1. Question ID 522: Let R(ABCDE) be a relation in Boyce-Codd Normal Form (BCNF). If ABC is the only key for R, describe all the nontrivial functional dependencies that hold for R. Identify one of these FD's from the following list.

   Question Explanation: We don't know exactly what the given FD's are. However, it is possible that they are ABC → D and ABC → E. Notice that we could have another set of FD's, like AB → D and AC → E, for which ABC is still the one and only key. However, any FD that **necessarily** holds for R would have to follow from ABC → D and ABC → E. The correct choices all have this property; two are these FD's themselves, one --- ABCD → E --- certainly follows logically, and the fourth --- AB → A --- is trivial. The incorrect choices, however, all have the property that they are nontrivial, but do not have all of ABC on the left side. Therefore, all the closures of the left sides are just the left sides themselves. Since the right side attributes of the incorrect answers are never in the left sides, none of these FD's follow from ABC → D and ABC → E.

2. Question ID 523: Which of the following relations is in Boyce-Codd Normal Form (BCNF)?

   Question Explanation: It is easy to check whether a given relation is in BCNF --- just check that the left side of each given FD is a superkey. We test it is a superkey by computing its closure and verifying that the closure is all the attributes.

3. Question ID 524: Which of the following relations is in Third normal form (3NF)?

   Question Explanation: If the relation is in BCNF, then it surely is in 3NF. Since there is an easy test for BCNF --- check that every left side is a superkey, i.e., its closure is all attributes --- that's an easy way to show some relations in 3NF. But what if some left sides have a closure that is less than all attributes. Then the relation violates BCNF, but might not violate 3NF if the right side is prime --- that is, a member of some key. To check, we have to discover a key of which the right side is a member, and there is no easy way to do this testin general; each set of FD's requires its own reasoning.

   For an example, one of the correct choices consists of FD's ACD → B, AC → D, AC → B, and D → C. The closures of ACD and AC are each ABCD, so those FD's do not violate

BCNF or 3NF. But D+ = DC, so D → C is a BCNF violation. To check that it is not a 3NF violation, we need to verify that C is in some key. We already know AC is a superkey. But it is easy to check that neither A nor C are superkeys, and thus AC is a key. we conclude this relation is in 3NF.

4. Question ID 525: Let R(ABCD) be a relation with functional dependencies

A → B, C → D, AD → C, BC → A

Which of the following is a lossless-join decomposition of R into Boyce-Codd Normal Form (BCNF)?

Question Explanation: The trick to telling whether a decomposition has a lossless join is to imagine that there is a tuple, say (a,b,c,d), in the result of projecting R(A,B,C,D) onto the decomposed relations, and then rejoining them. If we can prove, using the given dependencies, that (a,c,b,d) was really in R, then we have proved the decomposition has a lossless join. Of course, we must check that the decomposition also has only BCNF relations, but in most cases the suggested schemas have only two attributes, and every two-attribute relation is in BCNF.

For example, consider one of the correct choices {AB, AC, CD}. The tuple (a,b,c,d) could only come from some tuples in R of the form (a,b,c1,d1), (a,b2,c,d2), and (a3,b3,c,d), which project, respectively, as (a,b) in AB, (a,c) in AC, and (c,d) in CD. But A → B tells us b2=b, and C → D tells us d2=d. Thus, the tuple (a,b2,c,d2) is really (a,b,c,d). Since we know the former is in R, the latter is in R, which is exactly what we wanted to prove.

NOTE: This Gradiance solution simply explains informally our test for lossless join.