Stanford University Computer Science Department
2002 Comprehensive Exam in Databases – SAMPLE SOLUTION

1. **(6 points)** Consider an Entity-Relationship diagram with a 3-way relationship R connecting entity sets A, B, and C. In the E/R diagram, there are arrows entering B and C from R, but no arrow entering A. Suppose the number of entities currently in entity sets A, B, and C are a, b, and c, respectively. Also suppose that there are currently r triples in the relationship set for R.

   (a) It must be that $r \leq ab$. Explain why.

   Answer: The arrow into C tells us that for any A-B pair, there can be at most one associated C in the R relationship set. Thus, there can be at most ab triples in that set.

   (b) What other nontrivial constraints on values a, b, c, and r must hold?

   Answer: $r \leq ac$ holds as well. Nothing else that does not follow from $r \leq ab$ and $r \leq ac$ holds.

2. **(12 points)** Suppose we have a relation $R(A, B, C, D, E, F, G, H, I, J)$ with the following nine functional dependencies:

$$\begin{array}{rcl} FI & \rightarrow & C \\ AC & \rightarrow & I \\ EJ & \rightarrow & H \\ GI & \rightarrow & E \\ CJ & \rightarrow & E \\ DE & \rightarrow & B \\ EF & \rightarrow & H \\ BE & \rightarrow & A \\ IJ & \rightarrow & D \end{array}$$

   (a) Compute the closure $(CFGJ)^+$, that is, the set of attributes that must be equal in any two tuples that agree on all of C, F, G, and J.

   Answer: $CEFGHJ$.

   (b) Three of the ten attributes must be in any key. Which three are they, and why? (There is a brief, simple, explanation that does not require you to find all the keys.)

   Answer: F, G, and J must be in every key, since they do not appear on the right side of any FD.

(c) Find the smallest key, and explain why there is no other key of *the same size* or smaller.

Answer: $FGIJ$ is the smallest key. It is easy to check that its closure is all the attributes, so it is a superkey. Every key contains FGJ, but FGJ by itself is not a key — it is its own closure. Thus, at least 4 attributes are necessary, and therefore $FGIJ$ is a key, not just a superkey. We have only to check that there are no other 4-attribute keys. These could only be FGJ and one of A, B, C, D, E, or H. We could check each set, and see that its closure is less than all the attriubutes, but there is a quicker argument. Observe the first two FD's, and note that these are the only ways to add I or C to a closure. But each requires that the other be present in the closure already. Thus, only $CFGJ$ could be a 4-attribute key. But we ruled that out in part (a).

3. **(6 points)** Suppose relation $R(A, B, C, D, E)$ satisfies the multivalued dependencies $A \twoheadrightarrow B$ and $B \twoheadrightarrow D$. Also suppose that R contains the tuples $(0, 1, 2, 3, 4)$ and $(0, 5, 6, 7, 8)$. List the tuples in the smallest possible relation that R could be.

   Answer: The eight tuples described as follows. A is 0. B is 1 or 5. D is 3 or 7. C and E are either 2 and 4, respectively, or 6 and 8, respectively.

4. **(10 points)** In the following, all outerjoins should be assumed "natural."

   (a) If we use the set model of relations, then it is possible to express the full outerjoin of two relations as the union of the left-outerjoin and right-outerjoin of these two relations. Explain why this formula produces the full outerjoin.

   Answer: A tuple in the outerjoin is either (1) in the (inner) join, or (2) it is a padded tuple from the left argument, or (3) it is a padded tuple from the right argument. The left outerjoin gives us (1) and (2), while the right outerjoin gives us (1) and (3). Thus, their union gives us exactly (1), (2), and (3).

   (b) Show that the the formula from (a) does not work if we use the bag (multiset) model of relations.

   Answer: The tuples of the inner join (1) are counted twice when we take the bag union. Thus, as long as the inner join is nonempty, there will be more tuples in the union of the left- and right-outerjoins than in the full outerjoin.

   (c) For the bag model, is there any way to express the full outerjoin in terms of the left- and right-outerjoin and other operations of relational algebra? Either give such a formula or show that none exists.

   Answer: The union of the left- and right-outerjoins, minus the (inner) join has exactly the right number of occurrences of each tuple.

5. **(6 points)** Consider relations $R(A, B, C)$ and $S(A, B, C)$ with no assumptions about keys. For the following two statements using relational algebra, state the constraint or property that

is specified by the given statement. Please be specific about the actual attributes involved. In both cases the correct answer corresponds to a specific database concept and can be stated in a few words. Much longer answers will be considered incorrect.

Note: Symbol $\ltimes$ is the *semijoin* operator. $E_1 \ltimes E_2 \equiv \pi_{schema(E_1)}(E_1 \bowtie E_2)$.

(a) $\pi_A(R) \ltimes \pi_A(S) = \pi_A(R)$

Answer: Referential integrity from $R.A$ to $S.A$

(b) $\sigma_{B1 \neq B2}(\rho_{R1(A,B1,C1)}(R) \bowtie \rho_{R2(A,B2,C2)}(R)) = \emptyset$

Answer: Functional dependency $A \rightarrow B$ in R

6. **(8 points)** Consider the following relations in a SQL database:

```
PhDstud(name,year,email)   // name is a key
Aligned(student,adviser)   // student is a foreign key
                              referencing PhDstud.name
```

Suppose the owner (creator) of these relations is a user named "Kathi," and Kathi wants to grant to a user named "Rajeev" the ability to read all information about all first-year PhD students who are not aligned with an adviser (and only those students). Specify a command or sequence of commands that achieves this goal.

Answer:

```
create view Slackers as
  (select * from PhDstud
   where year = 1
    and name not in (select student from Aligned));
grant select on Slackers to Rajeev;
```

7. **(12 points)** Consider a relation `Visit(inspector,restaurant,date)` recording visits inspectors make to certain restaurants on certain dates. Assume no more than one inspector visits any given restaurant on a given date. Values for the `date` attribute can be compared using `=`, `<`, `<=`, etc., and aggregate functions `min`, `max`, and `count` may be applied to the `date` column. Write a SQL query that returns the two most recent visit dates to each restaurant.

The result of your query should be an ordered two-column relation, where each restaurant appearing in the original `Visit` relation has two adjacent tuples, with the more recent visit for each restaurant listed first. If a restaurant has only one visit, your result should still include a single tuple for that restaurant. For example, we might see the following result, where "Thai Cafe" has only one visit:

restaurant	date
Bytes	4/1/02
Bytes	3/31/02
Coffee House	9/15/01
Coffee House	8/1/00
Thai Cafe	1/10/02
Tresidder	3/10/01
Tresidder	3/9/01

Your SQL query will be graded on simplicity as well as correctness.

Answer:

```
select restaurant, date
from Visit V1
where (select count(*) from Visit V2
       where V2.restaurant = V1.restaurant
       and V2.date >= V1.date) <= 2
order by restaurant, date desc
```