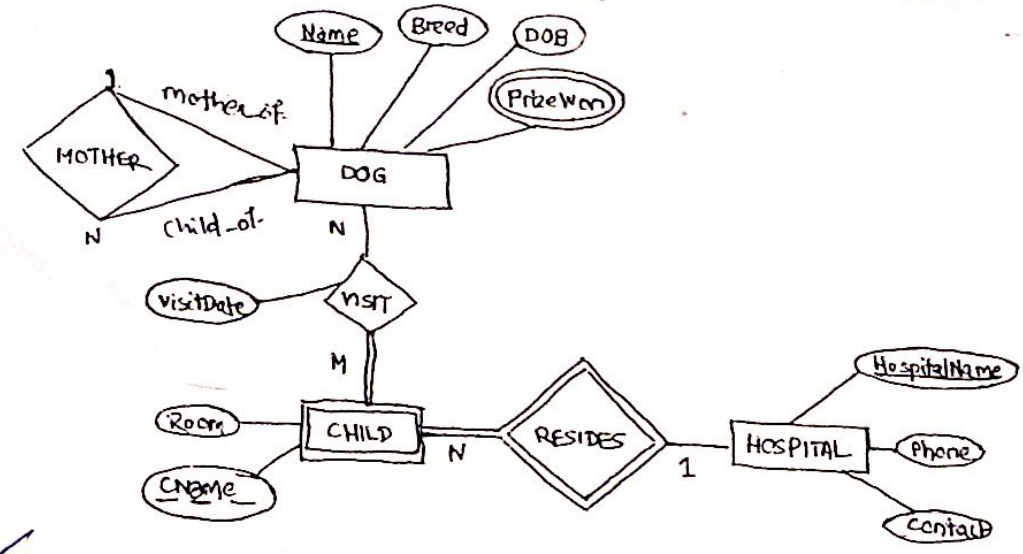
The LNM Institute of Information Technology, Jaipur
CSE-220: Database Management Systems (DBMS)
End-Semester Examination (29th April 2014)

Time: 2 Hours 30 Minutes

Total Marks: 100

Instructions: Answer all the bits of a question in one place.

1. Map the following ER schema into a relational schema and show the primary key(s) and referential integri constraint(s) with appropriate symbols. [10 M]



2. Answer the following. [10 * 2 = 20 M]

a. Given a relation $R(A, B, C, D)$ with functional dependencies $S=\{A\rightarrow BC, B\rightarrow C, A\rightarrow B, AB\rightarrow C, AC\rightarrow D\}$, the minimal cover of S

b. Using Armstrong's inference rules prove or disprove the following inference rules: (i) If $A\rightarrow B$ and C then $AC\rightarrow BD$. (ii) if $A\rightarrow B$ and $B\rightarrow CD$ then $A\rightarrow C$.

3. A student file with *student_ID* as key field includes records with the following *student_ID* values: 9, 6, 2, 8, 10, 7. Suppose the search field values are inserted in the given order in a B+ -tree of order $p=4$ and $p_{leaf}=3$ how the tree will expand and what the final tree looks like. [10 M]

4. Suppose we are using extendible hashing on a file to store the following records in the given order.

Record No.	Search-key value (K)	h(K)
R1	18	10010
R2	5	00101
R3	21	10101
R4	14	01110
R5	22	10110
R6	10	01010
R7	24	11000
R8	12	01100

Show the step-by-step implementation of extendible hashing for this case. Assume the directory buckets initially and each bucket can hold only two records. [10 M]

Figure 1 shows the log corresponding to a particular schedule at the point of a system crash for four transactions *T1*, *T2*, *T3*, and *T4*. Suppose that we use the *immediate update protocol* with checkpointing. Describe the recovery process from the system crash. Specify which transactions are rolled back, which operations in the log are redone and which (if any) are undone, and whether any cascading rollback takes place. Also specify the final value of data items A, B, C and D after recovery. **[15 M]**

[start_transaction, T1]
[read_item, T1, A]
[read_item, T1, D]
[write_item, T1, D, 20, 25]
[commit, T1]
[checkpoint]
[start_transaction, T2]
[read_item, T2, B]
[write_item, T2, B, 12, 18]
[start_transaction, T4]
[read_item, T4, D]
[write_item, T4, D, 25, 15]
[start_transaction, T3]
[write_item, T3, C, 30, 40]
[read_item, T4, A]
[write_item, T4, A, 30, 20]
[commit, T4]
[read_item, T2, D]
[write_item, T2, D, 15, 25]

Figure 1. A sample schedule and its corresponding log

Consider the following two transactions and answer the questions.

T1: R(X), R(Y), W(X), W(Y)
T2: R(X), W(X), R(Y), W(Y)

a. Give an example schedule, consisting of T1 and T2, which is not conflict serializable. Give explanation why your schedule is not conflict serializable. [7 M]

b. In case the system follows 2PL protocol then show whether your schedule is possible or not. If poss then show the corresponding schedule (use *Lock(W)* to lock an item *W*, *Unlock(W)* to unlock an item otherwise explain why it is not possible. [8 M]

Apply the timestamp ordering algorithm to the following schedule *S* and determine whether the algorithm allow the execution of *S*. Consider the transaction id as the timestamp. [10 M]
S = R2(Z), R2(Y), W2(Y), R3(Y), R3(Z), R1(X), W1(X), R2(X), W1(Y), W2(X), C3, C1, C2.
[Hint: Use *Read_TS* and *Write_TS* of each data item]

Consider the following two schedules: [10 M]
S1: R1(X), R3(Z), W1(X), R2(Y), W2(X), R3(X), W4(Y), R1(W), R2(Z), W4(W), R1(Z), C2, C3, C1, C4;
S2: R1(X), W3(Z), W1(X), R2(X), R1(Y), W2(X), R3(Z), C3, W1(Y), C1, C2;
For each of the above schedules determine and explain whether the schedule is (i) conflict serializable (ii) strictest recoverability condition that it satisfies.