CSE 344 Final Examination

December 12, 2012, 8:30am - 10:20am

Name: ____________________________________

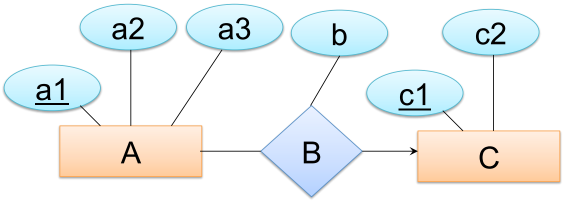
Question	Points	Score
1	30	
2	20	
3	30	
4	20	
Total:	100	

- This exam is open book and open notes but NO laptops or other portable devices.
- You have 1h:50 minutes; budget time carefully.
- Please read all questions carefully before answering them.
- Some questions are easier, others harder; if a question sounds hard, skip it and return later.
- Good luck!

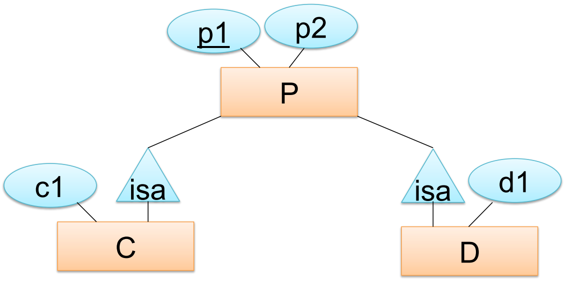
1 E/R Diagrams, Constraints, Conceptual Design

1. (30 points)

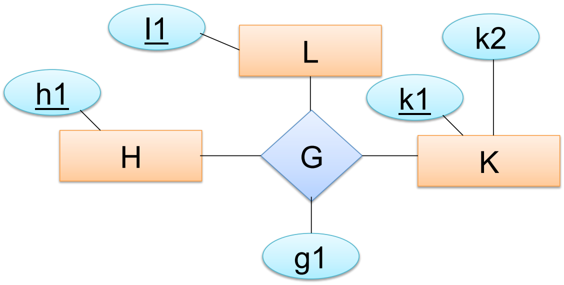
   (a) (10 points) For each E/R diagram, indicate if the CREATE TABLE statement that follows the diagram is consistent with the diagram or not. If it is not consistent, make the necessary changes. Only make changes that are necessary.



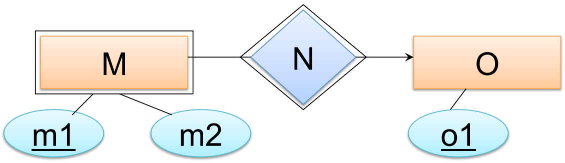
```
CREATE TABLE A (a1 int PRIMARY KEY,
                a2 int,
                b  int,
                c1 int NOT NULL REFERENCES C)
```



```
CREATE TABLE C (p1 int PRIMARY KEY,
                c1 int)
```



```
CREATE TABLE G (h1 int REFERENCES H,
                k1 int REFERENCES K,
                l1 int REFERENCES L,
                g1 int,
                PRIMARY KEY(h1,k1,l1 )
```



```
CREATE TABLE O (o1 int,
                m1 int REFERENCES M,
                PRIMARY KEY(m1,o1))
```

Answer (Corrections, if any, to the above CREATE TABLE statements): Write below or edit directly above.

(b) (10 points) You are working with a customer who just designed his E/R diagram and started to write the following CREATE TABLE statements.

```
CREATE TABLE Product (pid INT PRIMARY KEY,
                      name VARCHAR(20))

CREATE TABLE Inventory(pid INT PRIMARY KEY,
                       quantity INT,
                       FOREIGN KEY (pid) REFERENCES Product)

CREATE TABLE Supplier(sid INT PRIMARY KEY,
                      name VARCHAR(20),
                      address VARCHAR(50))

CREATE TABLE Supplies(sid INT REFERENCES Supplier,
                      pid INT REFERENCES Product,
                      PRIMARY KEY (sid, pid))

CREATE TABLE PurchaseOrder (pid INT REFERENCES Product,
                            quantity INT)
```

The customer would like to add certain constraints to his database. Advise your customer on the appropriate implementation for each of the following constraints. If appropriate, show how to modify the CREATE TABLE statements:

- Ensure that attribute `quantity` from table `Inventory` is always greater than or equal to 0.

  Answer (Your suggestion):

- Whenever someone deletes a tuple in `Supplier`, any tuple in `Supplies` that referred to it should also be deleted.

  <u>Answer</u> (Your suggestion):

- If the quantity of a product in inventory goes down to zero, automatically insert a tuple associated with this product in the `PurchaseOrder` relation (explain your suggestion in English):
  <u>Answer</u> (Your suggestion):

(c) (10 points) Consider the following relational schema and set of functional dependencies.

R(A,B,C,D,E,F,G) with functional dependencies:

A → D
D → C
F → EG
DC → BF

Decompose R into BCNF. Show your work for partial credit. Your answer should consist of a list of table names and attributes and an indication of the keys in each table (underlined attributes).

Answer (Decompose R into BCNF):

2 Transactions

2. (20 points)

   (a) (10 points) Consider the following transaction schedules. For each schedule, indicate if it is **conflict-serializable** or not:

   r1(A); r2(B); r1(B); w2(B); w1(A); w1(B); r2(A); w2(A); c1; c2

   Answer (YES/NO):

   r1(A); r2(B); r3(A); r2(A); r3(C); r1(B); r3(B); r1(C); r2(C); c1; c2; c3

   Answer (YES/NO):

   w1(A); w2(A); w1(B); w3(B); w1(C); w3(C); w2(C); c1; c2; c3

   Answer (YES/NO):

(b) (10 points) Your friend Bob just wrote a Java application that talks to a back-end SQL Server DBMS. The application enables students to register for courses. Looking through the code, you notice that Bob's application performs the following sequence of operations.

```
Prompt the user for a student ID and password.

Start a new transaction.

Look up the student in the database.

If the student ID is not in the database or
the password is incorrect, abort the transaction.

Look up the courses recommended for the student.
Display the courses on the screen.

While the user does not choose QUIT
  Prompt the user to select a course.
  If the course is available, then register the student.
End while

Commit the transaction.
```

Explain the problem in Bob's design. Explain why this is a problem. Explain how to fix the problem. You do not need to write the updated application pseudo-code.

Answer (Be careful to answer all three parts of the question):

3 Parallel Data Processing

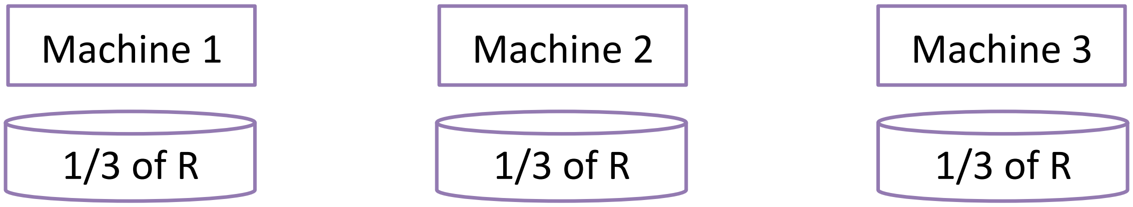
3. (30 points)

   (a) (15 points) Consider a relation R(a,b) that is horizontally partitioned across $N = 3$ machines as shown in the diagram below. Each machine locally stores approximately $\frac{1}{N}$ of the tuples in R. The tuples are randomly organized across machines (i.e., R is block partitioned across machines).

   Show a relational algebra plan for this query and how it will be executed across the $N = 3$ machines. Pick an *efficient* plan that leverages the parallelism as much as possible. Include operators that need to re-shuffle data and add a note explaining how these operators will re-shuffle that data.

```
SELECT a, max(b) as topb
FROM R
WHERE a > 0
GROUP BY a
```

   Answer (Draw the parallel query plan):



(b) (10 points) Explain how the query would be executed in MapReduce (**not Pig**). Make sure to specify the computation performed in the map and the reduce functions.

Answer (Describe the map and reduce functions):

(c) (5 points) What would change if we hash-partitioned `R` on `R.a` *before* executing the above query? Please discuss the case of the parallel DBMS and the case of MapReduce.

Answer (Explain the difference for a parallel DBMS and for MapReduce):

4 NoSQL and Data Integration

4. (20 points)

   (a) (10 points) In order to scale a DBMS, one approach is to horizontally partition each relation across a set of machines in a cluster. Why is it expensive to execute transactions that touch arbitrary data in such a configuration?

   Answer:

(b) (10 points) Jack is looking at customer data from two different branches of his store. Luckily, Jack made sure that the database in each branch followed the same schema. Each branch has a table CustomersBranch holding the ID, name, and state of each customer. He thus integrates the data by defining the following view

```
CREATE VIEW CustomersData AS
SELECT B1.name, B1.state
FROM CustomersBranch1 B1
UNION
SELECT B2.name, B2.state
FROM CustomersBranch2 B2
```

He now wants to analyze this data and he executes the following SQL query:

```
SELECT state, count(*)
FROM CustomersData
GROUP BY state
```

But the results are different from what he expected: Both the number of groups and the counts of customers are much higher than what he expected. Give a possible explanation for each of these two problems..

Answer (Give two reasons):