# University of California
# Department of Information and Computer Science
# ICS 184 – Database Systems, Winter 2002
# Final Exam
### Max. Points: 100

**Read the following instructions carefully:**

- Total time for the exam is 2 hours. No extra time will be awarded, so budget your time accordingly. The exam is closed books and closed notes.
- The exam is divided into 2 sections *without negative grading*.
  - 10 multiple-choice questions. Each question has only one correct answer. Choosing the correct answer will get you 3 points, an incorrect or empty answer will give you 0 points.
  - 7 other questions.
- If you find ambiguities in a question or in the schema, write down the interpretation you are taking, and then answer the question based on your interpretation.
- This exam contains 16 pages. You can use pages 15-16 as scratch paper.

**NAME:** **ALIAS:**

| Part | Points | Score |
|---|---|---|
| I: 1 – 10 | 30 | |
| II: 11 – 17 | 70 | |
| Total | 100 | |

1. Consider a relation R(A,B), and the following two sequences of queries.

```
Q1:    UPDATE R SET B = 3 WHERE B = 2;

Q2:    INSERT INTO R
       SELECT A, 3 FROM R WHERE B = 2;

       DELETE FROM R WHERE B = 2;
```

Which of the following statements is TRUE?

(A) Q1 and Q2 always produce different answers.
(B) Q1 and Q2 always produce the same answer.
(C) The answer to Q1 is always contained in the answer to Q2
(D) The answer to Q2 is always contained in the answer to Q1.

**ANSWER: (                    )**

2. Suppose we have a relation declared by:

```
CREATE TABLE R(name VARCHAR(50) PRIMARY KEY,
               salary INT CHECK(salary <= 40000) );
```

Initially, the relation has three records:

| Name | Salary |
|---|---|
| Tom | 10000 |
| Joe | 20000 |
| Sue | 30000 |

We execute the following sequence of modifications. Some of them may be rejected due to the constraints in the relation.

(1) INSERT INTO R VALUES ('Fred', 12000);
(2) UPDATE R SET salary = 50000 WHERE name = 'Sue';
(3) INSERT INTO R VALUES (‘Tom’, 13000);
(4) DELETE FROM R WHERE name = 'Joe';

At the end of these statements, the sum of the salaries over all the tuples in R is:

(A) 52,000
(B) 62,000
(C) 65,000
(D) 72,000

**ANSWER: (                    )**

The next two questions are based on the following relations:

Emps(id, name, dept, salary)
Managers(dept, mgr)

The first table gives the employee ID, their name, department, and salary. The second table gives for each department, the manager of that department, which is the employee ID of the person managing the department.

3. We want to constrain the data so that in no department's total salary of its employees is greater than $1,000,000. The following is a framework for an assertion that will enforce this constraint:

CREATE ASSERTION cheap CHECK (NOT EXISTS(**Q**));

Which of the following queries enforces this constraint?

(A)
```
SELECT      *
FROM        Emps
WHERE       salary > 1000000;
```

(B)
```
SELECT      dept, SUM(salary)
FROM        Emps
GROUP BY    dept;
```

(C)
```
SELECT      dept
FROM        Emps
GROUP BY    dept
HAVING      salary > 1000000;
```

(D)
```
SELECT      dept
FROM        Emps
GROUP BY    dept
HAVING      SUM(salary) > 1000000;
```

**ANSWER: (                    )**

4. We wish to constrain the relations so that for each Managers tuple, its **mgr** value must appear as an employee's ID in Emps. Which of the following changes, by itself, *always* enforces this constraint?

(A) In the declaration of Managers, add for attribute mgr an attribute-based check
CHECK (EXISTS (SELECT * FROM Emps WHERE id = mgr))

(B) In the declaration of Emps, add the constraint:
FOREIGN KEY id REFERENCES Managers(mgr)

(C) In the declaration of Managers, add the constraint:
FOREIGN KEY mgr REFERENCES Emps(id)

(D) Both (A) and (C)

**ANSWER: (                    )**

5. Suppose relation R(A,B) has tuples {(1,2), (1,2), (3,4)}, and relation S(B,C) has {(2,5), (2,5), (4,6), (7,8)}. What is the number of tuples in the result of the SQL query:

```
SELECT *
FROM R NATURAL RIGHT OUTER JOIN S;
```

(A) 2
(B) 3
(C) 5
(D) 6

**ANSWER: (          )**

6. Given two relations R(A,B) and S(C), consider two queries:

```
Q1:   SELECT   A
      FROM     R
      WHERE    R.B > ALL (SELECT C FROM S);

Q2:   SELECT   A
      FROM     R
      WHERE    R.B > ANY (SELECT C FROM S);
```

Which of the following statements is TRUE?

(A) Q1 and Q2 always produce the same answer.
(B) The answer to Q1 is always contained in the answer to Q2.
(C) The answer to Q2 is always contained in the answer to Q1.
(D) None of the above.

**ANSWER: (          )**

**7**. Consider a relation R(A) and the following two queries.

Q1:
```
SELECT A
FROM R r1
WHERE NOT EXISTS (SELECT * FROM R WHERE A > r1.A);
```

Q2:
```
SELECT MAX(A) FROM R;
```

Let |Q1| and |Q2| be the number of the tuples in the results of Q1 and Q2, respectively. Which of the following statements is TRUE?

(A) |Q1| = |Q2| is always true.
(B) |Q1| >= |Q2| is always true.
(C) |Q1| <= |Q2| is always true.
(D) None of the above.

**ANSWER: ( )**

8. Consider a relation R(A,B,C,D,E) with FDs: A→B, B→C, AC→D. If we run the algorithm in the class to decompose R to 3NF relations, how many relations will the algorithm generate?

(A) 1
(B) 2
(C) 3.
(D) 4.

**ANSWER: ( )**

The next two questions are based on a relation *R(A, B, C, D, E)* with the following FDs:

AB -> C, B -> D, DE -> A

9. We can show that B -> D is a BCNF violation for R. Suppose we decide to decompose R into $R_1$(B,D) and $R_2$(A,B,C,E) using this violation. Which of the following statement(s) are TRUE?

I. *{AB -> C}* is a minimal set of all FDs that hold in $R_2$.
II. *AB -> C* is a BCNF violation for $R_2$.

(A) I only (B) II only (C) Both I and II (D) Neither I nor II

**ANSWER: ( )**

10. We want to use the BCNF decomposition algorithm covered in class to decompose R to relations in BCNF. Which of the following statements are TRUE?

I. In the first step of the algorithm, instead of decomposing *R* using *B -> D*, we could also decompose *R* using *DE -> A*.
II. It does not matter whether we start with *B -> D* first or *DE -> A* first. At the end of the BCNF decomposition algorithm, we will get the same set of relations

(A) I only (B) II only (C) Both I and II (D) Neither I nor II

**ANSWER: ( )**

**PART II**

**11.** (5 points) To declare a candidate key when creating a table, we can either use **PRIMARY KEY** or **UNIQUE**. Give at least two differences between them.

**12.** (5 points) Consider TRUE (T), FALSE (F), and UNKNOWN (U) as all possible values for variables X, Y, and Z. Use T, F, and U to fill out the five entries in the table.

| **X** | **Y** | **Z** | **(X AND Y) OR (NOT Z)** |
|---|---|---|---|
| U | T | T | |
| U | F | F | |
| U | F | T | |
| U | T | U | |
| U | U | U | |

**13.** (5 points) Given a table Dept(dno, location) and a table created using the statement:

```
CREATE TABLE emp
( name  char(15),
  dno    int,
  FOREIGN KEY dno REFERENCES dept(dno)
        ON DELETE SET NULL
        ON UPDATE CASCADE);
```

**Emp**

| **Name** | **dno** |
|---|---|
| Tom | 111 |
| Mary | 111 |
| Jack | 222 |
| Henry | 222 |

**Dept**

| **Dno** | **Location** |
|---|---|
| 111 | Irvine |
| 222 | LA |
| 333 | SF |

We first execute query:

DELETE * FROM Dept WHERE Dno = 111;

Show the tuples of the two tables after this query.

**After that**, we execute query:

UPDATE Dept SET Dno = 444 WHERE Dno = 222;

Show the tuples of the two tables after this query.

**14.** (15 points) Consider the following schema for a personnel database for a company.

Dept (<u>DNo</u>, name, budget)
Emp (<u>SSN,</u> name, salary, DNo)
Mgr (<u>SSN</u>, assistantSSN)

The company is divided into multiple departments. Each department has a name and a budget, and is uniquely identified by a department number (DNo). Each employee has a name and a salary, and is uniquely identified by a SSN. Each employee works for exactly one department. Certain employees are managers. Managers manage the departments they work for, and each manager has exactly one administrative assistant. Let us say you have the following Create Table commands:

```
CREATE TABLE Dept (Dno integer primary key,
                   name char(30),
                   budget float);

CREATE TABLE Emp (SSN char(9) primary key,
                  name char(30),
                  salary float,
                  DNo integer)

CREATE TABLE Mgr (SSN char(9) primary key,
                  assistantSSN char(9) );
```

(a) (3 pts) Show how to modify the CREATE TABLE commands to encode the referential integrity constraint that "Assistants to managers are also employees."

(b) (3 pts) Write an **attribute-based check** to enforce the constraint "The budget of each department cannot be more than 1000000."

(c) (3 pts) Write a **tuple-based check** to encode constraint: "Those who work in Department 13 have a salary cap of 20,000."

(d) (3 pts) Write an SQL **assertion** to encode the constraint: "If a department has a budget less than $1,000,000, then it must have no more than two managers."

(e) (3 pts) Briefly (using one sentence please) explain the meaning of the following trigger.

```
CREATE TRIGGER FooTrigger
AFTER UPDATE OF salary ON Emp
REFERENCING
  OLD AS OldTuple,
  NEW AS NewTuple
WHEN ((OldTuple.salary > NewTuple.salary) AND
      (OldTuple.name = 'Tom Smith'))
  UPDATE Emp
  SET    salary = OldTuple.salary
  WHERE  SSN = NewTuple.SSN
FOR EACH ROW;
```

**15.** (20 points) Consider the following relations. Each relation has a candidate key including those underlined attribute(s).

Employee (<u>SSN</u>, name, salary, DNo)
Department (<u>DNo</u>, DeptName, MgrSSN)
Project (<u>PNo</u>, location, ProjName)
HourLog (<u>SSN, PNo</u>, hours)

Suppose that each department has only one manager. An employee can be assigned to any number (including zero) projects. Each project has at least one person assigned to it. The HourLog relation lists for each project the number of hours of work for each employee who is assigned to that project. Write the following queries in SQL.

(a) (3 pts) Find the name and SSN of everyone who works more than 100 hours on a project.

(b) (3 pts) Find the SSN of everyone who is not working on any project.

(c) (4 pts) Find the name and SSN of everyone who works for **department number** 10 and also work on **project number** 345.

(d) (5 pts) Find the name and the SSN of everyone who works on at least two projects.

(e) (5 pts) For each project, find the SSN of everyone who works the longest hours for this project.

**16.** (10 points) Give a relation R(A,B,C,D,E) and the following FDs:

$A \rightarrow BC$
$CD \rightarrow E$
$B \rightarrow D$
$E \rightarrow A$

(a) (3 pts) Write down all the candidate keys of the relation.

(b) (4 pts) Decompose R into relations that are in BCNF using the algorithm in class. Show the steps of the decomposition.

(c) (3 pts) Is R in 3NF? Explain your answer.

**17.** (10 points) Consider a relation R(A,B,C,D).

(a) (5 pts) Prove A→→B and B→→ C can imply A →→ C.

(b) (5 pts) Do A→→B, B → → C, and C →D imply C →→ A? If so, give a proof. If no, give a counterexample.

**You may use this page as your scratch paper.**

**You may use this page as your scratch paper.**