19

SCHEMA REFINEMENT AND NORMAL FORMS

Exercise 19.1 Briefly answer the following questions:

1. Define the term *functional dependency*.
2. Why are some functional dependencies called *trivial*?
3. Give a set of FDs for the relation schema *R(A,B,C,D)* with primary key AB under which R is in 1NF but not in 2NF.
4. Give a set of FDs for the relation schema *R(A,B,C,D)* with primary key AB under which R is in 2NF but not in 3NF.
5. Consider the relation schema *R(A,B,C)*, which has the FD $B \rightarrow C$. If A is a candidate key for R, is it possible for R to be in BCNF? If so, under what conditions? If not, explain why not.
6. Suppose we have a relation schema *R(A,B,C)* representing a relationship between two entity sets with keys A and B, respectively, and suppose that R has (among others) the FDs $A \rightarrow B$ and $B \rightarrow A$. Explain what such a pair of dependencies means (i.e., what they imply about the relationship that the relation models).

Answer 19.1

1. Let R be a relational schema and let X and Y be two subsets of the set of all attributes of R. We say Y is functionally dependent on X, written X → Y, if the Y-values are determined by the X-values. More precisely, for any two tuples r_1 and r_2 in (any instance of) R

$$\pi_X(r_1) = \pi_X(r_2) \quad \Rightarrow \quad \pi_Y(r_1) = \pi_Y(r_2)$$

2. Some functional dependencies are considered trivial because they contain superfluous attributes that do not need to be listed. Consider the FD: $A \rightarrow AB$. By reflexivity, A always implies A, so that the A on the right hand side is not necessary and can be dropped. The proper form, without the trivial dependency would then be $A \rightarrow B$.

3. Consider the set of FD: $AB \rightarrow CD$ and $B \rightarrow C$. AB is obviously a key for this relation since $AB \rightarrow CD$ implies $AB \rightarrow ABCD$. It is a primary key since there are no smaller subsets of keys that hold over *R(A,B,C,D)*. The FD: $B \rightarrow C$ violates 2NF since:
   - $C \in B$ is false; that is, it *is not* a trivial FD
   - B *is not* a superkey
   - C *is not* part of some key for R
   - B *is* a proper subset of the key AB (transitive dependency)

4. Consider the set of FD: $AB \rightarrow CD$ and $C \rightarrow D$. AB is obviously a key for this relation since $AB \rightarrow CD$ implies $AB \rightarrow ABCD$. It is a primary key since there are no smaller subsets of keys that hold over *R(A,B,C,D)*. The FD: $C \rightarrow D$ violates 3NF but not 2NF since:
   - $D \in C$ is false; that is, it *is not* a trivial FD
   - C *is not* a superkey
   - D *is not* part of some key for R

5. The only way R could be in BCNF is if B includes a key, *i.e.* B is a key for R.

6. It means that the relationship is one to one. That is, each A entity corresponds to at most one B entity and vice-versa. (In addition, we have the dependency $AB \rightarrow C$, from the semantics of a relationship set.)

Exercise 19.2 Consider a relation R with five attributes $ABCDE$. You are given the following dependencies: $A \rightarrow B$, $BC \rightarrow E$, and $ED \rightarrow A$.

1. List all keys for R.
2. Is R in 3NF?
3. Is R in BCNF?

Answer 19.2

1. CDE, ACD, BCD
2. R is in 3NF because B, E and A are all parts of keys.
3. R is not in BCNF because none of A, BC and ED contain a key.

Exercise 19.3 Consider the relation shown in Figure 19.1.

1. List all the functional dependencies that this relation instance satisfies.

X	Y	Z
x_1	y_1	z_1
x_1	y_1	z_2
x_2	y_1	z_1
x_2	y_1	z_3

Figure 19.1 Relation for Exercise 19.3.

2. Assume that the value of attribute Z of the last record in the relation is changed from z_3 to z_2. Now list all the functional dependencies that this relation instance satisfies.

Answer 19.3

1. The following functional dependencies hold over R: $Z \rightarrow Y$, $X \rightarrow Y$, and $XZ \rightarrow Y$
2. Same as part 1. Functional dependency set is unchanged.

Exercise 19.4 Assume that you are given a relation with attributes $ABCD$.

1. Assume that no record has NULL values. Write an SQL query that checks whether the functional dependency $A \rightarrow B$ holds.
2. Assume again that no record has NULL values. Write an SQL assertion that enforces the functional dependency $A \rightarrow B$.
3. Let us now assume that records could have NULL values. Repeat the previous two questions under this assumption.

Answer 19.4 Assuming...

1. The following statement returns 0 iff no statement violates the FD $A \rightarrow B$.

```
SELECT COUNT (*)
FROM   R AS R1, R AS R2
WHERE  (R1.B != R2.B) AND (R1.A = R2.A)
```

2.
```
CREATE ASSERTION ADeterminesB
CHECK  ((SELECT COUNT (*)
        FROM   R AS R1, R AS R2
        WHERE  (R1.B != R2.B) AND (R1.A = R2.A))
        =0)
```

3. Note that the following queries can be written with the NULL and NOT NULL interchanged. Since we are doing a full join of a table and itself, we are creating tuples in sets of two therefore the order is not important.

```
SELECT COUNT (*)
FROM   R AS R1, R AS R2
WHERE  ((R1.B != R2.B) AND (R1.A = R2.A))
       OR ((R1.B is NULL) AND (R2.B is NOT NULL)
              AND (R1.A = R2.A))

CREATE ASSERTION ADeterminesBNull
CHECK  ((SELECT COUNT (*)
       FROM   R AS R1, R AS R2
       WHERE  ((R1.B != R2.B) AND (R1.A = R2.A)))
              OR ((R1.B is NULL) AND (R2.B is NOT NULL)
                     AND (R1.A = R2.A))
       =0)
```

Exercise 19.5 Consider the following collection of relations and dependencies. Assume that each relation is obtained through decomposition from a relation with attributes *ABCDEFGHI* and that all the known dependencies over relation *ABCDEFGHI* are listed for each question. (The questions are independent of each other, obviously, since the given dependencies over *ABCDEFGHI* are different.) For each (sub)relation: (a) State the strongest normal form that the relation is in. (b) If it is not in BCNF, decompose it into a collection of BCNF relations.

1. *R1(A,C,B,D,E)*, $A \rightarrow B$, $C \rightarrow D$
2. *R2(A,B,F)*, $AC \rightarrow E$, $B \rightarrow F$
3. *R3(A,D)*, $D \rightarrow G$, $G \rightarrow H$
4. *R4(D,C,H,G)*, $A \rightarrow I$, $I \rightarrow A$
5. *R5(A,I,C,E)*

Answer 19.5

1. 1NF. BCNF decomposition: AB, CD, ACE.
2. 1NF. BCNF decomposition: AB, BF
3. BCNF.
4. BCNF.
5. BCNF.

Exercise 19.6 Suppose that we have the following three tuples in a legal instance of a relation schema S with three attributes ABC (listed in order): (1,2,3), (4,2,3), and (5,3,3).

1. Which of the following dependencies can you infer does *not* hold over schema S?

   (a) $A \rightarrow B$, (b) $BC \rightarrow A$, (c) $B \rightarrow C$

2. Can you identify any dependencies that hold over S?

Answer 19.6

1. $BC\rightarrow A$ does not hold over S (look at the tuples (1,2,3) and (4,2,3)). The other tuples hold over S.

2. No. Given just an instance of S, we can say that certain dependencies (e.g., $A \rightarrow B$ and B $\rightarrow C$) are not violated by this instance, but we cannot say that these dependencies hold with respect to S. To say that an FD holds w.r.t. a relation is to make a statement about *all* allowable instances of that relation!

Exercise 19.7 Suppose you are given a relation R with four attributes $ABCD$. For each of the following sets of FDs, assuming those are the only dependencies that hold for R, do the following: (a) Identify the candidate key(s) for R. (b) Identify the best normal form that R satisfies (1NF, 2NF, 3NF, or BCNF). (c) If R is not in BCNF, decompose it into a set of BCNF relations that preserve the dependencies.

1. $C \rightarrow D,\ C \rightarrow A,\ B \rightarrow C$
2. $B \rightarrow C,\ D \rightarrow A$
3. $ABC \rightarrow D,\ D \rightarrow A$
4. $A \rightarrow B,\ BC \rightarrow D,\ A \rightarrow C$
5. $AB \rightarrow C,\ AB \rightarrow D,\ C \rightarrow A,\ D \rightarrow B$

Answer 19.7

1. (a) Candidate keys: B
   (b) R is in 2NF but not 3NF.
   (c) $C \rightarrow D$ and $C \rightarrow A$ both cause violations of BCNF. One way to obtain a (lossless) join preserving decomposition is to decompose R into AC, BC, and CD.
2. (a) Candidate keys: BD
   (b) R is in 1NF but not 2NF.

   (c) Both $B \rightarrow C$ and $D \rightarrow A$ cause BCNF violations. The decomposition: *AD*, *BC*, *BD* (obtained by first decomposing to *AD*, *BCD*) is BCNF and lossless and join-preserving.

3. (a) Candidate keys: *ABC*, *BCD*

   (b) *R* is in 3NF but not BCNF.

   (c) *ABCD* is not in BCNF since $D \rightarrow A$ and *D* is not a key. However if we split up *R* as *AD*, *BCD* we cannot preserve the dependency $ABC \rightarrow D$. So there is no BCNF decomposition.

4. (a) Candidate keys: *A*

   (b) *R* is in 2NF but not 3NF (because of the FD: $BC \rightarrow D$).

   (c) $BC \rightarrow D$ violates BCNF since *BC* does not contain a key. So we split up *R* as in: *BCD*, *ABC*.

5. (a) Candidate keys: *AB*, *BC*, *CD*, *AD*

   (b) R is in 3NF but not BCNF (because of the FD: C → A).

   (c) $C \rightarrow A$ and $D \rightarrow B$ both cause violations. So decompose into: *AC*, *BCD* but this does not preserve $AB \rightarrow C$ and $AB \rightarrow D$, and *BCD* is still not BCNF because $D \rightarrow B$. So we need to decompose further into: *AC*, *BD*, *CD*. However, when we attempt to revive the lost functioanl dependencies by adding *ABC* and *ABD*, we that these relations are not in BCNF form. Therefore, there is no BCNF decomposition.

Exercise 19.8 Consider the attribute set $R = ABCDEGH$ and the FD set $F = \{AB \rightarrow C,\ AC \rightarrow B,\ AD \rightarrow E,\ B \rightarrow D,\ BC \rightarrow A,\ E \rightarrow G\}$.

1. For each of the following attribute sets, do the following: (i) Compute the set of dependencies that hold over the set and write down a minimal cover. (ii) Name the strongest normal form that is not violated by the relation containing these attributes. (iii) Decompose it into a collection of BCNF relations if it is not in BCNF.

   (a) *ABC*, (b) *ABCD*, (c) *ABCEG*, (d) *DCEGH*, (e) *ACEH*

2. Which of the following decompositions of $R = ABCDEG$, with the same set of dependencies *F*, is (a) dependency-preserving? (b) lossless-join?

   (a) {*AB*, *BC*, *ABDE, EG* }

   (b) {*ABC, ACDE, ADG* }

Answer 19.8

1. (a) i. *R1* = *ABC*: The FD's are $AB \rightarrow C$, $AC \rightarrow B$, $BC \rightarrow A$.

ii. This is already a minimal cover.

iii. This is in BCNF since AB, AC and BC are candidate keys for *R1*. (In fact, these are all the candidate keys for *R1*).

(b) i. $R2 = ABCD$: The FD's are $AB \rightarrow C$, $AC \rightarrow B$, $B \rightarrow D$, $BC \rightarrow A$.

ii. This is a minimal cover already.

iii. The keys are: AB, AC, BC. *R2* is not in BCNF or even 2NF because of the FD, $B \rightarrow D$ (B is a proper subset of a key!) However, it is in 1NF. Decompose as in: ABC, BD. This is a BCNF decomposition.

(c) i. $R3 = ABCEG$; The FDs are $AB \rightarrow C$, $AC \rightarrow B$, $BC \rightarrow A$, $E \rightarrow G$.

ii. This is in minimal cover already.

iii. The keys are: ABE, ACE, BCE. It is not even in 2NF since E is a proper subset of the keys and there is a FD $E \rightarrow G$. It is in 1NF . Decompose as in: ABE, ABC, EG. This is a BCNF decompostion.

(d) i. $R4 = DCEGH$; The FD is $E \rightarrow G$.

ii. This is in minimal cover already.

iii. The key is $DCEH$; It is not in BCNF since in the FD $E \rightarrow G$, E is a subset of the key and is not in 2NF either. It is in 1 NF Decompose as in: $DCEH$, EG

(e) i. $R5 = ACEH$; No FDs exist.

ii. This is a minimal cover.

iii. Key is $ACEH$ itself.

iv. It is in BCNF form.

2. (a) The decomposition. { AB, BC, ABDE, EG } is *not* lossless. To prove this consider the following instance of R:

$$\{(a_1, b, c_1, d_1, e_1, g_1),\ (a_2, b, c_2, d_2, e_2, g_2)\}$$

Because of the functional dependencies $BC \rightarrow A$ and $AB \rightarrow C$, $a_1 \neq a_2$ if and only if $c_1 \neq c_2$. It is easy to that the join AB $\bowtie$ BC contains 4 tuples:

$$\{(a_1, b, c_1),\ (a_1, b, c_2),\ (a_2, b, c_1),\ (a_2, b, c_2)\}$$

So the join of AB, BC, $ABDE$ and EG will contain at least 4 tuples, (actually it contains 8 tuples) so we have a lossy decomposition here.

This decomposition does not preserve the FD, $AB \rightarrow C$ (or $AC \rightarrow B$)

(b) The decomposition {ABC, ACDE, ADG } is lossless. Intuitively, this is because the join of ABC, $ACDE$ and ADG can be constructed in two steps; first construct the join of ABC and ACDE: this is lossless because their (attribute) intersection is AC which is a key for $ABCDE$ (in fact $ABCDEG$) so this is lossless. Now join this intermediate join with ADG. This is also lossless because the attribute intersection is is AD and $AD \rightarrow ADG$. So by the test mentioned in the text this step is also a lossless decomposition.