Sample 3NF Problem

Questions

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

S = $\{A \rightarrow CD, ACF \rightarrow G, AD \rightarrow BEF, BCG \rightarrow D, CF \rightarrow AH, CH \rightarrow G, D \rightarrow B, H \rightarrow DEG\}$

1. Compute all keys for R.

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

3. 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.

4. Does your schema allow redundancy?

Explain all your answers and show your rough work.

Solutions

Although one can often skip ahead to some of the conclusions or combine steps, these solutions are very systematic, so that you can see the full pattern.

1. Compute all keys for R.

- Examining all subsets of the attributes would be very time-consuming because there are 2^8 of them. With some careful reasoning we can speed-up the process by avoiding computing many closures.
- By inspection, we can see that $A^+=ACDBEFHG$, which means that A is a key and no superset of A can be a key.
- Also, $CF^+=CFAHGDBE$, which means that CF is a key and no superset of CF can be a key. (C alone or F alone could be part of a key, but CF cannot.)
- But there is no key that has C but not F. We know this because even if we use every other attribute except A (which we know can't be in any other key), we don't have a key: $BCDEGH^+=BCDEGH$.
- Similarly, there is no key that has F but not C. We know this because even if we use every other attribute except A (which we know can't be in any other key), we don't have a key: $BDEFGH^+=BDEFGH$.
- Therefore, the only keys are A and CF.
- A common mistake students make is to consider only the left-hand sides of FDs as possible keys.

This can definitely overlook keys. For example, if the set of FDs S had $C \rightarrow F$ and $F \rightarrow AH$ instead of $CF \rightarrow AH$, CF would be a key even though it never appears as the left-hand side of any FD.

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

- To find a minimal basis, we'll first eliminate redundant FDs. The order in which we do this will affect the results we get, but we will always get a minimal basis.
- We'll simplify to singleton right-hand sides before doing so, since it may be possible to eliminate some but not all of FDs that we get from one of our original FDs. We'll also number the resulting FDs for easy reference, and call this set S1:
  1 $A \rightarrow C$
  2 $A \rightarrow D$
  3 $ACF \rightarrow G$
  4 $AD \rightarrow B$
  5 $AD \rightarrow E$
  6 $AD \rightarrow F$
  7 $BCG \rightarrow D$
  8 $CF \rightarrow A$
  9 $CF \rightarrow H$
  10 $CH \rightarrow G$
  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.

FD	Exclude these from S1 when computing closure	Closure	Decision
1	1	There's no way to get C without this FD	keep
2	2	$A^+=AC$	keep
3	3	$ACF^+=ACFDBEHG$	**discard**
4	3, 4	$AD^+=ADCEFB$ …	**discard**
5	3, 4, 5	$AD^+=ADCFGBE$ …	**discard**
6	3, 4, 5, 6	There's no way to get F without this FD	keep
7	3, 4, 5, 7	$BCG^+= BCG$	keep
8	3, 4, 5, 8	There's no way to get A without this FD	keep
9	3, 4, 5, 9	There's no way to get H without this FD	keep
10	3, 4, 5, 10	$CH^+=CHDEG$…	**discard**
11	3, 4, 5, 10, 11	There's no way to get B without this FD	keep
12	3, 4, 5, 10, 12	$H^+=HEG$	keep
13	3, 4, 5, 10, 13	There's no way to get E without this FD	keep
14	3, 4, 5, 10, 14	There's no way to get G without this FD	keep

- Let's call the remaining FDs S2:
  1 $A \rightarrow C$
  2 $A \rightarrow D$
  6 $AD \rightarrow F$
  7 $BCG \rightarrow D$
  8 $CF \rightarrow A$
  9 $CF \rightarrow H$
  11 $D \rightarrow B$
  12 $H \rightarrow D$
  13 $H \rightarrow E$
  14 $H \rightarrow G$

- Now let's try reducing the LHS of any FDs with multiple attributes on the LHS. For these closures, we will close over the full set S2, including even the FD being considered for simplification; remember that we are not considering removing the FD, just strengthening it.

  6 $AD \rightarrow F$

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

  7 BCG→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 $CF \rightarrow A$

  $C^+=C$ 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.

- Let's call the set of FDs that we have after reducing left-hand sides S3:

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

- Although we've looked at every FD for elimination and have tried simplifying every LHS with multiple attributes, we must look again in case any of the changes we made allow further simplification.

FD	Exclude these from S1 when computing closure	Closure	Decision
1	1	There's no way to get *C* without this FD	keep
2	2	$A^+ = ACFHD\ldots$	**discard**
6'	3, 4, 5, 6	There's no way to get F without this FD	keep
7	3, 4, 5, 7	$BCG^+ = BCG$	keep
8	3, 4, 5, 8	There's no way to get *A* without this FD	keep
9	3, 4, 5, 9	There's no way to get *H* without this FD	keep
11	3, 4, 5, 10, 11	There's no way to get *B* without this FD	keep
12	3, 4, 5, 10, 12	$H^+ = HEG$	keep
13	3, 4, 5, 10, 13	There's no way to get *E* without this FD	keep
14	3, 4, 5, 10, 14	There's no way to get *G* without this FD	keep

How is it possible that we can discard FD 2 now, when we tried and failed to discard it earlier? Because we have FD 6' ($A \rightarrow F$) now instead of FD 6 ($AD \rightarrow F$). This allows a closure to get to *F* from *A* alone, without *D*.

- No further simplifications are possible.
- So the following set S4 is a minimal basis:
  1 $A \rightarrow C$
  6' $A \rightarrow F$
  7 $BCG \rightarrow D$
  8 $CF \rightarrow A$
  9 $CF \rightarrow H$
  11 $D \rightarrow B$
  12 $H \rightarrow D$
  13 $H \rightarrow E$
  14 $H \rightarrow G$

3. 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 S5:
  $A \rightarrow CF$
  $BCG \rightarrow D$
  $CF \rightarrow AH$
  $D \rightarrow B$
  $H \rightarrow DEG$

- The set of relations that would result would have these attributes:
  $R1(A, C, F)$, $R2(B, C, D, G)$, $R3(A, C, F, H)$, $R4(B, D)$, $R5(D, E, G, H)$
- Since the attributes BD occur within $R2$, we don't need to keep the relation $R3$. 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:
  $R2(B, C, D, G)$, $R3(A, C, F, H)$, $R5(D, E, G, H)$

4. 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 $D \rightarrow B$ will project onto the relation $R2$. And $D^+=DB$, so D is not a superkey of this relation.
- So yes, these schema allows redundancy.