CS 33007 Introduction to Database System Design, Spring 2018

Final Exam Solution

Instructions:

- *This examination is closed book (no access to book, lecture notes, phone, laptop, tablet etc.). But you can have notes on* ***one side of a regular paper****.*
- *Please write your answer in the given blank space for each question. If your answer doesn't fit in the given space, you can use back side of the papers but write question number.*

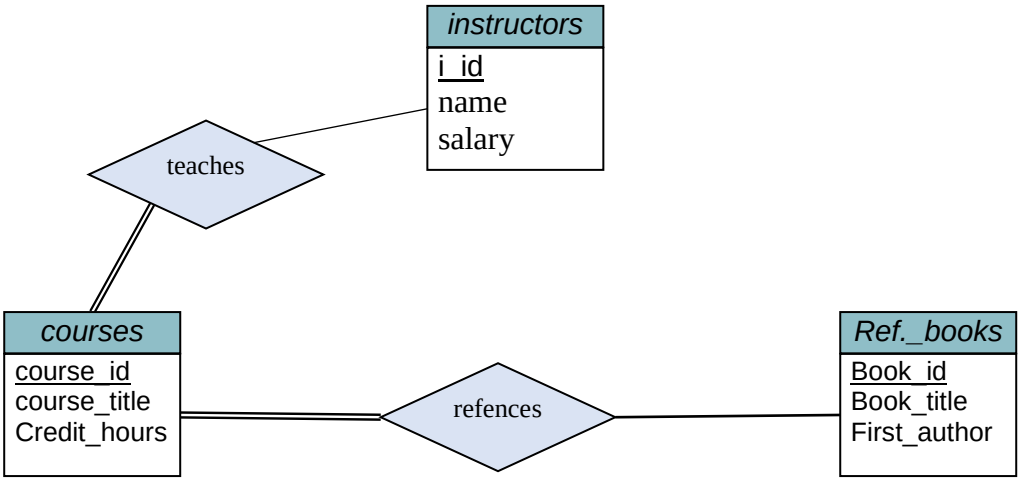
Total Points:100 (25 points for each question) **Time:** 12:45 PM – 3PM

---

1. Suppose computer science department has three entity sets *instructors(i_id*, name, salary), *courses*(*course_id*, course_title, credit_hours) and reference_books(*book_id*, book_title, first_author).

    a. Draw an ER diagram for the entity sets. Please note that each course must have at least one reference book. **[15 points]**

    Answer:



Note for Grader: The diagram would have multiple versions depending on the considerations. If you feel the diagram matches with any real practice, then you give them full points. Otherwise give partial credit if possible.

    b. Convert the ER diagram to non-redundant relation schemas. **[10 points]**

    Answer:

instructors(i_id, name, salary),
courses(*course_id*, course_title, credit_hours)
reference_books(*book_id*, book_title, first_author)
teaches(i_id, course_id)
references(course_id, book_id)

2.

a. "If a relation is in BCNF then it is also in 3NF" - Explain. **[10 points]**

b. Given F = {A → B, C → A, CG → H, CG → I, B → H}, set of functional dependencies on R (A, B, C, G, H, I), check if **CG** is a candidate key of R. **[15 points]**

a) Answer:

A relation schema *R* is in **third normal form (3NF)** if for all: $\alpha \rightarrow \beta$ in F^+ at least one of the following holds:

1. $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \in \alpha$)
2. α is a superkey for *R*
3. Each attribute *A* in $\beta - \alpha$ is contained in a candidate key for *R.*

A relation to be in BCNF, for all: $\alpha \rightarrow \beta$ in F^+ , any of first two conditions needs to be true. This means every BCNF relation is 3NF.

b) **Details Answer: (Note for grader: short but correct answer accepted)**

Calculate $(CG)^+$:
Initialization:
result = CG
Steps in while loop:
1. *result = ACGHI* (C → A, *CG → H, CG → I)*
2. *result = ABCGHI* (*A → B* and *B → H)*

As, $(CG)^+ \supseteq$ R, so CG->R holds, in other words CG is super key of R.

Calculate $(C)^+$ = ABCH, so C->R doesn't hold. C is not a super key.

Calculate $(G)^+$ = G, So G->R doesn't hold, G can't be a super key

So, we can say, CG is the minimal attribute set to be super key for R. So, CG is candidate key for R.

3.

a. How does RAID provide high capacity, speed and reliability? **[13 points]**

b. Give realistic example (one for each) of following mapping cardinality constraints **[12 points]**

i. One to one
ii. One to many
iii. Many to many

a) Answer:

i. High capacity – creating large storage system with many disks.
ii. High speed – allowing parallel data access from multiple disk.
iii. High reliability – By redundant data (storing duplicate data, recovery data etc) which can be used to recover the system from disk failure.

b) Answer:

i. One to one: countries --- capitals
ii. One to many: Mother---children
iii. Many to many: instructors---courses

4.

a. What are the benefits of using index file for a relation? Explain the tradeoff between space overhead and query execution time for using index file. **[12 points]**
b. What is lossless-join decomposition? If a relation *R* = (*ID, name, salary, dept_name, building, budget*) is decomposed into two relations R1(*ID, name, salary, dept_name*) and R2 = (*dept_name, building, budget)* then test whether the decomposition is lossless-join or not. **[13 points]**

a) Answer:

Index file speed up access to desired data by allowing direct access to the associated records.

Large index file can store every possible information to provide faster access to records (e.g. dense index file) while small index file (e.g. sparse index) might enforce some sequential searching of records in main data table which makes the overall query execution time high.

b) Answer:

Lossless-join decomposition: natural joining of the relations obtained from decomposition must return back the original relation.

Here R1 ∩ R2 = "dept_name", which is primary key (i.e superkey) of R2. So, R1 ∩ R2 -> R2 holds. Which is one of the conditions to be satisfied for lossless-join decomposition. In other words, performing natural join between R1 and R2 would always give R because each record in R1 would be able to associate with only one record in R2.