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→A.

Now consider pairs of attributes:
$\{AB\}^+$ = ABCD, so we get new dependency AB→D. $\{AC\}^+$ = ACD, and AC→D is nontrivial. $\{AD\}^+$ = AD, so nothing new. $\{BC\}^+$ = ABCD, so we get BC→A, and BC→D. $\{BD\}^+$ = ABCD, giving us BD→A and BD→C. $\{CD\}^+$ = ACD, giving CD→A.

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

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

The collection of 11 new dependencies mentioned above are:
C→A, AB→D, AC→D, BC→A, BC→D, BD→A, BD→C, CD→A, ABC→D, ABD→C, and BCD→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 3.2.10a

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→A and AB→C. Note that BC->A is true, but follows logically from C->A, and therefore may be omitted from our list.

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→A, C→D, D→A, AB→D, AB→ C, AC→D, BC→A, BC→D, BD→A, BD→C, CD→A, ABC→D, ABD→C, and BCD→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→A, C→D, D→A, AC→D, and CD→A.

One choice is to decompose using the violation C→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→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 3.3.4

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.