CHAPTER 15: Database Design Theory: Normalization Algorithms

Answers to Selected Exercises

15.17 - Show that the relation schemas produced by Algorithm 15.4 are in 3NF.

Answer:
We give a proof by contradiction. Suppose that one of the relations R i resulting from Algorithm 15.1 is not in 3NF. Then a FD Y -> A holds R i in where: (a) Y is not a superkey of R, and (b) A is not a prime attribute. But according to step 2 of the algorithm, R i will contain a set of attributes X union A 1 union A 2 union ... union A n , where X -> A i for i=1, 2, ..., n, implying that X is a key of R i and the A i are the only non-prime attributes of R i . Hence, if an FD Y -> A holds in R i where A is non-prime and Y is not a superkey of R i , Y must be a proper subset of X (otherwise Y would contain X and hence be a superkey). If both Y -> A and X -> A hold and Y is a proper subset of X, this contradicts that X -> A is a FD in a minimal set of FDs that is input to the algorithm, since removing an attribute from X leaves a valid FD, thus violating one of the minimality conditions. This produces a contradiction of our assumptions. Hence, R i must be in 3NF.

15.18 - Show that, if the matrix S resulting from Algorithm 15.3 does not have a row that is all "a" symbols, then projecting S on the decomposition and joining it back will always produce at least one spurious tuple.

Answer:
The matrix S initially has one row for each relation R i in the decomposition, with "a" symbols under the columns for the attributes in R i . Since we never change an "a" symbol into a "b" symbol during the application of the algorithm, then projecting S on each R i at the end of applying the algorithm will produce one row consisting of all "a" symbols in each S(R i). Joining these back together again will produce at least one row of all "a" symbols (resulting from joining the all "a" rows in each projection S(R i)). Hence, if after applying the algorithm, S does not have a row that is all "a", projecting S over the R i 's and joining will result in at least one all "a" row, which will be a spurious tuple (since it did not exist in S but will exist after projecting and joining over the R i 's).

15.19 - Show that the relation schemas produced by Algorithm 15.5 are in BCNF.

Answer:
This is trivial, since the algorithm loop will continue to be applied until all relation schemas are in BCNF.

15.20 – **No Solution Provided**

15.21 - Specify a template dependency for join dependencies.

Answer:
The following template specifies a join dependency JD(X,Y,Z).

R = { A , B , C }

	A	B	C
hypothesis	a	b	c_1
	a	b_1	c
	a_1	b	c
conclusion	a	b	c

X = { A , B }
Y = { B , C }
Z = { A , C }

15.22 - Specify all the inclusion dependencies for the relational schema of Figure 3.5.

Answer:
The inclusion dependencies will correspond to the foreign keys shown in Figure 3.7.

15.23 - Prove that a functional dependency satisfies the formal definition of multi-valued dependency.

Answer:
Suppose that a functional dependency X -> Y exists in a relation R={X, Y, Z}, and suppose there are two tuples with the same value of X. Because of the functional dependency, they must also have the same value of Y. Suppose the tuples are t 1 = < x, y, z 1 > and t 2 = < x, y, z 2 >. Then, according to the definition of multivalued dependency, we must have two tuples t 3 and t 4 (not necessarily distinct from t 1 and t 2) satisfying: t 3 [X]= t 4 [X]= t 1 [X]= t 2 [X], t 3 [Y]= t 2 [Y], t 4 [Y]= t 1 [Y], t 3 [Z]= t 1 [Z], and t 4 [Z]= t 2 [Z]. Two tuples satisfying this are t 2 (satisfies conditions for t 4) and t 1 (satisfies conditions for t 3). Hence, whenever the condition for functional dependency holds, so does the condition for multivalued dependency.

15.24 - 15.31: No solutions provided.

15.32 - Consider the relation REFRIG(MODEL#, YEAR, PRICE, MANUF_PLANT, COLOR), which is abbreviated as REFRIG(M, Y, P, MP, C), and the following set of F of functional dependencies: F={M -> MP, {M,Y} -> P, MP -> C}

(a) Evaluate each of the following as a candidate key for REFRIG, giving reasons why it can or cannot be a key: {M}, {M,Y}, {M.C}

(b) Based on the above key determination, state whether the relation REFRIG is in 3NF and in BCNF, giving proper reasons.

(c) Consider the decomposition of REFRIG into D={R1(M,Y,P), R2(M,MP,C)}. Is this decomposition lossless? Show why. (You may consult the test under Property LJ1 in Section 15.2.4)

Answers:
(a)
- {M} IS NOT a candidate key since it does not functionally determine attributes Y or P.
- {M, Y} IS a candidate key since it functionally determines the remaining attributes P, MP, and C.
i.e.
{M, Y} P, But M MP
By augmentation {M, Y} MP
Since MP C, by transitivity M MP, MP C, gives M C
By augmentation {M, Y} C
Thus {M, Y} P, MP, C and {M, Y} can be a candidiate key
- {M, C} IS NOT a candidate key since it does not functionally determine attributes Y or P.

(b)
REFRIG is not in 2NF, due to the partial dependency {M, Y} MP (since {M} MP holds). Therefore REFRIG is neither in 3NF nor in BCNF.
Alternatively: BCNF can be directly tested by using all of the given dependencies and finding out if the left hand side of each is a superkey (or if the right hand side is a prime attribute). In the two fields in REFRIG: M MP and MP C. Since neither M nor MP is a superkey, we can conclude that REFRIG is is neither in 3NF nor in BCNF.

(c)

R = {M, Y, P, MP, C}
R1 = {M, Y, P}
R2 = {M, MP, C}

F = {M MP, {M, Y} P, MP C}
F^+ = { $\{M\}^+$ {M, MP, C},
$\{M, Y\}^+$ {M, Y, P, MP, C},
$\{MP\}^+$ {MP, C} }

R1 $\cap$ R2 = M
R2 − R1 = {MP, C}

D(R1, R2} has the lossless join property since
Property: *LJ1: FD ((R1 $\cap$ R2) (R2 − R1)) is in F^+*
is satisfied (due to M {MP, C}).