University of Toronto
CSC**343**

Sample 3NF Problem

Questions

Consider a relation R with attributes $ABCDEFGH$ and functional dependencies S:

$$S = \{A \to CD, \quad ACF \to G, \quad AD \to BEF,$$

$$BCG \to D, \quad CF \to AH, \quad CH \to G, \quad D \to B, \quad H \to DEG\}$$

1. Compute a minimal basis for S. In your final answer, put the FDs into alphabetical order.
2. Using the minimal basis from part (b), employ the 3NF synthesis algorithm to obtain a lossless and dependency-preserving decomposition of relation R into a collection of relations that are in 3NF.
3. Does your schema allow redundancy?

Explain all your answers and show your rough work.

Solutions

Note: It is important to do simplification of LHSs before the elimination of redundant FDs Doing these steps in the opposite order is a legitimate approach, and is how it is described in the text, but it requires iterating over these two steps until no more changes are possible. We have learned to do the simplification of LHSs first, and because of that we don't need to iterate.

1. Compute a minimal basis for S. In your final answer, put the FDs into alphabetical order.

   - To find a minimal basis, we'll first simplify FDs to singleton right-hand sides. We'll also number the resulting FDs for easy reference, and call this set $S1$:

1	$A \to C$
2	$A \to D$
3	$ACF \to G$
4	$AD \to B$
5	$AD \to E$
6	$AD \to F$
7	$BCG \to D$
8	$CF \to A$
9	$CF \to H$
10	$CH \to G$
11	$D \to B$
12	$H \to D$
13	$H \to E$
14	$H \to G$

   - The next step is to try reducing the LHS of FDs with multiple attributes on the LHS. For these closures, we will close over the full set $S1$, including the FD being considered for simplification; remember that we are not considering removing the FD, just strengthening it. Note: The order in which we consider attributes to reduce may affect the results we get, but we will always get *a* minimal basis.

     3 $ACF \to G$

     $A^+ = ACDBEFHG$ so we can reduce the LHS to A.

     4 $AD \to B$

     $A^+ = ACDBEFHG$ so we can reduce the LHS to A.

     5 $AD \to E$

     $A^+ = ACDBEFHG$ so we can reduce the LHS to A.

     6 $AD \to F$

     $A^+ = ACDBEFHG$ so we can reduce the LHS to A.

     7 $BCG \to D$

     $B^+ = B$ so we can't reduce the LHS to B.
     $C^+ = C$ so we can't reduce the LHS to C.
     $G^+ = G$ so we can't reduce the LHS to G.
     $BC^+ = BC$ so we can't reduce the LHS to BC.
     $BG^+ = BG$ so we can't reduce the LHS to BG.
     $CG^+ = CG$ so we can't reduce the LHS to CG.
     So this FD remains as it is.

     8 $CF \to A$

     $C^+ = C$ so we can't reduce the LHS to C.
     $F^+ = F$ so we can't reduce the LHS to F.
     So this FD remains as it is.

9 $CF \rightarrow H$

We saw above that $C^+ = C$ and $F^+ = F$, so this FD remains as it is.

10 $CH \rightarrow G$

$C^+ = C$ so we can't reduce the LHS to C.

$H^+ = HDEGB$ so we can reduce the LHS to H.

- Let's call the set of FDs that we have after reducing left-hand sides $S2$:

1	$A \rightarrow C$
2	$A \rightarrow D$
3	$A \rightarrow G$
4	$A \rightarrow B$
5	$A \rightarrow E$
6	$A \rightarrow F$
7	$BCG \rightarrow D$
8	$CF \rightarrow A$
9	$CF \rightarrow H$
10	$H \rightarrow G$
11	$D \rightarrow B$
12	$H \rightarrow D$
13	$H \rightarrow E$
14	$H \rightarrow G$

- Now we'll look for redundant FDs to eliminate. Each row in the table below indicates which of the 14 FDs we still have on hand as we consider removing the next one. Of course, as we do the closure test to see whether we can remove $X \rightarrow Y$, we can't use $X \rightarrow Y$ itself, so an FD is never included in its own row. Again, note that there might be more than one correct result depending on the order of elimination.

FD	Exclude these from $S2$ when computing closure	Closure	Decision
1	1	There's no way to get C without this FD	keep
2	2	$A^+ = ACGBEFDH$	discard
3	2, 3	$A^+ = ACBEFHGD$	discard
4	2, 3, 4	$A^+ = ACEFHGDB$	discard
5	2, 3, 4, 5	$A^+ = ACFHGDEB$	discard
6	2, 3, 4, 5, 6	There's no way to get F without this FD	keep
7	2, 3, 4, 5, 7	$BCG^+ = BCG$	keep
8	2, 3, 4, 5, 8	There's no way to get A without this FD	keep
9	2, 3, 4, 5, 9	There's no way to get H without this FD	keep
10	2, 3, 4, 5, 10	Duplicate FD to (14)	discard
11	2, 3, 4, 5, 10, 11	There's no way to get B without this FD	keep
12	2, 3, 4, 5, 10, 12	$H^+ = HEG$	keep
13	2, 3, 4, 5, 10, 13	There's no way to get E without this FD	keep
14	2, 3, 4, 5, 10, 14	There's no way to get G without this FD	keep

- No further simplifications are possible.
- So the following set $S3$ is a minimal basis (ordered alphabetically):

1 $A \to C$
6 $A \to F$
7 $BCG \to D$
8 $CF \to A$
9 $CF \to H$
11 $D \to B$
12 $H \to D$
13 $H \to E$
14 $H \to G$

2. Using the minimal basis from the previous step, employ the 3NF synthesis algorithm to obtain a lossless and dependency-preserving decomposition of relation R into a collection of relations that are in 3NF.

   - Following the 3NF synthesis algorithm, we would get one relation for each FD. However, we can merge the right-hand sides before doing so. This will yield a smaller set of relations and they will still form a lossless and dependency-preserving decomposition of relation R into a collection of relations that are in 3NF.
   - Let's call the revised FDs $S4$:
     $A \to CF$
     $BCG \to D$
     $CF \to AH$
     $D \to B$
     $H \to DEG$
   - The set of relations that would result would have these attributes:

     $$R1(A, C, F), \quad R2(B, C, D, G), \quad R3(A, C, F, H), \quad R4(B, D), \quad R5(D, E, G, H)$$

   - Since the attributes BD occur within $R2$, we don't need to keep the relation $R4$. Similarly, since the attributes ACF occur in $R3$, we don't need to keep the relation $R1$.
   - A is a key of R, so there is no need to add another relation that includes a key.
   - So the final set of relations is:

     $$R2(B, C, D, G), \quad R3(A, C, F, H), \quad R5(D, E, G, H)$$

3. Does your schema allow redundancy?

   - Because we formed each relation from an FD, the LHS of those FDs are indeed superkeys for their relations. However, there may be other FDs that violate BCNF and therefore allow redundancy. The only way to find out is to project the FDs onto each relation.
   - We can quite quickly find a relation that violates BCNF without doing all the full projections: Clearly $D \to B$ will project onto the relation $R2$. And $D^+ = DB$, so D is not a superkey of this relation.
   - So yes, these schema allows redundancy.