CMPS 182, Final Exam, Winter 2016, Shel Finkelstein

Student Name: _____

Student ID: _____

UCSC Email: _____

Final Points

Part	Max Points	Points
I	25	
II	24	
III	15	
IV	36	
Total	100	

Part I: (25 points, 5 each):

For the questions in Part I, assume that there are tables with the following schemas, and that attributes except for grade can't be NULL. Underlined attributes are keys.

Student(studentID, student_name, major)
For each studentID, gives the students name and major.

Course(courseID, course_name, department, instructor)
For each courseID, gives the course's name, department and instructor.

Enroll(studentID, courseID, grade)
For each studentID and courseID where the student has taken (or is taking) that course, gives the student's grade. Grade is a float number if the student has taken the class, and is NULL if the student is still taking the class.

For each question in Part I, write a SQL statement that does what's requested. You may use views that you define, if you wish.

Question 1: Find the instructors for courses that are in either the 'BIO' or the 'CHEM' department. List just the instructors, but don't list any instructor more than once.

Answer 1:

Question 2: Find the names of courses and instructors for which at least one student's grade is NULL. Don't include duplicates in your answer.

Answer 2:

Question 3: For each 'CMPS' department course that has at least 50 students enrolled, list the ID for that course and the number of students in that course.

Answer 3:

Question 4: Find the ID and name for courses where the maximum grade in that course is more than twice the average grade in that course. Order the result by the ID of the course, with the largest ID first and the smallest last.

Answer 4:

Question 5: Write a transaction that changes enrollment for the students who are enrolled in the course named 'CMPS 182' but who are not enrolled in the course named 'CMPS 17'. Your transaction should leave those students in 'CMPS 182', but it should also enroll them in 'CMPS 17', with a grade that's NULL.

For this question, assume that course name is unique.

It's okay to use programming language together with SQL to answer this question, e.g., using stored procedures, host language with SQL embedded, or connection tools/libraries such as JDBC.

Answer 5:

Part II: (24 Points, 4 each)

Question 6: If R(A,B) is a relation where A's domain is (a1, a2, a3) and A can also be NULL, and B's domain is (b1, b2, b3, b4, b5) but B cannot be NULL, what is the maximum number of different tuples that can be in R?

Answer 6: _____

Question 7: A database has the relation Employees, with primary key emp_id, and other attributes giving name, department and salary of an employee.

Employees(emp_id, ename, edept, esalary)

A programmer writes a SQL query intending to find the name, salary and department for employees who make the most money in their departments.

```
SELECT e1.ename, e1.salary, e1.edept
FROM Employees e1
WHERE e1.salary >=
        ( SELECT e2.salary
          FROM Employees e2
          WHERE e1.edept = e2.edept );
```

7a): What's wrong with writing the query this way?

Answer 7a):

7b): How would you fix the query so that it does what the programmer intended?

Answer 7b):

Question 8: What does the Serializability isolation level mean for transactions, and what is the benefit of Serializability?

Answer 8:

Question 9: Assume that relation R(A,B,C,D,E) includes the row (5,4,3,2,1), and that R has Functional Dependencies A,B → C and C,D → E,

Mark each row below with a check mark if it could also be in R, and mark each row with an X if it could not also be in R.

Answer 9:

_____ (5,4,3,3,1)

_____ (5,4,2,2,1)

_____ (1,2,3,4,5)

_____ (6,5,3,2,3)

Question 10: The following tables should be familiar.

Doctors (doctor_id, name, location, specialty)
Medicines (medicine_id, name, price)
Patients (patient_id, name, address, email, doctor_id, admitted)
Prescriptions (prescription_id, doctor_id, medicine_id, patient_id, prescription_date)

The Prescriptions table could have been created by the statement:

```
CREATE TABLE Prescriptions (
   prescription_id INTEGER PRIMARY KEY,
   doctor_id INTEGER,
   medicine_id INTEGER,
   patient_id INTEGER,
   prescription_date CHAR(20)
  );
```

Rewrite this CREATE statement (don't do an ALTER) so that doctor_id, medicine_id and patient_id are Foreign Keys that correspond to the keys of Doctors, Medicines and Patients, respectively. The policies for referential integrity should be:

- Doctors can't be deleted if there are prescriptions from that doctor.
- Deleting a medicine deletes all the prescriptions for that medicine.
- If a patient is deleted, then the patient_id in any prescriptions that were for that patient should become NULL.

(You don't have to deal with what happens if there's an update.)

Answer 10:

Question 11:

11a): Explain in words exactly what the regular expression (in red) for person in the following address book DTD requires.

```
<!DOCTYPE addressbook [
  <!ELEMENT addressbook (person*)>
  <!ELEMENT person
     (name, address+, ( homephone | workphone | mobile )*, email?)>
  <!ELEMENT name          (#PCDATA)>
  <!ELEMENT address       (#PCDATA)>
  <!ELEMENT homephone     (#PCDATA)>
  <!ELEMENT workphone     (#PCDATA)>
  <!ELEMENT mobile        (#PCDATA)>
  <!ELEMENT email         (#PCDATA)>
]>
```

Answer 11a):

11b): Does the following data conform to that DTD? (YES or NO)

```
person>
        <name> MacNiel, John </name>
        <addr> Rome, OH 98765 </addr>
        <addr> Paris, TX 75460 </addr>
        <workphone> (440) 555 1234 </homephone>
        <workphone> (903) 555 9876 </workphone>
        <mobile> (321) 555 2543 </mobile>
</person>
```

Answer 11b): _____

Part III: (15 points, 3 each):

Answer the following questions with **YES** or **NO**.

Question 12: For a database with the following relations, with primary keys underlined:

Employees(emp_id, ename, edept, esalary)
Departments(dept_id, dmanager, daddress)

Are the following two queries always equivalent?

```
SELECT ename, esalary
FROM Employees, Departments
WHERE edept = dept_id
AND esalary > 8000;
```

```
SELECT ename, esalary
FROM Employees
WHERE esalary > 8000
AND EXISTS ( SELECT *
             FROM Departments
             WHERE edept = dept_id );
```

Answer 12: _____

Question 13: Suppose that R and S are relations, and that:

- *condR* is a condition only on attributes of R
- *condS* is a condition only on attributes of S
- *condX* is a condition on attributes of both R and S
- *cond* is (*condR* AND *condS* AND *condX*)

Is the following relational algebra equality always true?

$$\sigma_{cond} (R \times S) = \sigma_{condX} (\sigma_{condR} (R) \times \sigma_{condS} (S))$$

Answer 13: _____

Question 14: If R and S are union-compatible SQL tables with attributes A, B, C, are the following queries always equivalent?

```
( SELECT DISTINCT *                ( SELECT *
  FROM R                             FROM R
  WHERE A = 5 )                      WHERE A = 5 )

UNION ALL                          UNION

( SELECT DISTINCT *                ( SELECT *
  FROM S                             FROM S
  WHERE  B  = 12 )                   WHERE B = 12 )
```

Answer 14: _____

Question 15: Is the following relation with the specified Functional Dependencies in Third Normal Form?

Company_Info(Emp, Dept, Manager)

Emp → Dept

Dept → Manager

Answer 15: _____

Question 16: For OLAP, with a star schema, the number of facts in the Fact Table must always be less than or equal to the product that you get if you multiply together the number of values in the Dimension Tables.

Answer 16: _____

Part IV: (36 points, 6 each):

The questions in Part IV are about the following tables:

Sailors(sid, sname, age, rating) // sailor id, sailor name, age, rating
Boats(bid, bname, color) // boat id, boat name, color of boat
Reserves(sid, bid, day) // sailor id, boat id, date that sid reserved bid.

Question 17: Write a SQL statement that creates the Sailor table with sid as a primary key, sname unique, and a default rating of 'Beginner'. Age can be NULL, but the other attributes can't be NULL. sid and age are integers, and sname and rating are character strings of length 30. Also, age must be between 18 and 90.

Answer 17:

Question 18:

18a): Write a SQL statement that inserts a tuple into the Sailors table for a sailor named Henry whose rating is Shark, whose sid is 678, and whose age is 30.

Answer 18a):

18b): Write a SQL statement that deletes all the reservations made by a sailor named Carol.

Answer 18b):

Question 19:

19a): Write a view MultiBoatSailors that finds the sids of all Sailors who reserved at least two different boats.

Answer 19a):

19b): Using the MultiBoatSailors view, write a query that gives the name of each sailor who reserved at least two boats, together with the name of each of the different boats that they reserved. Each tuple in your result should include the name of a sailor and the name of a boat they reserved.

Answer 19b):

Question 20:

20a): Write a statement that creates an index on Sailors' age and rating attributes.

Answer 20a):

20b): Is that the same as creating an index on Sailors' rating and age attributes? Explain your answer. (Don't bother to create another index.)

Answer 20b):

Question 21: Assume that a JDBC connection myCon has been established to our Sailors/Boats/Reserves database.

Print out all the names of all the boats that the sailor named Henry reserved. Don't bother with including libraries or variable declarations, and you can have an informal print statement if you want.

Here 's an outline of what you need to write:

// Execute the query
// Loop through the results
// For each tuple in the result, get the boat name, and print it

Answer 21:

Question 22: For this question only, let's assume that Sailors also has Functional Dependency **age → level**. Here's an instance of the Sailors table:

Sid	sname	Age	Level
123	John	30	Shark
333	Sam	32	Whale
456	Mary	30	Shark
663	Alan	20	Bass
789	Bob	20	Bass
953	Linda	24	Dolphin

22a): Explain <u>one</u> of the Anomalies that we discussed, identifying the Anomaly and giving an example based on the above instance and Functional Dependency.

Answer 22a):

22b) Could you determine, based on an instance, whether or not the Functional Dependency **age → level** holds for all instances? Explain your answer briefly.

Answer 22b):