CIS560 – Database System Concepts **Name:_______________**

Homework Assignment 4 [20 points] – due September 27 at 11:59PM

1. [3 points] Consider the relation `PatientOf` and the functional dependencies below. Describe, with examples, how redundancy, update, and delete anomalies can arise.

```
PatientOf (patient_no, name, address, doctor_no, since)

patient_no → name, address
patient_no, doctor_no → since
```

[1 points] *Redundancy anomaly:* A patient can see several doctors and the name and address information of the patient will be duplicated for each patient_no, doctor_no pair, resulting in redundancy.

[1 points] *Update anomaly:* A patient can have multiple doctors, so an update anomaly can happen if a patient's information (corresponding to a patient_no) is changed for only one doctor.

[1 points] *Delete anomaly*: Deleting a patient's information or number could result in the deletion of the doctor_no (if the corresponding row was the only entry for that doctor in the table).

2. [3 points] Consider a relation with attributes R(A, B, C, D, E) that satisfies the following functional dependencies:

```
AB → D
AC → E
BC → D
D  → A
E  → B
```

Find all the keys that contain the attribute A.

AC (and of course all its superkeys ACB, ACD, ACE, ACBD, ACBE, ACDE, ACBDE).

3. [2 points] Use the definition of functional dependencies to prove that if functional dependencies

$A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$ and $C_1, \ldots, C_k \rightarrow D_1, \ldots, D_j$ hold,

then functional dependency

$A_1, \ldots, A_n, C_1, \ldots, C_k \rightarrow B_1, \ldots, B_m, D_1, \ldots, D_j$ also holds.

In the above functional dependency, we should remove one copy of any attribute that appears among both the A's and C's, and among both the B's and D's.

We know that $A_1,\ldots,A_n \rightarrow B_1,\ldots,B_m$ and $C_1,\ldots,C_k \rightarrow D_1,\ldots,D_j$ hold. From the definition of functional dependencies, we know that any two tuples that agree on the values of the attributes $A_1,\ldots,A_n$ will also agree on the values of $B_1,\ldots,B_m$. Similarly, any two tuples that agree on the values of the attributes $C_1,\ldots,C_k$ will also agree on the values of $D_1,\ldots,D_j$.That means, any two tuples that agree on $A_1,\ldots,A_n$ $C_1,\ldots,C_k$ will also agree on $B_1,\ldots,B_m$ $D_1,\ldots,D_j$ – otherwise the given functional dependencies would be violated. Therefore, from the definition of functional dependencies, it follows that $A_1,\ldots,A_n$ $C_1,\ldots,C_k$ -> $B_1,\ldots,B_m$ $D_1,\ldots,D_j$.

4. [4 points – 2 points for finding the decomposition, 2 points for identifying keys] Consider a relation R(A, B, C, D, E) with the following functional dependencies:

$$
\begin{aligned}
A &\rightarrow B \\
A &\rightarrow C \\
BC &\rightarrow A \\
D &\rightarrow E
\end{aligned}
$$

Decompose the relation R into a collection of Boyce-Codd Normal Form (BCNF) relations. For each step, show the relation that you are decomposing and the violation of BCNF that you are using during that decomposition step. Indicate clearly your end result: the relations, their attributes, and their keys.

There are several solutions to this problem, depending on the attributes that are first used to split on. Some sample solutions:

Solution-1:

$A^+ = ABC \neq ABCDE$ => Decompose R into R_1(<u>A</u>,B,C) (BCNF) and R_2(A,D,E) (not BCNF).

$D^+ = DE \neq ADE$ => Decompose R2 into R_{21}(<u>D</u>,E) (BCNF) and R_{22}(<u>D</u>,<u>A</u>) (BCNF).

Solution-2:

BC+=ABC $\neq$ ABCDE => Decompose R into R_1(A,<u>B</u>,<u>C</u>) (BCNF) and R_2(B,C,D,E) (not BCNF).
$D^+ = DE \neq BCDE$ => Decompose R2 into R_{21}(<u>D</u>,E) (BCNF) and R_{22}(<u>D</u>,<u>B</u>,<u>C</u>) (BCNF).

Solution-3:

$D^+ = DE \neq ABCDE$ => Decompose R into R_1(<u>D</u>,E) (BCNF) and R_2(D,A,B,C) (not BCNF).
$A^+ = ABC \neq ABCD$ => Decompose R_2 into R_{21}(<u>A</u>,B,C) (BCNF) and R_{22}(<u>A</u>,<u>D</u>) (BCNF).

5. [4 points] Suppose you are told that `R(A,B,C,D)` is in BCNF, and that 3 out of the 4 FDs listed below hold for R. Choose the FD that R does not satisfy, and explain your reasoning

```
1: A  → BCD
2: BC → A
3: CD → B
4: D  → C
```

The problem states that only 3 out of the given 4 FDs hold for R. So, there are 4 possibilities:

- If 2, 3, 4 are the ones that hold, then D is the only key, and therefore 2 would violate BCNF, so this is not a possibility.
- If 1,3,4 are the ones that hold, then the only key is A. But then both 3 and 4 would be in violation of BCNF, so this is not a good choice either.
- If 1, 2, 4 are the ones that hold, then the keys are A, and BC. But then 4 would violate BCNF.
- If 1, 2, 3 are the ones that hold, then A, BC and CD are all keys, so none of them violates BCNF. Therefore, this is the right choice. So, 4 is the FD that doesn't hold based on the problem description.

(Note that all 4 FDs could hold without violation of BCNF, but the problem definition tells you that only 3 hold, and under this assumption, there is only one possibility as listed above)

6. [4 points] Some of your old high-school friends have opened a new restaurant, and it has become so wildly successful that they need a computer to keep track of dinner reservations. Not knowing all that much about databases, they have created a single table to hold reservation information:

```
RESERVATION (Date, Time, Name, Phone, VIP)
```

Some customers have only made a single reservation, but many of them have multiple reservations in the table. No two customers have the same name and phone number, but some different customers have either the same name or same phone number. VIP is a boolean - true value indicates the very best customers, who receive extra special service.

a. [2 points] What are the possible key(s) and superkeys(s) for this relation? Justify your answer in terms of functional dependencies and closures.

We know that no two customers can have the same name and phone number, but some different customers have either the same name or the same phone number – this means that Name and Phone together identify a customer. Furthermore, every customer has one VIP value (either true or false), which means that Name Phone → VIP. There are no other non-trival dependencies for the given relation. Therefore, the sets {Date,Time,Name,Phone,VIP} and {Date,Time,Name,Phone} are superkeys, because their closures include all of the attributes. However, only {Date,Time,Name,Phone} is a (minimal) key.

b. [2 points] Identify any "bad" functional dependencies in the RESERVATION table and use them to decompose it into relations that are BCNF. Are all the functional dependencies preserved? Explain.

The non-trivial functional dependency Name Phone → VIP violates BCNF because {Name,Phone} is not a superkey for the RESERVATION relation. So we decompose the original RESERVATION table into the following two tables:

RESERVATION1 (<u>Date, Time, Name, Phone</u>)
CUSTOMER (<u>Name, Phone</u>, VIP)

Both RESERVATION1 and CUSTOMER relations are in BCNF. The functional dependency Name Phone → VIP is preserved in the CUSTOMER relation.