Homework 2

1. (10 points) Consider the following relation r with attributes A, B, C, and D.

r			
A	B	C	D
6	2	3	4
1	3	8	5
6	7	8	5

Somehow we know that r has a key consisting of a single attribute. With this clue state which single attribute is a key and which cannot be. In each case informally provide good reasons.

a) Prove that the functional dependency C -> D is satisfied by r. Give the most concise answer you can.

[3 points] The first and third tuples are the only two that agree on C. Since they agree on D, we have C->D.

b) Prove that r does not satisfy C -> B. Give the most concise answer you can.

[3 points] The second and third tuples agree on C, but not on B.

c) Prove that B -> ACD is satisfied by r. Give the most concise answer you can.

[4 points] Each tuple has a unique value on B.

2. (15 points) Given a set of functional dependencies $\mathcal{F}$ = {AG -> B, B -> CD, BD -> E, CE -> F} over R = ABCDEFG.

a) Prove that $\mathcal{F}$ |= AG -> BDF. (This also means that $\mathcal{F}$ logically implies AG -> BDF, or AG -> BDF can be deduced from $\mathcal{F}$).

[5 points] Since B->CD and CD->C and CD->D, we have B->C and B->D (transitivity).
Since AG->B and B->D, we have AG->D (transitivity).

Since B->D, we have BB->BD (augmentation), since BD->E, we have B->E (transitivity).
Since B->C, we have BE->CE (augmentation), since B->E, we have BB->BE (augmentation)
Since BB=B->BE and BE->CE, we have B->CE. Given CE->F, we have B->F

Since AG->B, B->D, and B->F, we have AG->BDF.

b) Compute $(B)^+$, i.e., attribute closure of B

[5 points] (B)+ = {B, C, D, E, F}

c) Find a key of R.

[5 points] Since B->CDEF, AB is a key

3. (20 points) Give a minimal cover for each of the following sets of functional dependencies

a) {A -> B, B -> C, A -> C}

[4 points] Fmin = {A->B, B->C}

b) {ABCD -> CDEF}

[4 points] Fmin = {ABCD->E, ABCD->F}

c) {A -> BC, C -> D}

[4 points] Fmin = {A->B, A->C, C->D}

d) {AB -> CD, A -> B, B -> C}

[4 points] Fmin = {A->D, A->B, B->C}

e) {A -> B, ABCD -> E, EF -> GH, ACDF -> EG}

[4 points] Fmin = {A->B, ACD->E, EF->G, EF->H, ACD->G}

4. (15 points) Prove or disprove the following rules of inference:

a) From XY -> Z infer X -> Z.

[5 points] No.

X	Y	Z
X1	Y1	Z1
X1	Y2	Z2

In this instance, all tuples have unique values on the pair of X and Y, so XY->Z.

However, the two tuples have the same value on X, but different on Z

b) From X -> YZ infer X -> Y

[4 points] Yes. Since X->YZ and YZ->Y(reflexivity), we have X->Y (transitivity)

c) Prove that from {X -> YZ, Y -> W} we cannot infer Y -> Z by giving a two tuple counter example.

[4 points]

X	Y	Z	W
X1	Y1	Z1	W1

X2 Y1 Z2 W1

Note that the the table has unique values on X, so X is a key (i.e., X->YZ, everthing). The two tuples have same value on Y and W, so Y->W. However, their values on Z are different.

5. (15 points) Given a relational schema R with attributes A, B, C, and D, where functional dependencies B -> ACD and C -> D are supposed to hold.

a) What are all the keys in ABCD?

[5 points] B is the only key.

b) Give example of a superkey in ABCD that is not a key.

[4 points] Any set of attributes that contains B, e.g., AB

c) Give example of a trivial functional dependency over ABCD.

[4 points] ABCD->A.

6. (25 points) Given a relational schema R with attributes A, B, C and D where the functional dependencies AB -> C, C -> D, and D -> A are supposed to hold.

a) Is R in BCNF? If yes, explain why. If not, list all violations.

[5 points]

F = {AB->C, C->D, D->A} is a minimum cover.

{AB}+ = {A, B, C, D}, so AB is a super key, AB->C does not violate.
{C}+ = {C, D}, so C is not a super key, so C->D is a violation.
{D}+ = {D, A}, so D is not a super key, so D->A is a violation.

b) If R is NOT in BCNF, give it a lossless BCNF decomposition.

[5 points]

Not in BCNF. Since C->D is a violation, we decompose R(ABCD) into R1(ABC) and R2(CD).

c) Does your decomposition in (b) preserve the given functional dependencies? Explain.

[5 points]

No, D->A is lost

d) Give a 3NF decomposition for R.

[5 points]

Decompose R(ABCD) into R1(ABC), R2(CD), and R3(DA), where R3(AD) is there to ensure dependency D->A can be checked. All are in BCNF, so they are in 3NF too.

e) Does your decomposition in (d) preserve the given functional dependencies? Explain.

[5 points]

Yes, dependency D->A can be checked through R3(AD).