CS338: Computer Applications in Business: Databases

(Fall, 2009)

Assignment 3 (due on December 1st)

Overview: This assignment consists of four questions.

Assignment submission: All answers to questions should be typed and submitted in class by the end of the lecture on December 1st.

Question 1.

Consider the example ER diagram appearing on slide number 18 of the course slides on the ER model. A "universal relational schema" R for this data description would have an attribute for each ER attribute appearing in the diagram:

CourseNum	CourseName	SectionNum	ProfNum	ProfName	StudentNum	StudentName	Mark

For example, the tuple

[338, 'DBIntro', 2, 99, 'Weddell', 12345, 'Mary', 95]

informs us that "Mary, whose student number in 12345, obtained a grade of 95 in the second section of the DBIntro course, numbered 338, from professor Weddell, whose faculty number is 99."

(a) In a few sentences, discuss some potential problems with R that relate to supporting changes to the database. In particular, outline at least two problems each with inserting new information, changing existing information and deleting existing information.

(b) Give a list F of all the functional dependencies that should hold on R. (Note that the original E-R diagram is a good source of advise on this.)

(c) Apply the Boyce-Codd decomposition algorithm discussed in class (see slide 37 of the course slides on schema evaluation) to obtain a new relational database design for the original E-R diagram. The algorithm should take R and your collection of functional dependencies F derived in Part (b) of this question as input.

Question 2.

Determine whether or not each of the following four transaction execution histories is serializable. If a history is serializable, specify a serial order of transaction execution to which it is equivalent.

$H_1 = r_1[x]\, r_2[y]\, w_2[x]\, r_1[z]\, r_3[z]\, w_3[z]\, w_1[z]$
$H_2 = w_1[x]\, w_1[y]\, r_2[u]\, w_2[x]\, r_2[y]\, w_2[y]\, w_1[z]$
$H_3 = w_1[x]\, w_1[y]\, r_2[u]\, w_1[z]\, w_2[x]\, r_2[y]\, w_1[u]$
$H_4 = w_1[x]\, w_2[u]\, w_2[y]\, w_1[y]\, w_3[x]\, w_3[u]\, w_1[z]$

Question 3.

Consider the following sequence of requests. If the database system uses strict two-phase locking, will any of these requests cause a transaction to block? If so, indicate which will block. Will deadlock occur? Taking blocking into account, give an execution order which could result from these requests. Is your execution order serializable?

T_1: read x
T_2: read x
T_3: write x
T_2: read y
T_1: read y
T_1: read z
T_1: commit
T_4: write z
T_2: commit
T_5: write z
T_4: abort
T_3: commit
T_5: commit

Question 4.

Suppose that after a system failure, the transaction log looks as follows (the log tail is at the bottom):

(T_1,begin)
(T_1,X,0,10)
(T_2,begin)
(T_2,Y,10,20)
(T_3,begin)
(T_1,commit)
(T_3,X,10,400)
(T_4,begin)
(T_3,Z,0,100)
(T_5,begin)
(T_4,A,0,1)
(T_6,begin)
(T_4,abort)
(T_5,A,0,2)
(T_3,commit)
(T_6,X,400,0)
(T_5,commit)
(T_6,A,2,3)

A log entry (T_i,X,a,b) indicates that transaction T_i updated object X, changing its value from a to b.

Describe what the database system must do to recover from the system failure. Indicate which objects must be modified, and in what order those modifications occur. Indicate which transactions are committed and which are aborted after the failure recovery is complete.