
```

let  $F_c$  be a canonical cover for  $F$ ;
 $i := 0$ ;
for each functional dependency  $\alpha \rightarrow \beta$  in  $F_c$ 
     $i := i + 1$ ;
     $R_i := \alpha \beta$ ;
if none of the schemas  $R_j, j = 1, 2, \dots, i$  contains a candidate key for  $R$ 
    then
         $i := i + 1$ ;
         $R_i :=$  any candidate key for  $R$ ;
/* Optionally, remove redundant relations */
repeat
    if any schema  $R_j$  is contained in another schema  $R_k$ 
        then
            /* Delete  $R_j$  */
             $R_j := R_i$ ;
             $i := i - 1$ ;
until no more  $R_j$ s can be deleted
return  $(R_1, R_2, \dots, R_i)$ 

```

Figure 7.12 Dependency-preserving, lossless decomposition into 3NF.

The resultant set of schemas can contain redundant schemas, with one schema R_k containing all the attributes of another schema R_j . For example, R_2 above contains all the attributes from R_1 . The algorithm deletes all such schemas that are contained in another schema. Any dependencies that could be tested on an R_j that is deleted can also be tested on the corresponding relation R_k , and the decomposition is lossless even if R_j is deleted.

Now let us consider again the schema of the *class* relation of Section 7.5.1.2 and apply the **3NF decomposition algorithm**. The set of functional dependencies we listed there happen to be a canonical cover. As a result, the algorithm gives us the same three schemas *course*, *classroom*, and *section*.

The preceding example illustrates an interesting property of the 3NF algorithm. Sometimes, the result is not only in 3NF, but also in BCNF. This suggests an alternative method of generating a BCNF design. First use the 3NF algorithm. Then, for any schema in the 3NF design that is not in BCNF, decompose using the BCNF algorithm. If the result is not dependency-preserving, revert to the 3NF design.

7.5.3 Correctness of the 3NF Algorithm

The 3NF algorithm ensures the preservation of dependencies by explicitly building a schema for each dependency in a canonical cover. It ensures that the decomposition is a