UCLA
Computer Science Department
Spring 2011

Instructor: J. Cho

Student Name and ID: ____________________________________

CS143 Midterm: Closed Book, 90 minutes

(IMPORTANT PLEASE READ **):**

- *Simplicity and clarity of your solutions will count.* You may get as few as 0 point for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.
- If you need to make any assumption to solve a question, please write down your assumptions.
- To get partial credits, you may want to write down how you arrived at your answer step by step.
- You may use one-page double-sided cheat-sheet during exam. You are also allowed to use a calculator.
- Please, write your answers neatly. Attach extra pages as needed. Write your name and ID on the extra pages.

Problem	**Score**	
1	13	
2	24	
3	5	
4	9	
Total	45	

Problem 1 (Relational Algebra and SQL): 13 points

1. **Relational Algebra (5 points)** Consider two relations $R(A, B)$ and $S(A, B)$. You would like to compute their intersection $R \cap S$, but unfortunately you only have four relational algebra operators at your disposal: σ, π, $\times$, and $\bowtie$ (natural join). Is it possible to compute $R \cap S$ using just these four operators? If so, show the simplest equivalent expression you can come up with. If not, briefly explain why not.

2. **SQL (8 points)** Consider a table `T(A CHAR(1))`. Based on the SQL92 standard that we learned in the class, write a SQL query to find the largest number of duplicated values in `T`. (For example, if `T` contains three As, five Cs, and four Es, then your query should return the number 5.) Your query will be graded on simplicity as well as on correctness.

Problem 2 (Relational Algebra and SQL): 24 points

Each row of the table below shows two queries. In the blank third column of the table write "YES" if the two queries are equivalent, and "NO" if they are not equivalent. Remember that two queries are equivalent if they always return exactly the same answer on all database instances.

All queries refer to relations R(<u>A</u>,B) and/or S(<u>A</u>,B).

- In both relations R and S, attribute A is a key and attribute B is not a key.
- No attributes are permitted to contain NULL values.
- Do not make any other assumptions about the data.

You will get 2 points for each correct answer, **-1 point for each incorrect answer**, and 0 point for each answer left blank.

Query 1	Query 2	Equiv.?
$\pi_A(R - S)$	$\pi_A(R) - \pi_A(S)$	
$\pi_B(R - S)$	$\pi_B(R) - \pi_B(S)$	
$\pi_B(R \cup S)$	$\pi_B(R) \cup \pi_B(S)$	
$\pi_{R.A,S.A}(R \times S)$	$\pi_A(R) \times \pi_A(S)$	
$\pi_A(R) \cup \rho_{R(A)}(\pi_B(S))$	(SELECT A FROM R) UNION (SELECT B AS A FROM S)	
$\pi_{R.A}(\sigma_{R.B=S.B}(R \times S))$	SELECT R.A FROM R,S WHERE R.B=S.B	
SELECT B FROM R WHERE A IN (SELECT A FROM S)	SELECT R.B FROM R,S WHERE R.A=S.A	
SELECT A FROM R WHERE A NOT IN (SELECT A FROM S)	SELECT R.A FROM R,S WHERE R.A<>S.A	
$\pi_B(R) - \pi_{R1.B}(\sigma_{R1.B \geq R2.B}(\rho_{R1}(R) \times \rho_{R2}(R)))$	SELECT MIN(B) FROM R	
$\pi_B(R) - \pi_{R1.B}(\sigma_{R1.B < R2.B}(\rho_{R1}(R) \times \rho_{R2}(R)))$	SELECT MAX(B) FROM R	
SELECT A FROM R	SELECT A FROM R GROUP BY A	
SELECT B FROM R	SELECT B FROM R GROUP BY B	

Problem 3 (Constraints): 5 points

Consider the following three schema declarations. Assume that all attributes have the same `INT` type, which has been omitted for brevity. Assume that none of the attributes can take a null value, which again has been omitted for brevity.

```
Schema 1: CREATE TABLE R(A PRIMARY KEY, B)
          CREATE TABLE S(C, D REFERENCES R(A))

Schema 2: CREATE TABLE R(A PRIMARY KEY, B)
          CREATE TABLE S(C, D CHECK(D IN (SELECT A FROM R)))

Schema 3: CREATE TABLE R(A PRIMARY KEY, B)
          CREATE TABLE S(C, D)
          CREATE ASSERTION A CHECK(
               NOT EXISTS(SELECT * FROM S
                          WHERE D NOT IN (SELECT A FROM R)))
```

Which of the following statements is true? Circle exactly one.

- All three schemas are equivalent and permit the same operations.
- Schemas 1 and 2 are equivalent but Schema 3 is different.
- Schemas 1 and 3 are equivalent but Schema 2 is different.
- Schemas 2 and 3 are equivalent but Schema 1 is different.
- None of the three schemas are equivalent.

Problem 4 (Views): 9 points

Consider the following two tables. Enrollments table contains information about the number of students who took a course in a given year. Courses table shows the instructor name of a course in a given year.

`Enrollments(year,course,students)` // key is [year,course]
`Courses(course,year,instructor)` // key is [course,year]

Make no assumptions about the tables except for the specified keys. Consider the three view definitions below. Under the SQL92 standard, state whether a given modification for each view:

- ALWAYS generates a view-modification error;
- SOMETIMES (but not always) generates a view-modification error; or
- NEVER generates a view-modification error.

1. View:
```
CREATE VIEW V1 AS
    SELECT E.year, E.course
    FROM Enrollments E, Courses C
    WHERE E.year = C.year AND E.course = C.course
          AND C.instructor = 'Cho'
```
   Modification: `UPDATE V1.year ...`

   Will the above modification cause a view-modification error?

   Circle one: ALWAYS SOMETIMES NEVER

2. View:
```
CREATE VIEW V2 AS
    SELECT year, course
    FROM Enrollments
    WHERE students > 50
    WITH CHECK OPTION
```
   Modification: `INSERT INTO V2 ...`

   Will the above modification cause a view-modification error?

   Circle one: ALWAYS SOMETIMES NEVER

3. View: `CREATE VIEW V3 AS`

```
SELECT course, year
FROM Courses
WHERE year > 2000
WITH CHECK OPTION
```

   Modification: `DELETE FROM V3 ...`

   Will the above modification cause a view-modification error?

   Circle one: ALWAYS SOMETIMES NEVER