otherwise (please check it) hand your answers into the ITO, Appleton Tower or JCMB, by **4pm on Monday, 29 November, 2010.**

Marking Scheme This assignment counts for 7% of the course (exam + coursework) total. Each of the 5 questions carries equal weight.

1. Consider a variation on the mountains/climbers databases that we used in class:

   Munros(MId, Mname, Long, Lat, Height, Rating, . . .)
   Hikers(HId, HName, Age, Skill, . . .)
   Climbs(MId, HId, Date, Duration)

   and the query

   SELECT DISTINCT Hikers.HName, Hikers.Age
   FROM Munros, Hikers, Climbs
   WHERE Hikers.HId = Climbs.HId AND Climbs.MID = Munros.MId AND Height > 1200

   Translate the query into relational algebra and use algebraic laws of relational algebra to produce a better query plan. Notes:

   - The starting query will be of the form $\pi_{\ldots}(\sigma_{\ldots}(\textsf{Munros} \times \textsf{Hikers} \times \textsf{Climbs}))$
   - Assume that the Munros and Hikers tables have many columns.
   - Give each step as a rewrite of the query, and draw the final evaluation plan as a tree.
   - You may use the "annotated join" ($R \bowtie_{A=B} S$ for $\sigma_{A=B}(R \times S)$) of the textbook if you wish.
   - Be sure to state any assumptions about the selectivity of a condition.

2. Recall that the closure X^+ of an attribute set X with respect to a f.d. set $\mathcal{F}$ is the set $\bigcup\{Y \mid X \rightarrow Y \text{can be inferred from } \mathcal{F}\}$. Prove or disprove (by giving a counter-example) the following:

   (a) $X^+ = Y^+$ when $X \subseteq Y$
   (b) $(X^+)^+ = X^+$.
   (c) $X^+ \cup Y^+ = (X \cup Y)^+$

3. Basic Airways (BA) runs a very simple operation. They run the same flights at the same time every day. Each flight schedule has a flight number N, which determines a departure airport A and a departure time T. Moreover BA has a fleet of identical airplanes all with exactly the same number of seats, so that a booking (a seat S) is determined by a passenger P, a date D and a flight number N.

   (a) Write down a set of functional dependencies that arise from this basic specification
   (b) Provide a losless, dependency preserving decomposition into BCNF (give the keys)
   (c) Add a f.d. that prevents two passengers being placed in the same seat on the same flight. What form is your decomposition now in? Is it possible to achieve a BCNF deomposition?

(d) BA further observes that a passenger cannot be on two different planes at the same time (at least a passenger cannot depart on different planes at the same time). Express this as a functional dependency. What property does your decomposition now violate?

4. You are to provide a design for a university library database. You are to provide both an ER diagram and the SQL DDL. You can assume that you already have a design for a Book entity with key ISBN that has all the relevant information, such as authors, title, publisher etc. So this will appear in your answer as an (incomplete) entity box and table declaration.

   The major entities you need to consider are: Copy – the library keeps several copies of the same book, Client – the people who borrow books. Assume that there are two kinds of Client – Student and Staff, say, who have different privileges. You also need to represent the fact that books can be on loan to clients.

   This question is open-ended in the sense that there are all sorts of attributes that you can have. Please keep this to a basic set. Your design should – as far as possible – help the library in its task of administering loans.

5. Consider tables $R_1(A, B, C)$ and $R_2(C, D, E)$ with the following properties:

   - R_1 has 20,000 tuples, 25 tuples per page; and
   - R_2 has 45,000 tuples, 30 tuples per page.

   Suppose that the buffer size is 100. We want to compute the join $R_1 \bowtie R_2$. Which of the join algorithms (nested-loop join, block nested loop join, sort-merge join or hash-join) requires the least number of page accesses? Provide the estimated number of page accesses for each of these algorithms. Assume that C is a key, or is highly selective, in one of R_1 and R_2.