CSCI585 Summer '19 Midterm Exam

June 17th, 2019

CLOSED book and notes. No electronic devices. DO YOUR OWN WORK. Duration: 2 hours. If you are discovered to have cheated in any manner, you will get a 0 and be reported to SJACS. If you continue working on the exam after time is up you will get a 0.

Solutions are in red font.

Signature: ___________________________________

Problem Set	**Number of Points**
Q1	5
Q2	5
Q3	5
Q4	5
Q5	5
Q6	5
Q7	5
Total	**35**

Q1. (5 points total) INTRODUCTION AND DATA MODELING

Using your school's student information system, print your class schedule. The schedule probably would contain the student identification number, student name, class code, class name, class credit hours, class instructor name, the class meeting days and times, and the class room number.

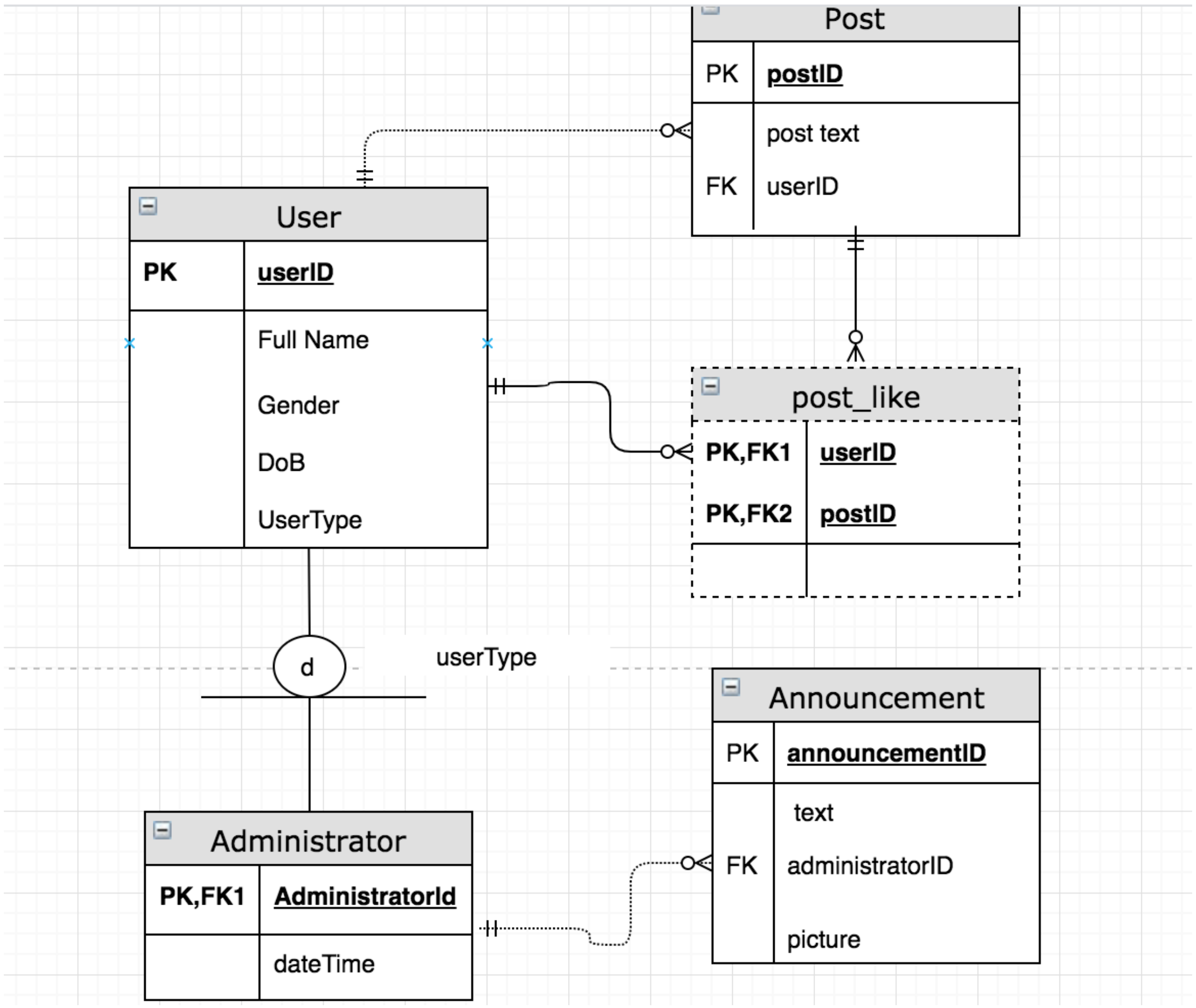
a. Create a spreadsheet using the table shown below and enter your current class schedule.
b. Enter the class schedule of two of your classmates into the same spreadsheet.
c. Discuss the redundancies and anomalies caused by this design.

STU_ ID	STU_NA ME	CLASS_C ODE	CLASS_NA ME	CRED_H RS	INSTR_NA ME	CLASS_D AYS	CLASS_TI MES	ROO M

Students are likely to identify the redundancies around the class information since all three schedules (the student's own schedule plus the schedules of the two classmates) will have at least the database class in common. (1 point)
This leads to discussions of separating the data into at least two tables in a database. (2 points)
However, that still leaves the redundancies of redundant student data with each class that they are taking. Students might realize that a table for student data, a table for class data, and a table to relate the students and classes is appropriate. (2 points)

Q2 (5 points total) ER MODELING

Design ERD using Crow's foot notation for the following description:
A Forum is a website for users to discuss and exchange ideas. Each user has one unique profile including name, gender, and date of birth. A user can be either a normal user or an administrator user. If a normal user become an administrator user, the most recent date of promotion should be recorded.
In the forum, a normal user can make a post with text content. Besides posts that a normal user can make, an administrator can also make an announcement, with both text content and one picture. Only post could be liked by multiple users, and the system would keep track of all the users who like a post.



(1pt) Reasonable entities with PKs defined (e.g. User, Post, Announcement)
(1pt) User-Administrator hierarchical specialization (or similar solution to address data redundancy)

(1pt) Bridge entity to address many to many relationship between user and post
(1pt) correct weak relationships between entities (with FK properly defined)
(1pt) correct strong relationships between entities (with FK properly defined)

Q3. (5 points total) SQL

Write the following queries for an Employee database. Below are the tables for the same, primary keys are underlined, foreign keys are italic. Min_Salary and Max_Salary represent the minimum and maximum salary given to a type of job.

Employees(Employee_Id, First_Name, Last_Name, Hire_Date, *Job_Id*, Salary, *Department_Id*)
Departments (Department_Id,Department_Name)
Jobs(Job_Id, Job_Name, Min_Salary, Max_Salary)
Job_History(*Employee_Id*,Start_Date,End_Date,*Job_Id*,*Department_Id*)

a.(3 points) Write a query to display the job_name, First_Name and Department_Name of all employees who started their jobs before 8 August, 2015.

```
SELECT Job_Name, Department_Name, First_Name , Start_Date
FROM Job_History
WHERE jobs.Job.Id=Job_History.Job_Id AND
Departments.Department_Id=Job_History.Department_Id AND
Employees.Employee_Id=Job_History.Employee_Id AND
start_date<'2015-08-08';
```

b.(2 points) Write a query to display Job_Name , first and last name of employees who make at least $10000 less than maximum salary at their current job.

```
SELECT Job_Name, First_Name, Last_Name
FROM Employees, Jobs
WHERE Employees.Job_Id= Jobs. Job_Id AND
Jobs.max_salary-Employees.salary >10000;
```

Q4. (5 points total) NORMALIZATION

a. (1 point) Write down the highest Normal Form for the following table and explain why.

s_id	Course	hobby
1	Science	Football
1	Math	Tennis
2	Physics	Hockey
2	Php	Baseball

It satisfies 3NF, since the non prime keys (i.e. course and hobby) does not have transitive dependency. (1 point)
Optional explanation: It has multivalued dependency (two records associated with s_id = 1), and therefore does not meet 4NF.

b. (4 points) Convert the following table to 3NF, show or explain the dependency diagram and the primary key of the table(s).

emp_id	emp_name	zipcode	state	city	district
1001	John	282005	UP	Arga	Dayal Bagh
1002	Joseph	222008	TN	Chennai	M-City
1003	Lily	282007	TN	Chennai	Urrapakkam
1004	Steve	292008	UK	Pauri	Bhagwan

Transitive Dependencies:
zipcode -> state, city, district

3NF tables:
(**emp_id**, emp_name, zipcode)
(**zipcode**, state, city, district)

(1 point was deducted for those who created a new table for emp_id, zipcode)

Another Accepted solution:
(**emp_id**, emp_name, zipcode)
(**zipcode**, district)
(**district**,city)
(**city**,state)

Q4. (5 points total) NORMALIZATION – CONTINUED

Please use this as extra space for solving this (normalization) question.

Q5. (5 points total) TRANSACTION MANAGEMENT

Based on a simple relation TA(SID, CID, Stipend) stores TA assignments and stipends. TA's are identified by their student ID's (SID's). Consider the following, if transaction executed by 2 different clients at approximately the same time. Before either transaction starts, there is a tuple (987, 'CS585', 1900) in TA.

T1	T2

Step1:
UPDATE TA
SET Stipend= Stipend + 200
where CID= 'CS585' ;
 	
Step2:
UPDATE TA
SET Stipend= Stipend + 200
where CID= 'CS585' ;
COMMIT
	UPDATE TA
	SET Stipend= 2000
	where Stipend > 2000 ;
	COMMIT

a. (2 points) Suppose T1 and T2 executes with the possibility of interleaving of operations of the two. What could be the possible final stipend values for TA987?

(2 points)
1) 2000 is the result of the execution sequence: T_1.step1, T_1.step2, T_2 (0.5 point)
2) 2200 is the result of the execution sequence: T_1.step1, T_2, T_1.step2 (1 point)
3) 2300 is the result of the execution sequence: T_2, T_1.step1, T_1.step2 (0.5 point)

b. (1 point) Now, suppose instead T1 and T2 executes with isolation level guaranteed. What could be the possible final stipend values for TA987?

(1 point) 2000 and 2300 are still possible, but 2200 is not, because T2 cannot get in between T_1.step1 and T_1.step2 to read the uncommitted value written by T_1.step1. (0.5 point each value)

Q5. (5 points total) TRANSACTION MANAGEMENT - CONTINUED
Consider the following schedule:

T1	T2

	R(A)
	A=A+10
	R(B)
	A=B+10
	W(A)

R(B)
B=B+10
R(A)
B=A+10
W(B)
Commit	

	Commit

c. (1 point) Which of the transaction problems is present in the given schedule and why?

(1 point) Dirty Read/Uncommitted data as transaction T1 is reading the uncommitted data modified by T2.
No points for deadlock/Inconsistent retrieval/ lost update(as both are writing different data values one is writing A and another B)

d. (1 point) Name a technique that can help us avoid the above problem using an example.

(1 point) 2PL Locking. It does that by ensuring serializability of the transactions.

(0.5 point for technique,
0.5 point for explanation/example)

Q6. (5 points total) QUERY OPTIMIZATION

a. (1 point) What are some of the reasons for poor performance of a query?

(1 point) if at least one reasonable answer is listed (answers might vary).
Some of possible answers:
1) No indexes
2) Excess recompilations of stored procedures.
3) Procedures and triggers without SET NOCOUNT ON.
4) Poorly written query with unnecessarily complicated joins
5) Highly normalized database design.
6) Excess usage of cursors and temporary tables.

b. (4 points) Optimize the following two queries and provide explanations for your choice of optimization techniques used

```
SELECT *
FROM employee
WHERE salary != 98000
```

```
SELECT *
FROM employee
WHERE salary > 98000 or salary <98000
```

(1 point for optimized query)

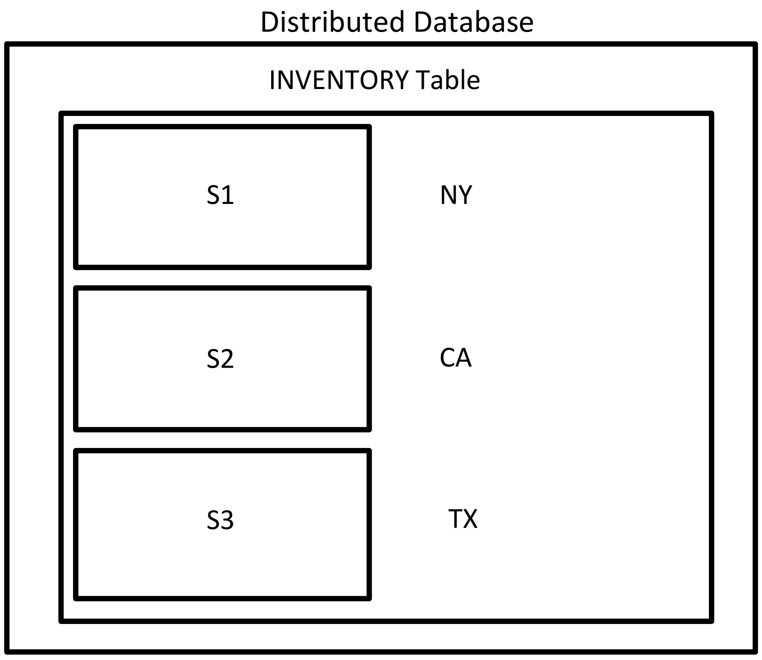
SQL engine prefers '=' instead of '!=' when it comes to query optimization. In order for this optimization to work, It is assumed that there's an index (sorted) on salary column.
(1 point for explanation of issue/solution - a point will be given when issue is correctly identified, even if student couldn't think of a correct solution)

```
SELECT id, name, salary
FROM employee
WHERE salary + 10000 < 35000;
```

```
SELECT id, name, salary
FROM employee
WHERE salary < 25000;
```
(1 point for optimized query)

This query is performing (35000-10000=25000) calculation for every row (as many times as there are rows in the salary table). The solution is to calculate that value only once.
(1 point for explanation of issue/solution - a point will be given when issue is correctly identified, even if student couldn't think of a correct solution)
Q7. (5 points total) DISTRIBUTED DATABASES

Refer to the diagram below and explain what kind of distribution transparency is supported by the database when considering each query. Please write your answer next to each query.



The INVENTORY table is fragmented into 3 parts. Each fragment is stored at one of the nodes residing at 3 locations- NY, CA, TX.

Queries to list all the details of the books whose Quantity On Hand(QOH) is less than 50.

a. (1 point) Fragmentation transparency

SELECT *
FROM INVENTORY
WHERE QOH < 50;

b. (1 point) (Local) Mapping Transparency

SELECT *
FROM S1 NODE NY
WHERE QOH < 50
UNION
SELECT *
FROM S2 NODE CA
WHERE QOH < 50
UNION
SELECT *
FROM S3 NODE TX
WHERE QOH < 50

Q7. (5 points total) DISTRIBUTED DATABASES - CONTINUED

c. (1 point) Location Transparency

SELECT *
FROM S1
WHERE QOH < 50
UNION
SELECT *
FROM S2
WHERE QOH < 50
UNION
SELECT *
FROM S3
WHERE QOH < 50;

Consider the following inventory table of an electronics manufacturing company. Looking at each of the scenarios, suggest a fragmentation strategy to be used.

ID	**Product**	**Cost(USD)**	**QOH**	**State**
21345	Laptop	3000	10	CA
26312	Mobile	800	5	CA

15263	TabletPC	1500	5	TX
17854	Notebook	1900	12	AZ
28896	Laptop	3500	19	TX
95645	Mobile	4000	25	TX
78451	Laptop	700	25	AZ
95512	TabletPC	1800	14	CA

d. (1 point) The marketing team of this company needs to analyse and compare the cost of each of its products with the competitor's products.
(1 point) Vertical Fragmentation.
The attributes needed by the marketing team are ID, Product and Cost(USD).

e. (1 point) The offices at CA, AZ and TX need need to calculate the local sales of each product.
(1 point) Horizontal Fragmentation.
The rows can be divided as per the partition key- State.

SCRATCH PAPER PAGE - Please use this as your scratch paper (not graded).

If you'd like your work on this page graded, please make a note in the question that you're continuing to answer on this page.