Part 2

ZHANG QIANHAO / 1004654377

> 1. Consider a relation V with attributes $LMNOPQRST$ and functional dependencies W.
> $W = \{LPR \to Q, LR \to ST, M \to LO, MR \to N\}$

(a) State which of the given FDs violate BCNF. attributes never on RHS: M, R, P atrtributes only on RHS: Q, S, T, O, N

closure	closure content
MRP^+	$M, L, O, P, R, N, S, T, Q$

As seen, MRP is a minimum superkey. Since none of the given FDs has a superkey on its LHS, all of them violate BCNF.

(b) Employ the BCNF decomposition algorithm to obtain a lossless and redundancy-preventing decomposition of relation R into a collection of relations that are in BCNF. Make sure it is clear which relations are in the final decomposition, and don't forget to project the dependencies onto each relation in that final decomposition. Because there are choice points in the algorithm, there may be more than one correct answer. List the final relations in alphabetical order (order the attributes alphabetically within a relation, and order the relations alphabetically). Start with $LR \to ST$ $R_1 = LR^+ = LRST$ $R_2 = LMNOPQRST - (LR^+ - LR) = LMNOPQR$ Project FDs on R_1: $LR \to ST$ LR is the key. R_1 is BCNF. on R_2: MRP is still the key. $LPR \to Q$ $M \to LO$ $MR \to LON$ R_2 is not BCNF.

Break with $LPR \to Q$ $R_3 = LPR^+ = LPRQ$ $R_4 = LMNOPQR - (LPR^+ - LPR) = LMNOPR$ Project FDs on R_3: $LPR \to Q$ LPR is the key. R_3 is BCNF. on R_4: MRP is still the key. $M \to LO$ $MR \to N$ R_4 is not BCNF.

Break with $M \to LO$ $R_5 = M^+ = MLO$ $R_6 = LMNOPR - (M^+ - M) = MNPR$ Project FDs on R_5: $M \to LO$ M is the key. R_5 is BCNF. on R_6: MPR is still the key. $MR \to N$ R_6 is not BCNF.

Break with $MR \to N$ $R_7 = MR^+ = MRN$ $R_8 = MNPR - (MR^+ - MR) = MPR$ Project FDs on R_7: $MR \to N$ MR is the key. R_7 is BCNF. on R_8: $MRP \to MRP$ MRP is the key. R_8 is BCNF.

So finally: $R_1(L, R, S, T)$ $R_3(L, P, Q, R)$ $R_5(L, M, O)$ $R_7(M, N, R)$ $R_8(M, P, R)$

> 2. Consider a relation P with attributes $ABCDEFGH$ and functional dependencies T.
> $T = \{AB \to CD, ACDE \to BF, B \to ACD, CD \to AF, CDE \to FG, EB \to D\}$

(a) Compute a minimal basis for T. In your final answer, put the FDs into alphabetical order. Within a single FD, this means stating an FD as $XY \to A$, not as $YX \to A$. Also, list the FDs in alphabetical order ascending according to the left-hand side, then by the right-hand side. This means, $WX \to A$ comes before $WXZ \to A$ which comes before $WXZ \to B$. 1 - split RHS

$AB \to C$ $AB \to D$ $ACDE \to B$ $ACDE \to F$ $B \to A$ $B \to C$ $B \to D$ $CD \to A$ $CD \to F$ $CDE \to F$ $CDE \to G$ $EB \to D$ 2 - reduce LHS $B \to C$ $B \to D$ $ACDE \to B$ $CD \to F$ $B \to A$ $CD \to A$ $CDE \to G$ 3 - reduce FD $B \to C$ $B \to D$ $ACDE \to B$ $CD \to F$ $CD \to A$ $CDE \to G$ Finally: $ACDE \to B$ $B \to C$ $B \to D$ $CD \to A$ $CD \to F$ $CDE \to G$

(b) Using your minimal basis from the last subquestion, compute all keys for P. attributes never on RHS: E atrtributes only on RHS: G

closure	closure content
E^+	E
AE^+	AE
BE^+	$ABCDEFG$
CE^+	CE
DE^+	CE
FE^+	CE
ACE^+	ACE
ADE^+	ADE
AFE^+	AFE
CDE^+	$ABCDEFG$
CFE^+	ADE
DFE^+	DFE
$ACFE^+$	$ACFE$
$ADFE^+$	$ADFE$

As seen not all combinations are computed. This is because: 1 - E must be in the key. 2 - G does not need to be in the key. 3 - Once a key is obtained, all its superset must be a superkey. Finally: BE is a key. CDE is a key.

(c) Employ the 3NF synthesis algorithm to obtain a lossless and dependency-preserving decomposition of relation P into a collection of relations that are in 3NF. Do not “over normalize”. This means that you should combine all FDs with the same left-hand side to create a single relation. If your schema includes one relation that is a subset of another, remove the smaller one.

Minimum basis: $S = \{ ACDE \to B, B \to C, B \to D, CD \to A, CD \to F, CDE \to G, \}$ 1 - corresponding relations $R_1 = ABCDE$ $R_2 = BCD$ $R_3 = ACDF$ $R_4 = CDEG$ 2 - remove redundancy $R_1 = ABCDE$ $R_3 = ACDF$ $R_4 = CDEG$ where both R_1, R_3 contain a superkey. Finally: $R_1(ABCDE)$ $R_3(ACDF)$ $R_4(CDEG)$

(d) Does your schema allow redundancy? Explain how you know that it does or does not. Yes. In principle 3NF synthesis does not guarantee redundancy-free property. It allows attributes (or combination of them) that does not make a superkey determines something else. For instance, $R_1(ABCDE)$ has key BE or CDE. But the FD $CD \rightarrow A$ still holds in R_1. We can identify A knowing BE but A is still there in the relation, which means we have redundancy.