**CS157A Spring 2020 Final Exam Name \_Ian SooHoo\_\_ ID \_\_\_\_\_\_\_\_\_\_**

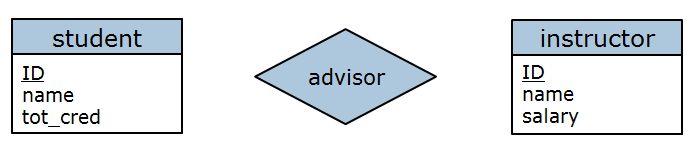
**Please read every question carefully!** Total/Max points: **50** Question 1 – 25 (2 points)

Extra Credit: Question 26-27 (1 point)

Disclaimer: “*You may not consult with other people. You may not use the internet as a reference.  Exam questions posted to chegg will be identified and we will work with the appropriate authorities to obtain account information on anyone posting exam questions.  Doing so risks expulsion from the university.*”

Appendix: Schemas

1. Add lines connecting the relationship set to the entity sets so that:
   * A student can have no advisor, or one advisor
   * An instructor can advise any number of students, including no student



1. Which operator of relational algebra is used when decomposing a schema during normalization? a) select, b) project, c) union. \_\_\_\_a)select\_\_\_\_\_\_\_\_\_\_\_
2. What does it mean for the decomposition of a relation schema into two relation schemas to be “lossless”?
   * every attribute of the original schema appears in one of the decomposed schemas
   * the decomposed schemas have as many rows as the original schema
   * a natural join of the decomposed schemas will give the original schema.
3. Write a functional dependency that proves that the following relation schema is not in Boyce Codd Normal Form:

contribution(candidate\_name, candidate\_id, amount, date)

candidate\_name -> candidate\_id

1. True/False: The right-hand side of a functional dependency must have exactly one attribute. FALSE
2. Fill in the blank: A relation schema is in Boyce Codd Normal Form if the left-hand side of every non-trivial functional dependency for the schema is a \_\_\_ superkey \_\_\_\_\_\_
3. Normalize the following relation schema so that it is in Boyce Codd Normal Form.

treats(doc\_SSN, patient\_SSN, patient\_name)

t1(patient\_SSN , doc\_SSN)

t2(patient\_SSN, patient\_name)

1. Write the primary key of this relation schema as a functional dependency:

Instructor (ID, name, dept\_name)

ID -> name, dept\_name

1. True/False: If a relation schema is in Boyce Codd Normal Form then it is also in Third Normal Form. \_\_\_\_\_\_\_TRUE\_\_\_\_\_\_\_\_\_
2. What does "isolation" mean?
   1. operations of different transactions don't interfere with each other
   2. every schedule contains at most one transaction
   3. either all operations of a transaction complete, or none do
3. Suppose you have written the following query on the campaign data:

select name

from contributor

where zip="93955";

If you wanted to speed up the evaluation of this query, what would be your first choice when creating an index?

* 1. attribute zip
  2. attribute name
  3. attribute contbr\_id (the primary key of the contributor table)

1. What is the difference between a ‘relation instance’ and a ‘table’?
   * a relation instance is a kind of schema
   * a relation instance refers to a subset of rows in a table
   * there is no difference
2. True/False: Every superkey is a candidate key. \_\_\_\_\_\_\_\_\_FALSE\_\_\_\_\_\_\_\_\_\_\_\_
3. An attribute of a relation schema that is the primary key of another relation schema is called a \_\_\_\_\_\_\_\_\_\_\_\_\_ FOREIGN KEY \_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. What is the most that any department spends on total instructor salaries? (output single number)

select max(tot\_sal) from(select sum(salary) as tot\_sal from instructor group by dept\_name);

1. How many instructors in “Biology” have a salary > $80,000?

select \* from instructor where dept\_name="Biology" AND salary >80000;

1. What are the names of students who have taken at least one CS class? Make sure there are no duplicates. The name of the CS department is "Comp. Sci.". Don’t do count

select distinct(name) from student natural join takes where dept\_name="Comp. Sci.";

1. For each department, how many students are in that department? Show departments with the least number of students first.

select dept\_name,count(distinct(STUDENT.ID)) from student natural join takes group by dept\_name;

1. Write JDBC pseudocode to insert couple of rows into a table in secure way.

* Import java sql library
* Connect to DB server/client and port/ user/pass
* Create statement object stmt
* Result = stmt.executeQuery(“insert into department values("Comp. Sci.", "John", 100000)”)
* Use next method
* Result = stmt.executeQuery(“insert into department values("Comp. Sci.", "Mike", 1000000)”)

1. Given a relation: Employees (Employee\_ID, Name, Salary, Manager\_ID)

Write SQL query and equivalent relational algebra expression to find name of each employee’s manager.

Select mgr from (select name as mgr, Employee\_ID from Employees) where Manager\_ID = Employee\_ID;

1. Create a new user FOO and grant System privilege (Create table) to FOO. Also grant update privilege on a table T1 owned by user BAR. Make sure FOO is only allowed to update column C2 of T1. Also, FOO should be able to grant update privilege on T1 to other users.

CREATE USER FOO IDENTIFIED by password;

GRANT create table to FOO;

GRANT update(C2) ON T1 to FOO WITH GRANT OPTION;

1. NoSQL databases are generally (select all that apply)
   1. schema-less
   2. expensive
   3. run on a cluster
   4. difficult to access from a program
2. Create a view named ‘course\_counts’ that gives, for each course\_id, the number of students who have taken (or who are taking) a section of the course. Use the name ‘student\_cnt’ for the attribute giving the number of students.

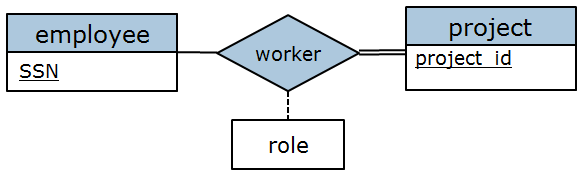
CREATE VIEW course\_counts AS

SELECT course\_id, count(\*) as student\_cnt from student natural join takes group by course\_id;

1. Use the view ‘course\_counts’ of the previous problem in a query that finds the course taken by the fewest number of students. Show the course ID and student count.

select course\_id,min(student\_cnt) from course\_counts;

1. Write relation schemas for the following ER diagram: (be sure to show primary keys)



employee(SSN)

project(project\_Id,SSN)

worker(SSN, project\_id,role)

**Extra Credit**:

1. Suppose we have a dense, clustering index on a table with about a million rows, where the search key is the primary key of the table. If we want to look up a record by its search key, how would we do it?
   1. linear search of the table
   2. linear search of the index, stopping when the key is found
   3. binary search of the index
   4. hash of the search key
2. In sequential file organization:
   1. records are stored in the file strictly by search key order
   2. records are chained together by pointers, in search key order
   3. the files that store the database tables are stored in order on disk

**Appendix.** Schema for the campaign data (all but primary key constraints have been removed)

create table department (

dept\_name varchar(20),

building varchar(15),

budget numeric(12,2)

primary key (dept\_name));

create table course (

course\_id varchar(8),

title varchar(50),

dept\_name varchar(20),

credits numeric(2,0),

primary key (course\_id));

create table instructor (

ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

salary numeric(8,2),

primary key (ID));

create table section (

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

building varchar(15),

room\_number varchar(7),

time\_slot\_id varchar(4),

primary key (course\_id, sec\_id,

semester, year));

create table teaches (

ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

primary key (ID,course\_id,sec\_id,semester,year));

create table student (

ID varchar(5),

name varchar(20) not null,

dept\_name varchar(20),

tot\_cred numeric(3,0,

primary key (ID));

create table takes (

ID varchar(5),

course\_id varchar(8),

sec\_id varchar(8),

semester varchar(6),

year numeric(4,0),

grade varchar(2),

primary key (ID,course\_id,sec\_id,semester,year));

create table time\_slot (