

### Exercise 3.1.1

Answers for this exercise may vary because of different interpretations.

Some possible FDs:

Social Security number  $\rightarrow$  name

Area code  $\rightarrow$  state

Street address, city, state  $\rightarrow$  zipcode

Possible keys:

{Social Security number, street address, city, state, area code, phone number}

Need street address, city, state to uniquely determine location. A person could have multiple addresses. The same is true for phones. These days, a person could have a landline and a cellular phone

### Exercise 3.1.2

Answers for this exercise may vary because of different interpretations

Some possible FDs:

ID  $\rightarrow$  x-position, y-position, z-position

ID  $\rightarrow$  x-velocity, y-velocity, z-velocity

x-position, y-position, z-position  $\rightarrow$  ID

Possible keys:

{ID}

{x-position, y-position, z-position}

The reason why the positions would be a key is no two molecules can occupy the same point.

### Exercise 3.1.3a

The superkeys are any subset that contains  $A_1$ . Thus, there are  $2^{(n-1)}$  such subsets, since each of the  $n-1$  attributes  $A_2$  through  $A_n$  may independently be chosen in or out.

### Exercise 3.1.3b

The superkeys are any subset that contains  $A_1$  or  $A_2$ . There are  $2^{(n-1)}$  such subsets when considering  $A_1$  and the  $n-1$  attributes  $A_2$  through  $A_n$ . There are  $2^{(n-2)}$  such subsets when considering  $A_2$  and the  $n-2$  attributes  $A_3$  through  $A_n$ . We do not count  $A_1$  in these subsets because they are already counted in the first group of subsets. The total number of subsets is  $2^{(n-1)} + 2^{(n-2)}$ .

### Exercise 3.1.3c

The superkeys are any subset that contains  $\{A_1, A_2\}$  or  $\{A_3, A_4\}$ . There are  $2^{(n-2)}$  such subsets when considering  $\{A_1, A_2\}$  and the  $n-2$  attributes  $A_3$  through  $A_n$ . There are  $2^{(n-2)} - 2^{(n-4)}$  such subsets when considering  $\{A_3, A_4\}$  and attributes  $A_5$  through  $A_n$  along with the individual attributes  $A_1$  and  $A_2$ . We get the  $2^{(n-4)}$  term because we have to discard the subsets that contain the key  $\{A_1, A_2\}$  to avoid double counting. The total number of subsets is  $2^{(n-2)} + 2^{(n-2)} - 2^{(n-4)}$ .

### Exercise 3.1.3d

The superkeys are any subset that contains  $\{A_1, A_2\}$  or  $\{A_1, A_3\}$ . There are  $2^{(n-2)}$  such subsets when considering  $\{A_1, A_2\}$  and the  $n-2$  attributes  $A_3$  through  $A_n$ . There are  $2^{(n-3)}$  such subsets when considering  $\{A_1, A_3\}$  and the  $n-3$  attributes  $A_4$  through  $A_n$ . We do not count  $A_2$  in these subsets because they are already counted in the first group of subsets. The total number of subsets is  $2^{(n-2)} + 2^{(n-3)}$ .

### Exercise 3.2.1a

We could try inference rules to deduce new dependencies until we are satisfied we have them all. A more systematic way is to consider the closures of all 15 nonempty sets of attributes.

For the single attributes we have  $\{A\}^+ = A$ ,  $\{B\}^+ = B$ ,  $\{C\}^+ = ACD$ , and  $\{D\}^+ = AD$ . Thus, the only new dependency we get with a single attribute on the left is  $C \rightarrow A$ .

Now consider pairs of attributes:

$\{AB\}^+ = ABCD$ , so we get new dependency  $AB \rightarrow D$ .  $\{AC\}^+ = ACD$ , and  $AC \rightarrow D$  is nontrivial.  $\{AD\}^+ = AD$ , so nothing new.  $\{BC\}^+ = ABCD$ , so we get  $BC \rightarrow A$ , and  $BC \rightarrow D$ .  $\{BD\}^+ = ABCD$ , giving us  $BD \rightarrow A$  and  $BD \rightarrow C$ .  $\{CD\}^+ = ACD$ , giving  $CD \rightarrow A$ .

For the triples of attributes,  $\{ACD\}^+ = ACD$ , but the closures of the other sets are each  $ABCD$ . Thus, we get new dependencies  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ , and  $BCD \rightarrow A$ .

Since  $\{ABCD\}^+ = ABCD$ , we get no new dependencies.

The collection of 11 new dependencies mentioned above are:

$C \rightarrow A$ ,  $AB \rightarrow D$ ,  $AC \rightarrow D$ ,  $BC \rightarrow A$ ,  $BC \rightarrow D$ ,  $BD \rightarrow A$ ,  $BD \rightarrow C$ ,  $CD \rightarrow A$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ , and  $BCD \rightarrow A$ .

### Exercise 3.2.1b

From the analysis of closures above, we find that  $AB$ ,  $BC$ , and  $BD$  are keys. All other sets either do not have  $ABCD$  as the closure or contain one of these three sets.

### Exercise 3.2.1c

The superkeys are all those that contain one of those three keys. That is, a superkey that is not a key must contain  $B$  and more than one of  $A$ ,  $C$ , and  $D$ . Thus, the (proper) superkeys are  $ABC$ ,  $ABD$ ,  $BCD$ , and  $ABCD$ .

### Exercise 3.2.2a

i) For the single attributes we have  $\{A\}^+ = ABCD$ ,  $\{B\}^+ = BCD$ ,  $\{C\}^+ = C$ , and  $\{D\}^+ = D$ . Thus, the new dependencies are  $A \rightarrow C$  and  $A \rightarrow D$ .

Now consider pairs of attributes:

$\{AB\}^+ = ABCD$ ,  $\{AC\}^+ = ABCD$ ,  $\{AD\}^+ = ABCD$ ,  $\{BC\}^+ = BCD$ ,  $\{BD\}^+ = BCD$ ,  $\{CD\}^+ = CD$ . Thus the new dependencies are  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $AC \rightarrow B$ ,  $AC \rightarrow D$ ,  $AD \rightarrow B$ ,  $AD \rightarrow C$ ,  $BC \rightarrow D$  and  $BD \rightarrow C$ .

For the triples of attributes,  $\{BCD\}^+ = BCD$ , but the closures of the other sets are each  $ABCD$ . Thus, we get new dependencies  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ , and  $ACD \rightarrow B$ .

Since  $\{ABCD\}^+ = ABCD$ , we get no new dependencies.

The collection of 13 new dependencies mentioned above are:

$A \rightarrow C$ ,  $A \rightarrow D$ ,  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $AC \rightarrow B$ ,  $AC \rightarrow D$ ,  $AD \rightarrow B$ ,  $AD \rightarrow C$ ,  $BC \rightarrow D$ ,  $BD \rightarrow C$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$  and  $ACD \rightarrow B$ .

ii) For the single attributes we have  $\{A\}^+ = A$ ,  $\{B\}^+ = B$ ,  $\{C\}^+ = C$ , and  $\{D\}^+ = D$ . Thus, there are no new dependencies.

Now consider pairs of attributes:

$\{AB\}^+ = ABCD$ ,  $\{AC\}^+ = AC$ ,  $\{AD\}^+ = ABCD$ ,  $\{BC\}^+ = ABCD$ ,  $\{BD\}^+ = BD$ ,  $\{CD\}^+ = ABCD$ . Thus the new dependencies are  $AB \rightarrow D$ ,  $AD \rightarrow C$ ,  $BC \rightarrow A$  and  $CD \rightarrow B$ .

For the triples of attributes, all the closures of the sets are each  $ABCD$ . Thus, we get new dependencies  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ACD \rightarrow B$  and  $BCD \rightarrow A$ .

Since  $\{ABCD\}^+ = ABCD$ , we get no new dependencies.

The collection of 8 new dependencies mentioned above are:

$AB \rightarrow D$ ,  $AD \rightarrow C$ ,  $BC \rightarrow A$ ,  $CD \rightarrow B$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ACD \rightarrow B$  and  $BCD \rightarrow A$ .

iii) For the single attributes we have  $\{A\}^+ = ABCD$ ,  $\{B\}^+ = ABCD$ ,  $\{C\}^+ = ABCD$ , and  $\{D\}^+ = ABCD$ . Thus, the new dependencies are  $A \rightarrow C$ ,  $A \rightarrow D$ ,  $B \rightarrow D$ ,  $B \rightarrow A$ ,  $C \rightarrow A$ ,  $C \rightarrow B$ ,  $D \rightarrow B$  and  $D \rightarrow C$ .

Since all the single attributes' closures are  $ABCD$ , any superset of the single attributes will also lead to a closure of  $ABCD$ . Knowing this, we can enumerate the rest of the new dependencies.

The collection of 24 new dependencies mentioned above are:

$A \rightarrow C$ ,  $A \rightarrow D$ ,  $B \rightarrow D$ ,  $B \rightarrow A$ ,  $C \rightarrow A$ ,  $C \rightarrow B$ ,  $D \rightarrow B$ ,  $D \rightarrow C$ ,  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $AC \rightarrow B$ ,  $AC \rightarrow D$ ,  $AD \rightarrow B$ ,  $AD \rightarrow C$ ,  $BC \rightarrow A$ ,  $BC \rightarrow D$ ,  $BD \rightarrow A$ ,  $BD \rightarrow C$ ,  $CD \rightarrow A$ ,  $CD \rightarrow B$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ACD \rightarrow B$  and  $BCD \rightarrow A$ .

### Exercise 3.2.2b

- i) From the analysis of closures in 3.2.2a(i), we find that the only key is A. All other sets either do not have ABCD as the closure or contain A.
- ii) From the analysis of closures 3.2.2a(ii), we find that AB, AD, BC, and CD are keys. All other sets either do not have ABCD as the closure or contain one of these four sets.
- iii) From the analysis of closures 3.2.2a(iii), we find that A, B, C and D are keys. All other sets either do not have ABCD as the closure or contain one of these four sets.

### Exercise 3.2.2c

- i) The superkeys are all those sets that contain one of the keys in 3.2.2b(i). The superkeys are AB, AC, AD, ABC, ABD, ACD, BCD and ABCD.
- ii) The superkeys are all those sets that contain one of the keys in 3.2.2b(ii). The superkeys are ABC, ABD, ACD, BCD and ABCD.
- iii) The superkeys are all those sets that contain one of the keys in 3.2.2b(iii). The superkeys are AB, AC, AD, BC, BD, CD, ABC, ABD, ACD, BCD and ABCD.

### Exercise 3.2.3a

Since  $A_1A_2\dots A_nC$  contains  $A_1A_2\dots A_n$ , then the closure of  $A_1A_2\dots A_nC$  contains B. Thus it follows that  $A_1A_2\dots A_nC \rightarrow B$ .

### Exercise 3.2.3b

From 3.2.3a, we know that  $A_1A_2\dots A_nC \rightarrow B$ . Using the concept of trivial dependencies, we can show that  $A_1A_2\dots A_nC \rightarrow C$ . Thus  $A_1A_2\dots A_nC \rightarrow BC$ .

### Exercise 3.2.3c

From  $A_1A_2\dots A_nE_1E_2\dots E_j$ , we know that the closure contains  $B_1B_2\dots B_m$  because of the FD  $A_1A_2\dots A_n \rightarrow B_1B_2\dots B_m$ . The  $B_1B_2\dots B_m$  and the  $E_1E_2\dots E_j$  combine to form the  $C_1C_2\dots C_k$ . Thus the closure of  $A_1A_2\dots A_nE_1E_2\dots E_j$  contains D as well. Thus,  $A_1A_2\dots A_nE_1E_2\dots E_j \rightarrow D$ .

### Exercise 3.2.3d

From  $A_1A_2\dots A_nC_1C_2\dots C_k$ , we know that the closure contains  $B_1B_2\dots B_m$  because of the FD  $A_1A_2\dots A_n \rightarrow B_1B_2\dots B_m$ . The  $C_1C_2\dots C_k$  also tell us that the closure of  $A_1A_2\dots A_nC_1C_2\dots C_k$  contains  $D_1D_2\dots D_j$ . Thus,  $A_1A_2\dots A_nC_1C_2\dots C_k \rightarrow B_1B_2\dots B_mD_1D_2\dots D_j$ .

### Exercise 3.2.4a

If attribute A represented Social Security Number and B represented a person's name, then we would assume  $A \rightarrow B$  but  $B \rightarrow A$  would not be valid because there may be many people with the same name and different Social Security Numbers.

### Exercise 3.2.4b

Let attribute A represent Social Security Number, B represent gender and C represent name. Surely Social Security Number and gender can uniquely identify a person's name (i.e.  $AB \rightarrow C$ ). A Social Security Number can also uniquely identify a person's name (i.e.  $A \rightarrow C$ ). However, gender does not uniquely determine a name (i.e.  $B \rightarrow C$  is not valid).

### Exercise 3.2.4c

Let attribute A represent latitude and B represent longitude. Together, both attributes can uniquely determine C, a point on the world map (i.e.  $AB \rightarrow C$ ). However, neither A nor B can uniquely identify a point (i.e.  $A \rightarrow C$  and  $B \rightarrow C$  are not valid).

### Exercise 3.2.5

Given a relation with attributes  $A_1A_2 \dots A_n$ , we are told that there are no functional dependencies of the form  $B_1B_2 \dots B_{n-1} \rightarrow C$  where  $B_1B_2 \dots B_{n-1}$  is n-1 of the attributes from  $A_1A_2 \dots A_n$  and C is the remaining attribute from  $A_1A_2 \dots A_n$ . In this case, the set  $B_1B_2 \dots B_{n-1}$  and any subset do not functionally determine C. Thus the only functional dependencies that we can make are ones where C is on both the left and right hand sides. All of these functional dependencies would be trivial and thus the relation has no nontrivial FD's.

### Exercise 3.2.6

Let's prove this by using the contrapositive. We wish to show that if  $X^+$  is not a subset of  $Y^+$ , then it must be that X is not a subset of Y.

If  $X^+$  is not a subset of  $Y^+$ , there must be attributes  $A_1A_2 \dots A_n$  in  $X^+$  that are not in  $Y^+$ . If any of these attributes were originally in X, then we are done because Y does not contain any of the  $A_1A_2 \dots A_n$ . However, if the  $A_1A_2 \dots A_n$  were added by the closure, then we must examine the case further. Assume that there was some FD  $C_1C_2 \dots C_m \rightarrow A_1A_2 \dots A_j$  where  $A_1A_2 \dots A_j$  is some subset of  $A_1A_2 \dots A_n$ . It must be then that  $C_1C_2 \dots C_m$  or some subset of  $C_1C_2 \dots C_m$  is in X. However, the attributes  $C_1C_2 \dots C_m$  cannot be in Y because we assumed that attributes  $A_1A_2 \dots A_n$  are only in  $X^+$  and are not in  $Y^+$ . Thus, X is not a subset of Y.

By proving the contrapositive, we have also proved if  $X \subseteq Y$ , then  $X^+ \subseteq Y^+$ .

### Exercise 3.2.7

The algorithm to find  $X^+$  is outlined on pg. 76. Using that algorithm, we can prove that  $(X^+)^+ = X^+$ . We will do this by using a proof by contradiction.

Suppose that  $(X^+)^+ \neq X^+$ . Then for  $(X^+)^+$ , it must be that some FD allowed additional attributes to be added to the original set  $X^+$ . For example,  $X^+ \rightarrow A$  where  $A$  is some attribute not in  $X^+$ . However, if this were the case, then  $X^+$  would not be the closure of  $X$ . The closure of  $X$  would have to include  $A$  as well. This contradicts the fact that we were given the closure of  $X$ ,  $X^+$ . Therefore, it must be that  $(X^+)^+ = X^+$  or else  $X^+$  is not the closure of  $X$ .

### Exercise 3.2.8a

If all sets of attributes are closed, then there cannot be any nontrivial functional dependencies. Suppose  $A_1A_2...A_n \rightarrow B$  is a nontrivial dependency. Then  $\{A_1A_2...A_n\}^+$  contains  $B$  and thus  $A_1A_2...A_n$  is not closed.

### Exercise 3.2.8b

If the only closed sets are  $\emptyset$  and  $\{A,B,C,D\}$ , then the following FDs hold:

$A \rightarrow B$	$A \rightarrow C$	$A \rightarrow D$
$B \rightarrow A$	$B \rightarrow C$	$B \rightarrow D$
$C \rightarrow A$	$C \rightarrow B$	$C \rightarrow D$
$D \rightarrow A$	$D \rightarrow B$	$D \rightarrow C$
$AB \rightarrow C$	$AB \rightarrow D$	
$AC \rightarrow B$	$AC \rightarrow D$	
$AD \rightarrow B$	$AD \rightarrow C$	
$BC \rightarrow A$	$BC \rightarrow D$	
$BD \rightarrow A$	$BD \rightarrow C$	
$CD \rightarrow A$	$CD \rightarrow B$	
$ABC \rightarrow D$		
$ABD \rightarrow C$		
$ACD \rightarrow B$		
$BCD \rightarrow A$		

### Exercise 3.2.8c

If the only closed sets are  $\emptyset$ ,  $\{A,B\}$  and  $\{A,B,C,D\}$ , then the following FDs hold:

$A \rightarrow B$		
$B \rightarrow A$		
$C \rightarrow A$	$C \rightarrow B$	$C \rightarrow D$
$D \rightarrow A$	$D \rightarrow B$	$D \rightarrow C$
$AC \rightarrow B$	$AC \rightarrow D$	
$AD \rightarrow B$	$AD \rightarrow C$	
$BC \rightarrow A$	$BC \rightarrow D$	
$BD \rightarrow A$	$BD \rightarrow C$	
$CD \rightarrow A$	$CD \rightarrow B$	
$ABC \rightarrow D$		
$ABD \rightarrow C$		

$ACD \rightarrow B$   
 $BCD \rightarrow A$

### Exercise 3.2.9

We can think of this problem as a situation where the attributes A,B,C represent cities and the functional dependencies represent one way paths between the cities. The minimal bases are the minimal number of pathways that are needed to connect the cities. We do not want to create another roadway if the two cities are already connected.

The systematic way to do this would be to check all possible sets of the pathways. However, we can simplify the situation by noting that it takes more than two pathways to visit the two other cities and come back. Also, if we find a set of pathways that is minimal, adding additional pathways will not create another minimal set.

The two sets of minimal bases that were given in example 3.11 are:

$\{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$   
 $\{A \rightarrow B, B \rightarrow A, B \rightarrow C, C \rightarrow B\}$

The additional sets of minimal bases are:

$\{C \rightarrow B, B \rightarrow A, A \rightarrow C\}$   
 $\{A \rightarrow B, A \rightarrow C, B \rightarrow A, C \rightarrow A\}$   
 $\{A \rightarrow C, B \rightarrow C, C \rightarrow A, C \rightarrow B\}$

### Exercise 3.2.10a

We need to compute the closures of all subsets of  $\{ABC\}$ , although there is no need to think about the empty set or the set of all three attributes. Here are the calculations for the remaining six sets:

$\{A\}^+ = A$   
 $\{B\}^+ = B$   
 $\{C\}^+ = ACE$   
 $\{AB\}^+ = ABCDE$   
 $\{AC\}^+ = ACE$   
 $\{BC\}^+ = ABCDE$

We ignore D and E, so a basis for the resulting functional dependencies for ABC is:  $C \rightarrow A$  and  $AB \rightarrow C$ . Note that  $BC \rightarrow A$  is true, but follows logically from  $C \rightarrow A$ , and therefore may be omitted from our list.

### Exercise 3.2.10b

We need to compute the closures of all subsets of  $\{ABC\}$ , although there is no need to think about the empty set or the set of all three attributes. Here are the calculations for the remaining six sets:

$\{A\}^+ = AD$   
 $\{B\}^+ = B$   
 $\{C\}^+ = C$   
 $\{AB\}^+ = ABDE$   
 $\{AC\}^+ = ABCDE$   
 $\{BC\}^+ = BC$

We ignore D and E, so a basis for the resulting functional dependencies for ABC is:  $AC \rightarrow B$ .

### Exercise 3.2.10c

We need to compute the closures of all subsets of  $\{ABC\}$ , although there is no need to think about the empty set or the set of all three attributes. Here are the calculations for the remaining six sets:

$\{A\}^+ = A$   
 $\{B\}^+ = B$   
 $\{C\}^+ = C$   
 $\{AB\}^+ = ABD$   
 $\{AC\}^+ = ABCDE$   
 $\{BC\}^+ = ABCDE$

We ignore D and E, so a basis for the resulting functional dependencies for ABC is:  $AC \rightarrow B$  and  $BC \rightarrow A$ .

### Exercise 3.2.10d

We need to compute the closures of all subsets of  $\{ABC\}$ , although there is no need to think about the empty set or the set of all three attributes. Here are the calculations for the remaining six sets:

$\{A\}^+ = ABCDE$   
 $\{B\}^+ = ABCDE$   
 $\{C\}^+ = ABCDE$   
 $\{AB\}^+ = ABCDE$   
 $\{AC\}^+ = ABCDE$   
 $\{BC\}^+ = ABCDE$

We ignore D and E, so a basis for the resulting functional dependencies for ABC is:  $A \rightarrow B$ ,  $B \rightarrow C$  and  $C \rightarrow A$ .

### Exercise 3.2.11

For step one of Algorithm 3.7, suppose we have the FD  $ABC \rightarrow DE$ . We want to use Armstrong's Axioms to show that  $ABC \rightarrow D$  and  $ABC \rightarrow E$  follow. Surely the functional dependencies  $DE \rightarrow D$



and  $DE \rightarrow E$  hold because they are trivial and follow the reflexivity property. Using the transitivity rule, we can derive the FD  $ABC \rightarrow D$  from the FDs  $ABC \rightarrow DE$  and  $DE \rightarrow D$ . Likewise, we can do the same for  $ABC \rightarrow DE$  and  $DE \rightarrow E$  and derive the FD  $ABC \rightarrow E$ .

For steps two through four of Algorithm 3.7, suppose we have the initial set of attributes of the closure as ABC. Suppose also that we have FDs  $C \rightarrow D$  and  $D \rightarrow E$ . According to Algorithm 3.7, the closure should become ABCDE. Taking the FD  $C \rightarrow D$  and augmenting both sides with attributes AB we get the FD  $ABC \rightarrow ABD$ . We can use the splitting method in step one to get the FD  $ABC \rightarrow D$ . Since D is not in the closure, we can add attribute D. Taking the FD  $D \rightarrow E$  and augmenting both sides with attributes ABC we get the FD  $ABCD \rightarrow ABCDE$ . Using again the splitting method in step one we get the FD  $ABCD \rightarrow E$ . Since E is not in the closure, we can add attribute E.

Given a set of FDs, we can prove that a FD F follows by taking the closure of the left side of FD F. The steps to compute the closure in Algorithm 3.7 can be mimicked by Armstrong's axioms and thus we can prove F from the given set of FDs using Armstrong's axioms.

### Exercise 3.3.1a

In the solution to Exercise 3.2.1 we found that there are 14 nontrivial dependencies, including the three given ones and eleven derived dependencies. They are:  $C \rightarrow A$ ,  $C \rightarrow D$ ,  $D \rightarrow A$ ,  $AB \rightarrow D$ ,  $AB \rightarrow C$ ,  $AC \rightarrow D$ ,  $BC \rightarrow A$ ,  $BC \rightarrow D$ ,  $BD \rightarrow A$ ,  $BD \rightarrow C$ ,  $CD \rightarrow A$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ , and  $BCD \rightarrow A$ .

We also learned that the three keys were AB, BC, and BD. Thus, any dependency above that does not have one of these pairs on the left is a BCNF violation. These are:  $C \rightarrow A$ ,  $C \rightarrow D$ ,  $D \rightarrow A$ ,  $AC \rightarrow D$ , and  $CD \rightarrow A$ .

One choice is to decompose using the violation  $C \rightarrow D$ . Using the above FDs, we get ACD and BC as decomposed relations. BC is surely in BCNF, since any two-attribute relation is. Using Algorithm 3.12 to discover the projection of FDs on relation ACD, we discover that ACD is not in BCNF since C is its only key. However,  $D \rightarrow A$  is a dependency that holds in ABCD and therefore holds in ACD. We must further decompose ACD into AD and CD. Thus, the three relations of the decomposition are BC, AD, and CD.

### Exercise 3.3.1b

By computing the closures of all 15 nonempty subsets of ABCD, we can find all the nontrivial FDs. They are  $B \rightarrow C$ ,  $B \rightarrow D$ ,  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $BC \rightarrow D$ ,  $BD \rightarrow C$ ,  $ABC \rightarrow D$  and  $ABD \rightarrow C$ . From the closures we can also deduce that the only key is AB. Thus, any dependency above that does not contain AB on the left is a BCNF violation. These are:  $B \rightarrow C$ ,  $B \rightarrow D$ ,  $BC \rightarrow D$  and  $BD \rightarrow C$ .

One choice is to decompose using the violation  $B \rightarrow C$ . Using the above FDs, we get BCD and AB as decomposed relations. AB is surely in BCNF, since any two-attribute relation is. Using Algorithm 3.12 to discover the projection of FDs on relation BCD, we discover that BCD is in

BCNF since B is its only key and the projected FDs all have B on the left side. Thus the two relations of the decomposition are AB and BCD.

### Exercise 3.3.1c

In the solution to Exercise 3.2.2(ii), we found that there are 12 nontrivial dependencies, including the four given ones and the eight derived ones. They are  $AB \rightarrow C$ ,  $BC \rightarrow D$ ,  $CD \rightarrow A$ ,  $AD \rightarrow B$ ,  $AB \rightarrow D$ ,  $AD \rightarrow C$ ,  $BC \rightarrow A$ ,  $CD \rightarrow B$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ACD \rightarrow B$  and  $BCD \rightarrow A$ .

We also found out that the keys are AB, AD, BC, and CD. Thus, any dependency above that does not have one of these pairs on the left is a BCNF violation. However, all of the FDs contain a key on the left so there are no BCNF violations.

No decomposition is necessary since all the FDs do not violate BCNF.

### Exercise 3.3.1d

In the solution to Exercise 3.2.2(iii), we found that there are 28 nontrivial dependencies, including the four given ones and the 24 derived ones. They are  $A \rightarrow B$ ,  $B \rightarrow C$ ,  $C \rightarrow D$ ,  $D \rightarrow A$ ,  $A \rightarrow C$ ,  $A \rightarrow D$ ,  $B \rightarrow D$ ,  $B \rightarrow A$ ,  $C \rightarrow A$ ,  $C \rightarrow B$ ,  $D \rightarrow B$ ,  $D \rightarrow C$ ,  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $AC \rightarrow B$ ,  $AC \rightarrow D$ ,  $AD \rightarrow B$ ,  $AD \rightarrow C$ ,  $BC \rightarrow A$ ,  $BC \rightarrow D$ ,  $BD \rightarrow A$ ,  $BD \rightarrow C$ ,  $CD \rightarrow A$ ,  $CD \rightarrow B$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ACD \rightarrow B$  and  $BCD \rightarrow A$ .

We also found out that the keys are A,B,C,D. Thus, any dependency above that does not have one of these attributes on the left is a BCNF violation. However, all of the FDs contain a key on the left so there are no BCNF violations.

No decomposition is necessary since all the FDs do not violate BCNF.

### Exercise 3.3.1e

By computing the closures of all 31 nonempty subsets of ABCDE, we can find all the nontrivial FDs. They are  $AB \rightarrow C$ ,  $DE \rightarrow C$ ,  $B \rightarrow D$ ,  $AB \rightarrow D$ ,  $BC \rightarrow D$ ,  $BE \rightarrow C$ ,  $BE \rightarrow D$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ABE \rightarrow C$ ,  $ABE \rightarrow D$ ,  $ADE \rightarrow C$ ,  $BCE \rightarrow D$ ,  $BDE \rightarrow C$ ,  $ABCE \rightarrow D$ , and  $ABDE \rightarrow C$ . From the closures we can also deduce that the only key is ABE. Thus, any dependency above that does not contain ABE on the left is a BCNF violation. These are:  $AB \rightarrow C$ ,  $DE \rightarrow C$ ,  $B \rightarrow D$ ,  $AB \rightarrow D$ ,  $BC \rightarrow D$ ,  $BE \rightarrow C$ ,  $BE \rightarrow D$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ADE \rightarrow C$ ,  $BCE \rightarrow D$  and  $BDE \rightarrow C$ .

One choice is to decompose using the violation  $AB \rightarrow C$ . Using the above FDs, we get ABCD and ABE as decomposed relations. Using Algorithm 3.12 to discover the projection of FDs on relation ABCD, we discover that ABCD is not in BCNF since AB is its only key and the FD  $B \rightarrow D$  follows for ABCD. Using violation  $B \rightarrow D$  to further decompose, we get BD and ABC as decomposed relations. BD is in BCNF because it is a two-attribute relation. Using Algorithm 3.12 again, we discover that ABC is in BCNF since AB is the only key and  $AB \rightarrow C$  is the only nontrivial FD. Going back to relation ABE, following Algorithm 3.12 tells us that ABE is in

BCNF because there are no keys and no nontrivial FDs. Thus the three relations of the decomposition are ABC, BD and ABE.

### Exercise 3.3.1f

By computing the closures of all 31 nonempty subsets of ABCDE, we can find all the nontrivial FDs. They are:  $C \rightarrow B$ ,  $C \rightarrow D$ ,  $C \rightarrow E$ ,  $D \rightarrow B$ ,  $D \rightarrow E$ ,  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $AB \rightarrow E$ ,  $AC \rightarrow B$ ,  $AC \rightarrow D$ ,  $AC \rightarrow E$ ,  $AD \rightarrow B$ ,  $AD \rightarrow C$ ,  $AD \rightarrow E$ ,  $BC \rightarrow D$ ,  $BC \rightarrow E$ ,  $BD \rightarrow E$ ,  $CD \rightarrow B$ ,  $CD \rightarrow E$ ,  $CE \rightarrow B$ ,  $CE \rightarrow D$ ,  $DE \rightarrow B$ ,  $ABC \rightarrow D$ ,  $ABC \rightarrow E$ ,  $ABD \rightarrow C$ ,  $ABD \rightarrow E$ ,  $ABE \rightarrow C$ ,  $ABE \rightarrow D$ ,  $ACD \rightarrow B$ ,  $ACD \rightarrow E$ ,  $ACE \rightarrow B$ ,  $ACE \rightarrow D$ ,  $ADE \rightarrow B$ ,  $ADE \rightarrow C$ ,  $BCD \rightarrow E$ ,  $BCE \rightarrow D$ ,  $CDE \rightarrow B$ ,  $ABCD \rightarrow E$ ,  $ABCE \rightarrow D$ ,  $ABDE \rightarrow C$  and  $ACDE \rightarrow B$ . From the closures we can also deduce that the keys are AB, AC and AD. Thus, any dependency above that does not contain one of the above pairs on the left is a BCNF violation. These are:  $C \rightarrow B$ ,  $C \rightarrow D$ ,  $C \rightarrow E$ ,  $D \rightarrow B$ ,  $D \rightarrow E$ ,  $BC \rightarrow D$ ,  $BC \rightarrow E$ ,  $BD \rightarrow E$ ,  $CD \rightarrow B$ ,  $CD \rightarrow E$ ,  $CE \rightarrow B$ ,  $CE \rightarrow D$ ,  $DE \rightarrow B$ ,  $BCD \rightarrow E$ ,  $BCE \rightarrow D$  and  $CDE \rightarrow B$ .

One choice is to decompose using the violation  $D \rightarrow B$ . Using the above FDs, we get BDE and ABC as decomposed relations. Using Algorithm 3.12 to discover the projection of FDs on relation BDE, we discover that BDE is in BCNF since D, BD, DE are the only keys and all the projected FDs contain D, BD, or DE in the left side. Going back to relation ABC, following Algorithm 3.12 tells us that ABC is not in BCNF because since AB and AC are its only keys and the FD  $C \rightarrow B$  follows for ABC. Using violation  $C \rightarrow B$  to further decompose, we get BC and AC as decomposed relations. Both BC and AC are in BCNF because they are two-attribute relations. Thus the three relations of the decomposition are BDE, BC and AC.

### Exercise 3.3.2

Yes, we will get the same result. Both  $A \rightarrow B$  and  $A \rightarrow BC$  have A on the left side and part of the process of decomposition involves finding  $\{A\}^+$  to form one decomposed relation and A plus the rest of the attributes not in  $\{A\}^+$  as the second relation. Both cases yield the same decomposed relations.

### Exercise 3.3.3

Yes, we will still get the same result. Both  $A \rightarrow B$  and  $A \rightarrow BC$  have A on the left side and part of the process of decomposition involves finding  $\{A\}^+$  to form one decomposed relation and A plus the rest of the attributes not in  $\{A\}^+$  as the second relation. Both cases yield the same decomposed relations.

### Exercise 3.3.4

This is taken from Example 3.21 pg. 95.  
Suppose that an instance of relation R only contains two tuples.

A	B	C
1	2	3
4	2	5

The projections of R onto the relations with schemas  $\{A,B\}$  and  $\{B,C\}$  are:

A	B
1	2
4	2

B	C
2	3
2	5

If we do a natural join on the two projections, we will get:

A	B	C
1	2	3
1	2	5
4	2	3
4	2	5

The result of the natural join is not equal to the original relation R.

### Exercise 3.4.1a

This is the initial tableau:

A	B	C	D	E
a	b	c	d <sub>1</sub>	e <sub>1</sub>
a <sub>1</sub>	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c	d <sub>1</sub>	e

This is the final tableau after applying FDs  $B \rightarrow E$  and  $CE \rightarrow A$ .

A	B	C	D	E
a	b	c	d <sub>1</sub>	e <sub>1</sub>
a	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c	d <sub>1</sub>	e

Since there is not an unsubscripted row, the decomposition for R is not lossless for this set of FDs.

We can use the final tableau as an instance of R as an example for why the join is not lossless. The projected relations are:

A	B	C
a	b	c

a	b <sub>1</sub>	c
---	----------------	---

<b>B</b>	<b>C</b>	<b>D</b>
b	c	d <sub>1</sub>
b	c	d
b <sub>1</sub>	c	d <sub>1</sub>

<b>A</b>	<b>C</b>	<b>E</b>
a	c	e <sub>1</sub>
a	c	e

The joined relation is:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b	c	d <sub>1</sub>	e <sub>1</sub>
a	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c	d <sub>1</sub>	e <sub>1</sub>
a	b	c	d <sub>1</sub>	e
a	b	c	d	e
a	b <sub>1</sub>	c	d <sub>1</sub>	e

The joined relation has three more tuples than the original tableau.

### Exercise 3.4.1b

This is the initial tableau:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b	c	d <sub>1</sub>	e <sub>1</sub>
a <sub>1</sub>	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c	d <sub>1</sub>	e

This is the final tableau after applying FDs  $AC \rightarrow E$  and  $BC \rightarrow D$

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b	c	d	e
a <sub>1</sub>	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c	d <sub>1</sub>	e

Since there is an unsubscripted row, the decomposition for R is lossless for this set of FDs.

#### Exercise 3.4.1c

This is the initial tableau:

A	B	C	D	E
a	b	c	d <sub>1</sub>	e <sub>1</sub>
a <sub>1</sub>	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c	d <sub>1</sub>	e

This is the final tableau after applying FDs  $A \rightarrow D$ ,  $D \rightarrow E$  and  $B \rightarrow D$ .

A	B	C	D	E
a	b	c	d	e
a <sub>1</sub>	b	c	d	e
a	b <sub>1</sub>	c	d	e

Since there is an unsubscripted row, the decomposition for R is lossless for this set of FDs.

#### Exercise 3.4.1d

This is the initial tableau:

A	B	C	D	E
a	b	c	d <sub>1</sub>	e <sub>1</sub>
a <sub>1</sub>	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c	d <sub>1</sub>	e

This is the final tableau after applying FDs  $A \rightarrow D$ ,  $CD \rightarrow E$  and  $E \rightarrow D$

A	B	C	D	E
a	b	c	d	e
a <sub>1</sub>	b	c	d	e
a	b <sub>1</sub>	c	d	e

Since there is an unsubscripted row, the decomposition for R is lossless for this set of FDs.

#### Exercise 3.4.2

When we decompose a relation into BCNF, we will project the FDs onto the decomposed relations to get new sets of FDs. These dependencies are preserved if the union of these new sets is equivalent to the original set of FDs.

For the FDs of 3.4.1a, the dependencies are not preserved. The union of the new sets of FDs is  $CE \rightarrow A$ . However, the FD  $B \rightarrow E$  is not in the union and cannot be derived. Thus the two sets of FDs are not equivalent.

For the FDs of 3.4.1b, the dependencies are preserved. The union of the new sets of FDs is  $AC \rightarrow E$  and  $BC \rightarrow D$ . This is precisely the same as the original set of FDs and thus the two sets of FDs are equivalent.

For the FDs of 3.4.1c, the dependencies are not preserved. The union of the new sets of FDs is  $B \rightarrow D$  and  $A \rightarrow E$ . The FDs  $A \rightarrow D$  and  $D \rightarrow E$  are not in the union and cannot be derived. Thus the two sets of FDs are not equivalent.

For the FDs of 3.4.1d, the dependencies are not preserved. The union of the new sets of FDs is  $AC \rightarrow E$ . However, the FDs  $A \rightarrow D$ ,  $CD \rightarrow E$  and  $E \rightarrow D$  are not in the union and cannot be derived. Thus the two sets of FDs are not equivalent.

### **Exercise 3.5.1a**

In the solution to Exercise 3.3.1a we found that there are 14 nontrivial dependencies. They are:  $C \rightarrow A$ ,  $C \rightarrow D$ ,  $D \rightarrow A$ ,  $AB \rightarrow D$ ,  $AB \rightarrow C$ ,  $AC \rightarrow D$ ,  $BC \rightarrow A$ ,  $BC \rightarrow D$ ,  $BD \rightarrow A$ ,  $BD \rightarrow C$ ,  $CD \rightarrow A$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ , and  $BCD \rightarrow A$ .

We also learned that the three keys were AB, BC, and BD. Since all the attributes on the right sides of the FDs are prime, there are no 3NF violations.

Since there are no 3NF violations, it is not necessary to decompose the relation.

### **Exercise 3.5.1b**

In the solution to Exercise 3.3.1b we found that there are 8 nontrivial dependencies. They are  $B \rightarrow C$ ,  $B \rightarrow D$ ,  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $BC \rightarrow D$ ,  $BD \rightarrow C$ ,  $ABC \rightarrow D$  and  $ABD \rightarrow C$ .

We also found out that the only key is AB. FDs where the left side is not a superkey or the attributes on the right are not part of some key are 3NF violations. The 3NF violations are  $B \rightarrow C$ ,  $B \rightarrow D$ ,  $BC \rightarrow D$  and  $BD \rightarrow C$ .

Using algorithm 3.26, we can decompose into relations using the minimal basis  $B \rightarrow C$  and  $B \rightarrow D$ . The resulting decomposed relations would be BC and BD. However, none of these two sets of attributes is a superkey. Thus we add relation AB to the result. The final set of decomposed relations is BC, BD and AB.

### **Exercise 3.5.1c**

In the solution to Exercise 3.3.1c we found that there are 12 nontrivial dependencies. They are  $AB \rightarrow C$ ,  $BC \rightarrow D$ ,  $CD \rightarrow A$ ,  $AD \rightarrow B$ ,  $AB \rightarrow D$ ,  $AD \rightarrow C$ ,  $BC \rightarrow A$ ,  $CD \rightarrow B$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ACD \rightarrow B$  and  $BCD \rightarrow A$ .

We also found out that the keys are AB, AD, BC, and CD. Since all the attributes on the right sides of the FDs are prime, there are no 3NF violations.

Since there are no 3NF violations, it is not necessary to decompose the relation.

#### **Exercise 3.5.1d**

In the solution to Exercise 3.3.1d we found that there are 28 nontrivial dependencies. They are  $A \rightarrow B$ ,  $B \rightarrow C$ ,  $C \rightarrow D$ ,  $D \rightarrow A$ ,  $A \rightarrow C$ ,  $A \rightarrow D$ ,  $B \rightarrow D$ ,  $B \rightarrow A$ ,  $C \rightarrow A$ ,  $C \rightarrow B$ ,  $D \rightarrow B$ ,  $D \rightarrow C$ ,  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $AC \rightarrow B$ ,  $AC \rightarrow D$ ,  $AD \rightarrow B$ ,  $AD \rightarrow C$ ,  $BC \rightarrow A$ ,  $BC \rightarrow D$ ,  $BD \rightarrow A$ ,  $BD \rightarrow C$ ,  $CD \rightarrow A$ ,  $CD \rightarrow B$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ACD \rightarrow B$  and  $BCD \rightarrow A$ .

We also found out that the keys are A,B,C,D. Since all the attributes on the right sides of the FDs are prime, there are no 3NF violations.

Since there are no 3NF violations, it is not necessary to decompose the relation.

#### **Exercise 3.5.1e**

In the solution to Exercise 3.3.1e we found that there are 16 nontrivial dependencies. They are  $AB \rightarrow C$ ,  $DE \rightarrow C$ ,  $B \rightarrow D$ ,  $AB \rightarrow D$ ,  $BC \rightarrow D$ ,  $BE \rightarrow C$ ,  $BE \rightarrow D$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ABE \rightarrow C$ ,  $ABE \rightarrow D$ ,  $ADE \rightarrow C$ ,  $BCE \rightarrow D$ ,  $BDE \rightarrow C$ ,  $ABCE \rightarrow D$ , and  $ABDE \rightarrow C$ .

We also found out that the only key is ABE. FDs where the left side is not a superkey or the attributes on the right are not part of some key are 3NF violations. The 3NF violations are  $AB \rightarrow C$ ,  $DE \rightarrow C$ ,  $B \rightarrow D$ ,  $AB \rightarrow D$ ,  $BC \rightarrow D$ ,  $BE \rightarrow C$ ,  $BE \rightarrow D$ ,  $ABC \rightarrow D$ ,  $ABD \rightarrow C$ ,  $ADE \rightarrow C$ ,  $BCE \rightarrow D$  and  $BDE \rightarrow C$ .

Using algorithm 3.26, we can decompose into relations using the minimal basis  $AB \rightarrow C$ ,  $DE \rightarrow C$  and  $B \rightarrow D$ . The resulting decomposed relations would be ABC, CDE and BD. However, none of these three sets of attributes is a superkey. Thus we add relation ABE to the result. The final set of decomposed relations is ABC, CDE, BD and ABE.

#### **Exercise 3.5.1f**

In the solution to Exercise 3.3.1f we found that there are 41 nontrivial dependencies. They are:  $C \rightarrow B$ ,  $C \rightarrow D$ ,  $C \rightarrow E$ ,  $D \rightarrow B$ ,  $D \rightarrow E$ ,  $AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $AB \rightarrow E$ ,  $AC \rightarrow B$ ,  $AC \rightarrow D$ ,  $AC \rightarrow E$ ,  $AD \rightarrow B$ ,  $AD \rightarrow C$ ,  $AD \rightarrow E$ ,  $BC \rightarrow D$ ,  $BC \rightarrow E$ ,  $BD \rightarrow E$ ,  $CD \rightarrow B$ ,  $CD \rightarrow E$ ,  $CE \rightarrow B$ ,  $CE \rightarrow D$ ,  $DE \rightarrow B$ ,  $ABC \rightarrow D$ ,  $ABC \rightarrow E$ ,  $ABD \rightarrow C$ ,  $ABD \rightarrow E$ ,  $ABE \rightarrow C$ ,  $ABE \rightarrow D$ ,  $ACD \rightarrow B$ ,  $ACD \rightarrow E$ ,  $ACE \rightarrow B$ ,  $ACE \rightarrow D$ ,  $ADE \rightarrow B$ ,  $ADE \rightarrow C$ ,  $BCD \rightarrow E$ ,  $BCE \rightarrow D$ ,  $CDE \rightarrow B$ ,  $ABCD \rightarrow E$ ,  $ABCE \rightarrow D$ ,  $ABDE \rightarrow C$  and  $ACDE \rightarrow B$ .



We also found out that the keys are AB, AC and AD. FDs where the left side is not a superkey or the attributes on the right are not part of some key are 3NF violations. The 3NF violations are  $C \rightarrow E$ ,  $D \rightarrow E$ ,  $BC \rightarrow E$ ,  $BD \rightarrow E$ ,  $CD \rightarrow E$  and  $BCD \rightarrow E$ .

Using algorithm 3.26, we can decompose into relations using the minimal basis  $AB \rightarrow C$ ,  $C \rightarrow D$ ,  $D \rightarrow B$  and  $D \rightarrow E$ . The resulting decomposed relations would be ABC, CD, BD and DE. Since relation ABC contains a key, we can stop with the decomposition. The final set of decomposed relations is ABC, CD, BD and DE.

### **Exercise 3.5.2a**

The usual procedure to find the keys would be to take the closure of all 63 nonempty subsets. However, if we notice that none of the right sides of the FDs contains attributes H and S. Thus we know that attributes H and S must be part of any key. We eventually will find out that HS is the only key for the Courses relation.

### **Exercise 3.5.2b**

The first step to verify that the given FDs are their own minimal basis is to check to see if any of the FDs can be removed. However, if we remove any one of the five FDs, the remaining four FDs do not imply the removed FD.

The second step to verify that the given FDs are their own minimal basis is to check to see if any of the left sides of an FD can have one or more attributes removed without losing the dependencies. However, this is not the case for the four FDs that contain two attributes on the left side.

Thus, the given set of FDs has been verified to be the minimal basis.

### **Exercise 3.5.2c**

Since the only key is HS, the given set of FDs has some dependencies that violate 3NF. We also know that the given set of FDs is a minimal basis. Thus the decomposed relations are CT, HRC, HTR, HSR and CSG. Since the relation HSR contains a key, we do not need to add an additional relation. The final set of decomposed relations is CT, HRC, HTR, HSR and CSG.

None of the decomposed relations violate BCNF. This can be verified by projecting the given set of FDs onto each of the decomposed relations. All of the projections of FDs have superkeys on their left sides.

### **Exercise 3.5.3**

The usual procedure to find the keys would be to take the closure of all 63 nonempty subsets. However, if we notice that none of the right sides of the FDs contains attributes I and S. Thus we

know that attributes I and S must be part of any key. We eventually will find out that IS is the only key for the Stocks relation.

The first step to verify that the given FDs are their own minimal basis is to check to see if any of the FDs can be removed. However, if we remove any one of the four FDs, the remaining three FDs do not imply the removed FD.

The second step to verify that the given FDs are their own minimal basis is to check to see if any of the left sides of an FD can have one or more attributes removed without losing the dependencies. However, this is not the case for the one FD that contains two attributes on the left side.

Thus, the given set of FDs has been verified to be the minimal basis.

Since the only key is IS, the given set of FDs has some dependencies that violate 3NF. We also know that the given set of FDs is a minimal basis. Thus the decomposed relations are SD, IB, ISQ and BO. Since the relation ISQ contains a key, we do not need to add an additional relation. The final set of decomposed relations is SD, IB, ISQ and BO.

#### Exercise 3.5.4

This is the initial tableau:

A	B	C	D	E
a	b	c	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d	e <sub>2</sub>
a	b	c <sub>3</sub>	d <sub>3</sub>	e

This is the final tableau after applying FDs  $AB \rightarrow C$ ,  $C \rightarrow B$  and  $A \rightarrow D$ .

A	B	C	D	E
a	b	c	d	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d	e <sub>2</sub>
a	b	c	d	e

Since there is an unsubscripted row, the decomposition for R is lossless for this set of FDs.

#### Exercise 3.5.5

Suppose that our relation relates to the work environment and has three attributes, Name, RoomNumber and ComputerID. Suppose also that only the following FDs hold:

Name  $\rightarrow$  RoomNumber  
ComputerID  $\rightarrow$  RoomNumber

The first FD says that a person can only be in one office at a time. The second FD says that each computer can only be associated with one office at a time (i.e. the computer is fixed to each room). None of the left sides of the FDs are keys and the respective right side attributes do not belong to any key. Using algorithm 3.20 to decompose, we take the violating FD  $\text{Name} \rightarrow \text{RoomNumber}$  and use it to decompose the initial relation into two smaller relations:

R1(Name,RoomNumber)  
R2(Name,ComputerID)

We do not need to further decompose the two relations so we are done. However, we have lost the FD  $\text{ComputerID} \rightarrow \text{RoomNumber}$  with this decomposition. Suppose that ‘John Doe’ works in room number ‘1’. Suppose also that computer ID ‘A’ is located in room ‘1’. The original relation would contain the tuple:

{‘John Doe’, 1, ‘A’}

If we follow the decomposition, we would expect the tuple above to be broken into two smaller tuples:

{‘John Doe’, 1}  
{‘John Doe’, ‘A’}

Suppose that ‘John Doe’ moves to room 2 and ‘Jane Doe’ moves into room 1 and that ‘Jane Doe’ is using computer ID ‘A’. The following tuples would be in the respective relations:

{‘John Doe’, 2}  
{‘Jane Doe’, 1}  
{‘John Doe’, ‘A’}  
{‘Jane Doe’, ‘A’}

If we were to join back the two relations, we would get:

{‘John Doe’, 2, ‘A’}  
{‘Jane Doe’, 1, ‘A’}

which violates the FD  $\text{ComputerID} \rightarrow \text{RoomNumber}$

### Exercise 3.6.1

Since  $A \twoheadrightarrow B$ , and all the tuples have the same value for attribute A, we can pair the B-value from any tuple with the value of the remaining attribute C from any other tuple. Thus, we know that R must have at least the nine tuples of the form  $(a,b,c)$ , where b is any of  $b_1, b_2$ , or  $b_3$ , and c is any of  $c_1, c_2$ , or  $c_3$ . That is, we can derive, using the definition of a multivalued dependency, that each of the tuples  $(a,b_1,c_2)$ ,  $(a,b_1,c_3)$ ,  $(a,b_2,c_1)$ ,  $(a,b_2,c_3)$ ,  $(a,b_3,c_1)$ , and  $(a,b_3,c_2)$  are also in R.

### Exercise 3.6.2a

First, people have unique Social Security numbers and unique birthdates. Thus, we expect the functional dependencies  $ssNo \rightarrow name$  and  $ssNo \rightarrow birthdate$  hold. The same applies to children, so we expect  $childSSNo \rightarrow childname$  and  $childSSNo \rightarrow childBirthdate$ . Finally, an automobile has a unique brand, so we expect  $autoSerialNo \rightarrow autoMake$ .

There are two multivalued dependencies that do not follow from these functional dependencies. First, the information about one child of a person is independent of other information about that person. That is, if a person with social security number  $s$  has a tuple with  $cn, cs, cb$ , then if there is any other tuple  $t$  for the same person, there will also be another tuple that agrees with  $t$  except that it has  $cn, cs, cb$  in its components for the child name, child Social Security number, and child birthdate. That is the multivalued dependency

$$ssNo \twoheadrightarrow childName \ childBirthdate$$

Similarly, an automobile serial number and make are independent of any of the other attributes, so we expect the multivalued dependency

$$ssNo \twoheadrightarrow autoSerialNo \ autoMake$$

The dependencies are summarized below:

```
ssNo → name, birthdate
childSSNo → childName, childBirthdate
autoSerialNo → autoMake
ssNo →→ childSSNo, childName, childBirthdate
ssNo →→ autoSerialNo, autoMake
```

### Exercise 3.6.2b

We suggest the relation schemas:

```
{ssNo, name, birthdate}
{ssNo, childSSNo}
{childSSNo, childName childBirthdate}
{ssNo, autoSerialNo}
{autoSerialNo, autoMake}
```

An initial decomposition based on the two multivalued dependencies would give us:

```
{ssNo, name, birthDate}
{ssNo, childSSNo, childName, childBirthdate}
{ssNo, autoSerialNo, autoMake}
```

Functional dependencies force us to decompose the second and third of these.

### Exercise 3.6.3a

Since there are no functional dependencies, the only key is all four attributes, ABCD. Thus, each of the nontrivial multivalued dependencies  $A \twoheadrightarrow B$  and  $A \twoheadrightarrow C$  violate 4NF.

We must separate out the attributes of these dependencies, first decomposing into AB and ACD, and then decomposing the latter into AC and AD because  $A \twoheadrightarrow C$  is still a 4NF violation for ACD. The final set of relations is AB, AC, and AD.

### Exercise 3.6.3b

Since there are no functional dependencies, the only key is all four attributes, ABCD. Thus each of the nontrivial multivalued dependencies  $A \twoheadrightarrow B$  and  $B \twoheadrightarrow CD$  violate 4NF.

We must separate out the attributes of these dependencies, decomposing into AB and ACD. There are no 4NF violations for the two decomposed relations so we are done. The final set of relations is AB and ACD.

### Exercise 3.6.3c

From the FD  $B \rightarrow D$ , we can deduce that the only key is ABC. The MVD  $AB \twoheadrightarrow C$  and the derived MVD  $B \twoheadrightarrow D$  are both 4NF violations.

We must separate out the attributes of these dependencies, first decomposing into ABC and ABD, and then decomposing the latter into AB and BD because of the 4NF violation of  $B \twoheadrightarrow D$ . There are no more 4NF violations for the three decomposed relations so we are done. Since the attributes of relation ABC are a superset of the attributes of relation AB, we can discard relation AB. The final set of relations is ABC and BD.

### Exercise 3.6.3d

From the FDs  $A \rightarrow D$  and  $AB \rightarrow E$ , we can deduce that the only key is ABC. The MVDs  $A \twoheadrightarrow B$ ,  $AB \twoheadrightarrow C$  and the derived MVDs  $A \twoheadrightarrow D$  and  $AB \twoheadrightarrow E$  all violate 4NF.

We must separate out the attributes of these dependencies, first decomposing into AB and ACDE. However, there is still a 4NF violation in the latter from the MVD  $A \twoheadrightarrow D$  because A is not a superkey. Thus we further decompose ACDE into relations AD and ACE. There are no more 4NF violations for the three decomposed relations so we are done. The final set of relations is AB, AD and ACE.

### Exercise 3.6.4

We would not expect *name* to be functionally determined by the other four attributes because there could be more than one person living at the same address who starred in the same movie. For example, a husband and wife could star in a romance movie.

We would not expect *street* to be functionally determined by the other four attributes because a person could potentially own multiple houses in the same city.

We would not expect *city* to be functionally determined by the other four attributes because a person could potentially own two houses in different cities with the same street address.

We would not expect *title* to be functionally determined by the other four attributes because a person can star in more than one movie in the same year.

We would not expect *year* to be functionally determined by the other four attributes because a person could potentially star in two movies with the same title but in different years. An example would be a person appearing in the original and then the remake of a movie.

### Exercise 3.7.1a

Our starting tableau is:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>	e <sub>2</sub>

Applying MVD  $A \twoheadrightarrow BC$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>2</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>

Applying FD  $B \rightarrow D$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>

Applying MVD  $C \twoheadrightarrow E$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>

a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>2</sub>

Using the chase test, it appears that the FD  $A \rightarrow D$  holds in the relation.

### Exercise 3.7.1b

Our starting tableau is:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>1</sub>
a	b	c	d <sub>2</sub>	e

Applying MVD  $A \twoheadrightarrow BC$  we get:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>1</sub>
a	b	c	d <sub>2</sub>	e
a	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>2</sub>	e

Applying FD  $B \rightarrow D$  we get:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>1</sub>
a	b	c	d	e
a	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d	e

Applying MVD  $C \twoheadrightarrow E$  we get:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>1</sub>
a	b	c	d	e

a	b	c	d	e <sub>1</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d	e
a	b <sub>1</sub>	c <sub>1</sub>	d	e
a	b	c	d	e <sub>1</sub>
a	b	c	d	e
a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>1</sub>

Since a row of all unsubscripted attributes exists, then the MVD  $A \twoheadrightarrow D$  holds in the relation.

### Exercise 3.7.1c

Our starting tableau is:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>	e <sub>2</sub>

Applying MVD  $A \twoheadrightarrow BC$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>2</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>

Applying FD  $B \rightarrow D$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>

Applying MVD  $C \twoheadrightarrow E$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>2</sub>



a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>2</sub>

Using the chase test, it appears that the FD  $A \rightarrow E$  does not hold in the relation.

### Exercise 3.7.1d

Our starting tableau is:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e
a	b	c	d	e <sub>2</sub>

Applying MVD  $A \twoheadrightarrow BC$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e
a	b	c	d	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>2</sub>
a	b	c	d <sub>1</sub>	e

Applying FD  $B \rightarrow D$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d	e
a	b	c	d	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>2</sub>
a	b	c	d	e

Applying MVD  $C \twoheadrightarrow E$  we get:

A	B	C	D	E
a	b <sub>1</sub>	c <sub>1</sub>	d	e
a	b	c	d	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>2</sub>
a	b	c	d	e

a	b <sub>1</sub>	c <sub>1</sub>	d	e <sub>2</sub>
a	b	c	d	e
a	b <sub>1</sub>	c <sub>1</sub>	d	e
a	b	c	d	e <sub>2</sub>

Since a row of all unsubscripted attributes exists, then the MVD  $A \twoheadrightarrow E$  holds in the relation.

### Exercise 3.7.2

Using the list of simplifications on pg. 120, we can narrow the list of possible FDs down to  $A \rightarrow C$ ,  $A \rightarrow E$ ,  $C \rightarrow A$ ,  $C \rightarrow E$ ,  $AC \rightarrow E$ ,  $AE \rightarrow C$  and  $CE \rightarrow A$ . Similarly, we can narrow down the list of possible MVDs down to  $A \twoheadrightarrow C$ ,  $A \twoheadrightarrow E$ ,  $C \twoheadrightarrow A$  and  $C \twoheadrightarrow E$ .

From these lists, we will have to perform the chase to determine whether the FDs and MVDs hold. Fortunately, we only have to perform the chase once for each unique left hand side of FDs and each unique MVD. Furthermore, for the MVDs, we only need to prove one of  $A \twoheadrightarrow C$  or  $A \twoheadrightarrow E$  and one of  $C \twoheadrightarrow A$  or  $C \twoheadrightarrow E$  because of the complementation rule.

For the FDs  $A \rightarrow C$ ,  $A \rightarrow E$  the initial tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>	e <sub>2</sub>

The final tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e <sub>1</sub>

We conclude that neither  $A \rightarrow C$  nor  $A \rightarrow E$  hold in relation S.

For the FDs  $C \rightarrow A$ ,  $C \rightarrow E$  the initial tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a <sub>1</sub>	b <sub>1</sub>	c	d <sub>1</sub>	e <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c	d <sub>2</sub>	e <sub>2</sub>

The final tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a <sub>1</sub>	b <sub>1</sub>	c	d <sub>1</sub>	e <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c	d <sub>2</sub>	e <sub>2</sub>
a <sub>1</sub>	b <sub>1</sub>	c	d <sub>1</sub>	e <sub>2</sub>
a <sub>2</sub>	b <sub>2</sub>	c	d <sub>2</sub>	e <sub>1</sub>

We conclude that neither  $C \rightarrow A$  nor  $C \rightarrow E$  hold in relation S.

For the FD  $AC \rightarrow E$  the initial tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c	d <sub>2</sub>	e <sub>2</sub>

The final tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c	d <sub>1</sub>	e <sub>1</sub>
a	b <sub>2</sub>	c	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>1</sub>	c	d <sub>1</sub>	e <sub>2</sub>
a	b <sub>2</sub>	c	d <sub>1</sub>	e <sub>1</sub>

We conclude that  $AC \rightarrow E$  does not hold in relation S.

For the FD  $AE \rightarrow C$  the initial tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>	e

The final tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>	e
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>1</sub>	e

We conclude that the FD  $AE \rightarrow C$  does not hold in relation S.

For the FD  $CE \rightarrow A$  the initial tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a <sub>1</sub>	b <sub>1</sub>	c	d <sub>1</sub>	e
a <sub>2</sub>	b <sub>2</sub>	c	d <sub>2</sub>	e

The final tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a <sub>1</sub>	b <sub>1</sub>	c	d <sub>1</sub>	e
a <sub>2</sub>	b <sub>2</sub>	c	d <sub>2</sub>	e

We conclude that the FD  $CE \rightarrow A$  does not hold in relation S.

For the MVD  $A \twoheadrightarrow C$  the initial tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c	d <sub>1</sub>	e <sub>1</sub>
a	b	c <sub>2</sub>	d	e

The final tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	c	d	e <sub>1</sub>
a	b	c <sub>2</sub>	d	e
a	b <sub>1</sub>	c	d	e
a	b	c <sub>2</sub>	d	e <sub>1</sub>

We conclude that the MVD  $A \twoheadrightarrow C$  holds as well as the complement  $A \twoheadrightarrow E$ .

For the MVD  $C \twoheadrightarrow A$  the initial tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
a	b <sub>1</sub>	C	d <sub>1</sub>	e <sub>1</sub>
a <sub>2</sub>	b	C	d	e

The final tableau looks like:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
----------	----------	----------	----------	----------

a	b <sub>1</sub>	C	d <sub>1</sub>	e <sub>1</sub>
a <sub>2</sub>	b	C	d	e
a	b <sub>1</sub>	C	d <sub>1</sub>	e
a <sub>2</sub>	b	C	d	e <sub>1</sub>

We conclude that the MVD  $C \twoheadrightarrow A$  holds as well as the complement  $C \twoheadrightarrow E$ .

Therefore, the only dependencies that hold are  $A \twoheadrightarrow C$ ,  $A \twoheadrightarrow E$ ,  $C \twoheadrightarrow A$  and  $C \twoheadrightarrow E$ .

### Exercise 3.7.3a

Let  $W$  be the set of attributes not in  $X$ ,  $Y$ , or  $Z$ . Consider two tuples  $xyzw$  and  $xy'z'w'$  in the relation  $R$  in question. Because  $X \twoheadrightarrow Y$ , we can swap the  $y$ 's, so  $xy'zw$  and  $xyz'w'$  are in  $R$ . Because  $X \twoheadrightarrow Z$ , we can take the pair of tuples  $xy'zw$  and  $xyz'w'$  and swap the  $z$ 's to get  $xyz'w$  and  $xy'zw'$ . Similarly, we can take the pair  $xy'z'w'$  and  $xy'zw$  and swap  $Z$ 's to get  $xy'zw'$  and  $xy'z'w$ .

In conclusion, we started with tuples  $xyzw$  and  $xy'z'w'$  and showed that  $xyz'w'$  and  $xy'z'w$  must also be in the relation. That is exactly the statement of the MVD  $X \twoheadrightarrow (Y \cup Z)$ .

### Exercise 3.7.3b

Let  $W$  be the set of attributes not in  $X$ ,  $Y$ , or  $Z$ ,  $V$  be the set of attributes that  $Y$  and  $Z$  have in common,  $Y_1$  be the set of attributes of  $Y$  not in  $V$  and  $Z_1$  be the set of attributes of  $Z$  not in  $V$ . Consider the two tuples  $xy_1vz_1w$  and  $xy_1'v'z_1'w'$ . Because  $X \twoheadrightarrow Y$ , we can swap the  $y$ 's so tuples  $xy_1'v'z_1w$  and  $xy_1vz_1'w'$  are in  $R$ . Because  $X \twoheadrightarrow Z$ , we can take the pair  $xy_1vz_1w$  and  $xy_1vz_1'w'$  and swap the  $z$ 's to get  $xy_1vz_1'w$  and  $xy_1vz_1w'$ . Because  $X \twoheadrightarrow Z$ , we can take the pair  $xy_1'v'z_1'w'$  and  $xy_1'v'z_1w$  and swap the  $z$ 's to get  $xy_1'v'z_1'w$  and  $xy_1'v'z_1w'$ . Because  $X \twoheadrightarrow Z$ , we can take the pair  $xy_1'v'z_1w$  and  $xy_1vz_1'w'$  and swap the  $z$ 's to get  $xy_1'v'z_1w$  and  $xy_1vz_1'w'$ . Because  $X \twoheadrightarrow Z$ , we can take the pair  $xy_1vz_1'w'$  and  $xy_1'v'z_1w$  and swap the  $z$ 's to get  $xy_1v'z_1w'$  and  $xy_1'vz_1'w'$ .

In conclusion, we started with tuples  $xy_1vz_1w$  and  $xy_1'v'z_1'w'$  and showed that  $xy_1v'z_1w$  and  $xy_1'vz_1'w'$  must also be in the relation. That is exactly the statement of the MVD  $X \twoheadrightarrow (Y \cap Z)$ .

### Exercise 3.7.3c

Let  $W$  be the set of attributes not in  $X$ ,  $Y$ , or  $Z$ ,  $V$  be the set of attributes that  $Y$  and  $Z$  have in common,  $Y_1$  be the set of attributes of  $Y$  not in  $V$  and  $Z_1$  be the set of attributes of  $Z$  not in  $V$ . Consider the two tuples  $xy_1vz_1w$  and  $xy_1'v'z_1'w'$ . Because  $X \twoheadrightarrow Y$ , we can swap the  $y$ 's so tuples  $xy_1'v'z_1w$  and  $xy_1vz_1'w'$  are in  $R$ . Because  $X \twoheadrightarrow Z$ , we can take the pair  $xy_1'v'z_1w$  and  $xy_1vz_1'w'$  swap the  $z$ 's to get  $xy_1'vz_1w$  and  $xy_1v'z_1w'$ . Because  $X \twoheadrightarrow Z$ , we can take the pair  $xy_1vz_1'w'$  and  $xy_1'v'z_1w$  and swap the  $z$ 's to get  $xy_1v'z_1'w'$  and  $xy_1'vz_1'w'$ .

In conclusion, we started with tuples  $xy_1vz_1w$  and  $xy_1'v'z_1'w'$  and showed that  $xy_1'vz_1w$  and  $xy_1v'z_1'w'$  must also be in the relation. That is exactly the statement of the MVD  $X \twoheadrightarrow (Y - Z)$ .

### Exercise 3.7.3d

Let  $W$  be the set of attributes not in  $X$  or  $Y$ ,  $V$  be the set of attributes that  $X$  and  $Y$  have in common,  $Y_1$  be the set of attributes of  $Y$  not in  $V$  and  $X_1$  be the set of attributes of  $X$  not in  $V$ . Consider the two tuples  $x_1vy_1w$  and  $x_1vy_1'w'$ . Because  $X \twoheadrightarrow Y$ , we can swap the  $y$ 's so tuples  $x_1vy_1'w$  and  $x_1vy_1w'$  are in  $R$ .

In conclusion, we started with tuples  $x_1vy_1w$  and  $x_1vy_1'w'$  and showed that  $x_1vy_1'w$  and  $x_1vy_1w'$  must also be in the relation. That is exactly the statement of the MVD  $X \twoheadrightarrow (Y - X)$ .

### Exercise 3.7.4a

If we want to perform the chase test for  $A \twoheadrightarrow B$ , then an example of an initial tableau is:

A	B	C	D
a	b	c <sub>1</sub>	d <sub>1</sub>
a	b <sub>2</sub>	c	d

If we apply the MVD  $A \twoheadrightarrow BC$ , we get:

A	B	C	D
a	b	c <sub>1</sub>	d <sub>1</sub>
a	b <sub>2</sub>	c	d
a	b <sub>2</sub>	c	d <sub>1</sub>
a	b	c <sub>1</sub>	d

The above tableau does not satisfy  $A \twoheadrightarrow B$ . Thus we would not expect  $A \twoheadrightarrow B$  to follow from  $A \twoheadrightarrow BC$ .

### Exercise 3.7.4b

If we want to perform the chase test for  $A \rightarrow B$ , then an example of an initial tableau is:

A	B	C	D
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>

If we apply the MVD  $A \rightarrow B$ , we get:

A	B	C	D
a	b <sub>1</sub>	c <sub>1</sub>	d <sub>1</sub>
a	b <sub>2</sub>	c <sub>2</sub>	d <sub>2</sub>
a	b <sub>2</sub>	c <sub>1</sub>	d <sub>1</sub>
a	b <sub>1</sub>	c <sub>2</sub>	d <sub>2</sub>

The above tableau does not satisfy  $A \rightarrow B$ . Thus we would not expect  $A \rightarrow B$  to follow from  $A \rightarrow \rightarrow B$ .

### Exercise 3.7.4c

If we want to perform the chase test for  $A \rightarrow \rightarrow C$ , then an example of an initial tableau is:

A	B	C	D
a	b <sub>1</sub>	c	d <sub>1</sub>
a	b	c <sub>2</sub>	d

If we apply the MVD  $AB \rightarrow \rightarrow C$ , we get:

A	B	C	D
a	b <sub>1</sub>	c	d <sub>1</sub>
a	b	c <sub>2</sub>	d

The above tableau does not satisfy  $A \rightarrow \rightarrow C$ . We cannot apply the MVD  $AB \rightarrow \rightarrow C$  because none of the tuples match in both attributes A and B. Thus we would not expect  $A \rightarrow \rightarrow C$  to follow from  $AB \rightarrow \rightarrow C$ .