CS 33007 Introduction to Database System Design, Spring 2018

Midterm Solution

Instructions:

- *This examination is closed book (no access to book, lecture notes, phone, laptop, tablet etc.).*
- *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.*
- ***A separate sheet will be provided for university database relational schema.***

Total Points:100 **Time:** 11AM-12:15PM

1. 2. The schema for the faculty relation of an university is,
   faculty (ID, name, salary, dept_name, building, budget)
   (a) Identify the problems caused by the design of this table. **[*13 points*]**
   (b) Propose possible solutions for those problems. **[*12 points*]**

Answer: (Details in Text book, section 1.6.4)

Problems:

i. If a department has more than one instructor, the building name and budget get repeated multiple times. Updates to the building name and budget may get performed on some of the copies but not others, resulting in an inconsistent state where it is not clear what is the actual building name and budget of a department.

ii. A department needs to have at least one instructor for building and budget information to be included in the table.

iii. If all instructors in a department are deleted, the building and budget information are also lost. Ideally, we would like to have the department information in the database irrespective of whether the department has an associated instructor or not, without resorting to null values.

Possible solutions:

- *Null value:* Use null value in case of unavailable information. This solves (ii).
- *Normalization:* in this case you split the relation into two relations as *instructor (ID, name, dept_name, salary)* and *department (dept_name, building, budget).* This will solve all three problems mentioned above.

2.

(i) What is the difference between select and projection operation of relational algebra. Can we mimic select with projection and vice versa? How? ***[10 points]***

(ii) Write an equivalent SQL query for the following expression of relational algebra ***[10 points]***

$$\Pi_{A,\, r.B,\, C,\, r.D,\, E}(\sigma_{r.B = s.B \wedge r.D = s.D}(r \times s))$$

Answer

(i) "select" is used to bring result as set of rows, meaning all attributes from input tables. but projection is to get result with selected attributes from the table. Yes, we can mimic select with projection, but reverse is not true. Projection of all attributes from the table is same as select operation.

(ii) ***select*** *A, B, C, D, E* ***from*** *r* ***natural join*** *s; --- or ---* ***select*** *A, r.B, C, r.D, E* ***from*** *r* ***natural join*** *s;*

3. ***Write queries for the following sentences considering the given relational schema of university database*** ***[30 points]***

(i) Find all courses whose identifier contains substring "CS-1"

Answer: SELECT * **from** course **where** course_id **LIKE** "%CS%";

(ii) Find the name of those departments whose total salary of the instructors is greater than 600000.

Answer: select dept_name, SUM(salary) **as** Total_salary **from** instructor **GROUP BY** dept_name **having** Total_salary>600000;

(iii) Delete instructors with salary less than 60000.

Answer: DELETE from instructor **where** salary<500

(iv) Update the salary of each instructor to 10000 times the number of course sections they have taught.

Answer: `update instructor set salary = 10000*(SELECT COUNT(*) FROM teaches WHERE instructor.ID = teaches.ID)`

4. Using university database relational schema,

(i) write a SQL function that takes nothing as parameter but returns number of records in **cross join** between instructor and department tables. **[15 points]**

```
DELIMITER //
CREATE FUNCTION count_rec() RETURNS integer
BEGIN
DECLARE ct_record integer;
SELECT COUNT(*) INTO ct_record from instructor, department;
RETURN ct_record;
END //
DELIMITER ;
```

(ii) create a view which will allow corresponding users to see only name of the department with building name. **[10 points]**

Answer:

create VIEW dept_budget as **SELECT** dept_name, building **from** department;